The
Lightweight & Efficient Application Protocols (LEAP)
Manifesto

Using
Free Protocols & Free Software
to build the
Mobile & Wireless Applications Industry

Mohsen Banan
http://mohsen.banan.1.byname.net/ContactMe

Version 1.3
October 1, 2001

This document describes the Lightweight & Efficient Application Protocols (LEAP), a set of protocols for mobile and wireless applications.

Trademark Information

Contents

List of Figures

List of Tables

Chapter 1

Executive Summary

Until now, the Internet has been largely based upon simple protocols. However, the era of simple protocols is now over. The new Internet reality is that of wireless networks, providing service to legions of miniaturized, hand-held mobile devices. This reality places an entirely new set of requirements on the underlying communications protocols: they must now provide the power efficiency demanded by hand-held wireless devices, together with the bandwidth efficiency demanded by wide area wireless networks.

It is now time for a new generation of protocols to be implemented, designed to address the need for *performance*, rather than simplicity.

The industry-wide adoption of this new generation of powerful and efficient protocols will have enormous consequences. Protocols addressing the correct requirements will become the lynchpin of a huge new industry. The stakes are enormous, and ferocious competition is to be expected within all segments of the industry. All manner of wild claims and misrepresentations are also to be expected. At the time of writing, the main claimant to the protocol throne is the Wireless Applications Protocol, or WAP. However, WAP will eventually prove to be entirely inadequate to the role being claimed for it.

We have designed a set of protocols, the **Lightweight & Efficient Application Protocols**, or **LEAP**, which we believe is destined to displace WAP and become the *de facto* industry standard. These protocols, published as Internet RFC-2524 and RFC-2188, are designed to address all the technical requirements of the industry, and are oriented towards providing the greatest benefit to the industry and the consumer.

This manifesto is about our vision of the future of the Mobile and Wireless Applications Industry. In the remainder of the manifesto we present the details of our vision, and we justify our claims. We justify our assertion that the industry needs a new generation of protocols, we explain why our protocols fulfil this need, and we describe how and why these protocols will achieve dominance.

The protocols are free, open and in place. Open-source software implementations of the protocols are available for all major platforms. The combination of free protocols and open-source software ensures acceptance of the protocols in the Internet mainstream. There can be no stopping this.

1.1 Technological Scope

Most of our discussion throughout this Manifesto is framed in terms of a particular technology, namely, Mobile Messaging. It is important to bear in mind, however, that Mobile Messaging is just one aspect of a broader technology: Mobile Consumer Data Communications. Mobile Consumer Data Communications refers to the general ability of an end-user to send and receive digital data at a hand-held device via a wireless network. This technology includes Mobile

Messaging as a special case, but also includes other wireless data transfer capabilities such as general Internet access, web browsing, etc.

Much of the discussion set forth in this Manifesto applies with equal force to all mobile data communications applications, not just that of messaging. However, it is currently well understood that the dominant application for mobile data communications is, in fact, Mobile Messaging, not web browsing or other Internet applications. Therefore throughout this Manifesto we will focus our attention on the messaging application.

Though our discussion will be framed in terms of Mobile Messaging, the reader should bear in mind that the same principles apply to all forms of mobile data communications.

Also, whenever we speak of the Mobile Messaging industry, we are referring to the totality of what is required to accomplish effective mobile messaging capabilities for the end user.

We are not referring to the implementation of mobile messaging on any particular device, such as a mobile phone, PDA, palmtop PC, laptop PC, or two-way pager. Similarly, we are not restricting our focus to any specific technologies or standards, nor are we restricting our focus to a specific market or set of subscriber services.

Rather, we are referring to the entire set of technologies and constituencies which are required to enable Mobile Messaging. This includes: mobile handheld devices and their manufacturers, wireless modems and their manufacturers, wireless data networks and their operators, ISPs and other service providers, and the set of protocols and software implementations required to allow interplay and cooperation among these various consituencies.

Our purpose in writing this Manifesto is very ambitious: we wish to describe our vision of all that is required to build the entire Mobile Messaging industry.

1.2 Efficiency is the Key Requirement

Engineering is the art of making intelligent trade-offs between conflicting requirements. A perennial engineering trade-off is that which must be made between the need for *simplicity*, and the need for *performance*. In the case of wireless data communications, performance means such things as data transfer speed, power efficiency, and bandwidth efficiency.

The 1980s and 1990s were the decades of simple protocols - protocols such as the very aptly named Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP). A great deal of the success of these and other Internet protocols can be attributed to their simplicity.

The first generation of network engineers and network operators were only able to view network communications in relatively simple terms. It was appropriate to cater to that simplicity with simple protocols. A key reason for the success of these early protocols is the lack of technical sophistication on the part of first-generation network engineers and operators.

Simple protocols are easier to make widespread than “good” protocols (meaning those which have better capabilities and performance), for the basic reason that network engineers and operators are able to adopt and implement simple protocols much more easily than “good” protocols.

However, things have changed. Network communications has now expanded dramatically and forcefully into the wireless and mobile data communications arena, and *wireless applications demand efficiency*. The move to wide-area wireless has significantly shifted the location of the ideal engineering balance between simplicity and performance - moving it away from simplicity, and towards performance.

We therefore need a new generation of high-performance, efficient protocols, to cater to the demands of wireless applications. The point is sometimes made that the need for efficiency in the wireless arena is a temporary one – that advances in wireless engineering technology in the form of third generation (3G) systems will eliminate existing bandwidth limitations, obviating the need for efficient protocols. As long as the capacity of wireless networks remains finite, however, the need for efficiency will persist. Efficient usage is an inherent requirement for any finite resource, therefore the requirement for efficient bandwidth usage and battery longevity is permanent.

1.3 Conventional Origins of Protocols

Where will the required protocols come from? Traditionally, industry-wide protocols have their origins in one of two sources:

1. The major players in the industry itself. In the case of wireless communications, this means the major telecommunications and wireless network companies.
2. Professional protocol and standards producing associations. In the case of wireless communications, this means the IETF, ITU, ISO, ANSI, TIA and others.

Unfortunately, neither of these groups has produced a set of protocols which meets the industry's needs. The first group above, represented by a set of telephone companies, has generated the WAP specification. However, as we will argue in detail later, this specification is grossy unfit for its claimed purpose. Among other things it is poorly designed, not the product of open peer review, and crippled with Intellectual Property Right (IPR) restrictions. It is essentially a business construct, not an engineering one. In the long run WAP cannot possibly survive as a viable solution. In the short run it can only have a destructive effect on the wireless industry.

The second group above, most notably represented by IETF, has likewise failed to produce an acceptable standard. IETF represents the tradition of simple protocols, a tradition which wireless communications has made obsolete. Unfortunately, IETF remains rooted in this tradition, and has not adapted to the new realities of wireless communications. Until it does so, IETF will remain ineffective as a protocols and standards body. In the area of efficient protocols, IETF is simply bankrupt.

1.4 Expect the Unexpected

Fortunately, there are other sources of innovation. One of these is the radical new development that comes out of nowhere, taking everybody by surprise. Typically this originates in the actions of a small group of independent experts, with a deep understanding of the technology and industry, and who are passionate about and committed to its health and vigor.

Note that the World Wide Web itself originated in neither of the traditional sources, but instead came from an entirely different and unexpected direction: a group of physicists at the CERN laboratory in Switzerland. As another example, Pretty Good Privacy (PGP), now the *de facto* standard for electronic data encryption, also came from neither traditional source. It was essentially the creation of a single man: Phil Zimmermann. Armed with a vision and a belief in its value, Zimmermann single-handedly made PGP the dominant consumer encryption application - displacing the IETF alternatives in the process.

The solution to the current wireless application dilemma is also likely to come from an unexpected source – and we believe that we are that source. In the world of the Internet, we have learned to expect the unexpected.

1.5 Our Solution

We have developed a set of protocols which we believe address all aspects of the industry's needs. Beyond their purely technical requirements, a fundamental requirement of all industry-building protocols is that they be completely open and free from patents and other IPR restrictions – either because no patents actually exist, or because reasonably non-restrictive licenses are granted by the patent holder. In the rest of this document, this is what we mean when we speak of "patent-free" protocols.

The presence of patented components within a protocol is extremely undesirable, since this undermines the ultimate purpose of the protocol: its unrestricted adoption and usage. The process that we have followed in developing our protocols has been such as to ensure that they are entirely open and, as far as this can be guaranteed, patent-free. A significant part of this process consists of our full committment to the processes and procedures of the Free Protocols Foundation (FPF).

The FPF is an organizational framework for the development and maintenance of free protocols. It allows developers to declare publicly that the protocols they have developed are intended to be patent-free, and that it is their intention to keep them patent-free into perpetuity. We have made this declaration through the Free Protocols Foundation with regard to our own protocols.

Note that this is in sharp contrast to the WAP protocols, which include severe IPR restrictions. This creates an unfair market advantage in favor of the initial WAP designers. Our intention is to create a protocol which does not favor any one industry player over another, and places competition where it belongs: on the merits of each company's individual products and services.

We have created the general framework for a set of high-performance, efficient protocols which are ideal for mobile and wireless applications. We refer to this general framework as the Lightweight & Efficient Application Protocols (LEAP).

The need for efficient protocols extends across all aspects of wireless data communications, including e-mail, web browsing, and other applications. The LEAP architecture accommodates all of these applications. Our initial implementation, however, is focussed on the Mobile Messaging application, since we believe that this is the dominant application for wide-area wireless networks.

All efficient applications have the requirement for an efficient transport mechanism. For this reason, the initial focus of our protocol development effort has been on creating a general efficient transport mechanism. The resulting protocol is referred to as Efficient Short Remote Operations (ESRO). ESRO is a reliable, connectionless transport mechanism, forming the foundation for the development of efficient protocols when TCP is too much and UDP is too little.

Our Efficient Mail Submission and Delivery (EMSD) protocol is built on top of ESRO, and is designed to address the Mobile Messaging application.

Both of these protocols have been published as Internet RFCs: ESRO as RFC 2188, and EMSD as RFC 2524. RFC publication ensures that the protocols are freely, easily and permanently accessible to anyone who wishes to use them.

Note that this also is in stark contrast to WAP, which is self-published by the members-only WAP Forum. Furthermore, the WAP Forum reserves the right to make unilateral changes to its protocols; each of the WAP protocols carries on its cover page the disclaimer, "subject to change without notice."

Publication of a protocol as an Internet RFC ensures that the protocol will remain stable and permanently available to anyone who wishes to use it, and for this reason is the mainstream Internet publishing method. The declining of the WAP Forum to publish their specifications as Internet RFCs suggests either that the forum wishes to retain an inappropriate degree of control over the specifications, or that the specifications do not meet the minimum technical standards required for RFC publication.

1.6 A Brief History of LEAP

LEAP originated in 1994 as part of the research and development initiatives of McCaw Cellular's wireless data group (now AT&T Wireless Services). The development work that would eventually lead to LEAP was initially undertaken in the context of the CDPD network; its scope was later expanded to include the Narrowband PCS network also.

By 1996 McCaw Cellular was fully committed to paging, had recently purchased two nationwide narrowband wireless PCS licenses, and wished to develop an efficient wireless message transport and delivery system. Neda Communications, Inc., an independent consulting company working under contract to McCaw Cellular, played a significant role in the development of the required system. Neda Communications had also been involved from the outset in the development of the CDPD specification.

In 1997 however, soon after the purchase of McCaw Cellular by AT&T, the latter company abandoned narrowband PCS paging altogether. Prior to this event, Neda Communications had secured from AT&T the necessary rights to continue independent development of the protocols. Therefore, recognizing the eventual future need for these protocols, Neda then undertook to continue development of the protocols independently of AT&T. They were eventually completed by Neda, published as RFCs, and now form the cornerstone of the LEAP protocols.

1.7 Making Our Solution Widespread

Our ultimate goal is to make these protocols widespread. Developing and publishing a set of protocols, however, is just the beginning. Protocols become accepted as standards as a result of public review, modification by consensus, and ultimately by standing the test of usage in the industry at large.

To provide a forum for these processes, we have created EMSD.org and ESRO.org. Each of these organizations allows public review of the respective protocol, and provides a mechanism for correction and enhancement of the protocol as a result of collective experience. Any interested person can become a member of these organizations and participate in the further development of the protocols. The only requirement for membership is that participants must adhere to the principles and procedures of the Free Protocols Foundation, ensuring that the protocols remain permanently patent-free.

Note that this also is in sharp contrast to WAP. Participation in WAP, far from being open and public, requires a $27,000 membership fee (as of February 2000), and takes place entirely behind closed doors.

In order for the protocols to become widely accepted, they must be implemented in the form of software solutions that are readily available for deployment by end-users. We have therefore created open-source software implementations of the protocols for most common platforms. Protocol engines are available in the form of portable code which has been ported to a variety of platforms. On the device side, software is available for Windows CE, Palm OS, EPOC, and others. On the message center side, software is available for NT, Solaris, and Linux.

As noted above, our initial emphasis is on the Mobile Messaging application. Protocol engines are only a single component of a larger picture; in order to provide complete solutions to the user it is necessary to integrate these protocols into other existing pieces of software. To that end we have created MailMeAnywhere.org, where fully-integrated solutions in open-source format are made available to the user.

We will initially “prime the pump” by providing free subscriber services through ByName.net and ByNumber.net. This will provide initial support for adoption of the protocols by end-user devices. Usage of the protocols among a sufficient number of user devices will then provide the motivation for usage among the message center systems.

1.8 Complete and Ready

All the components that are needed to accomplish these goals are complete, in place, and ready to go. These components are:

The Protocols. The protocols are well-designed, meet all the technical requirements of the industry, and are published as RFCs – the mainstream Internet publishing procedure. The complete text of RFC 2188 and RFC 2524 is available at:

`http://www.rfc-editor.org`

Open Maintenance Organizations. The protocols are maintained at EMSD.org and ESRO.org, allowing open and non-exclusionary participation in the maintenance of the protocols. For complete details see:

`http://www.esro.org` and
`http://www.emsd.org`

Freedom from Patents. The protocols are patent-free to the best of our knowledge, and are guaranteed to stay that way. This ensures permanent, unrestricted access to the protocols. For more information see:

`http://www.FreeProtocols.org`

Open-Source Software Implementations. These are being made available for a wide variety of of platforms and end-user devices, including: pagers and cell-phones; hand-held PCs (Windows CE, Palm PC) and Palm Pilot; Windows 98, Windows 95, and Windows NT; Pine (UNIX, Windows, DOS). For complete details see:

`http://www.MailMeAnywhere.org`

Free Subscriber Services. These are provided to support initial deployment of the protocols in end-user devices. For complete details see:

`http://www.ByName.net` and
`http://www.ByNumber.net`

Collectively, the above components represent a complete recipe for the success of the LEAP protocols. All the pieces of the puzzle are complete, and there are no missing pieces.

1.9 How to Participate

As noted above, the LEAP protocols are entirely open, and any interested person or organization may participate in their development. To participate in the development of the LEAP protocols in general, visit the LEAP Forum website at http://www.leapforum.org/. To participate in the development of specific members of the LEAP family of protocols, visit the ESRO.org website at http://www.esro.org/, or the EMSD.org website at http://www.emsd.org/.

All of the above websites host mailing lists for commentary and general information exchange regarding the protocols. In particular, ESRO.org and EMSD.org host Working Group mailing lists for active development of their respective protocols.

In addition, we invite participation in the development of *The LEAP Manifesto* itself. We expect that as the LEAP family of protocols grows and becomes implemented on additional platforms, additional articles will be included in the Manifesto. Any person or organization may submit information or articles that they feel are appropriate for inclusion in the Manifesto; any such material will be given due consideration by the Manifesto editor.

In addition, we would welcome the translation of key Manifesto articles into foreign languages. One such translation has already taken place; the Manifesto article *The WAP Trap* is now available in French under the title *Le WAP a la Trappe*. Other key articles that would be greatly desirable in foreign language translations include *LEAP: One Alternative to WAP*, and *Operation WhiteBerry*. Persons interested in writing foreign language translations are asked to contact the Manifesto editor at info@leapforum.org.

We also invite general commentary and criticism of the Manifesto. Please let us know of any errors, omissions or ambiguities you may find in the Manifesto. Any input or commentary should be submitted to the Manifesto editor at info@leapforum.org.

1.10 Who We Are

Throughout the Manifesto, we frequently refer to ourselves in the first person, and we also refer to several organizations and domains that are in some way related to the LEAP protocols. The question may be asked, who exactly are "we"? Who are the authors of the Manifesto, and what is their relationship to the organizations involved in the development of LEAP? Who owns LEAP? In this section we provide the answers to these questions.

Mohsen Banan. Mohsen Banan is the principal editor of *The LEAP Manifesto*; he is also the author of many of its component articles. Several other authors also wrote and/or contributed material to certain component articles; these are acknowledged in the appropriate articles. First-person references throughout the Manifesto refer to the principal editor, Mr. Banan.

Mr. Banan is also the president of Neda Communications, Inc. He is also the president and a board member of the Free Protocols Foundation.

Neda Communications, Inc. Neda Communications, Inc. is a private, for-profit company located in Bellevue, WA. Neda provides consulting services and develops products and services relating to wireless data communications.

Neda has independently led the development of the LEAP protocol specifications since 1997. Neda has also developed a comprehensive set of software implementations of the LEAP protocols, which it intends to subject to the GNU Public License and make freely available.

The LEAP Protocols. The design and development of the LEAP protocols was primarily carried out by several engineers working at Neda Communications, Inc. The development effort was led and coordinated by Mohsen Banan. RFC-2188 was published jointly by Neda and AT&T personnel. RFC 2524 was published individually by Mohsen Banan. As the primary author of both RFCs, patent-free declarations for both protocols were made by Mohsen Banan and on behalf of Neda.

No one owns the LEAP protocols. The protocol specifications reside entirely in the public domain.

The LEAP Forum. The LEAP Forum is a clearing house for information and pointers relating to the LEAP protocols. The LEAP Forum is not a standards organization, it is not a legal entity of any kind, and it is not a membership organization. The LEAP Forum maintains a mailing list for the free interchange of information and commentary regarding the LEAP protocols. Any interested person or organization may subscribe to the mailing list. The LEAP Forum website and mailing list are presently hosted by Neda equipment and network resources, and managed by Neda personnel.

For more information, visit the LEAP Forum website at http://www.leapforum.org/.

ESRO.org and EMSD.org. ESRO.org and EMSD.org are open organizations for the development and maintenance of the ESRO and EMSD protocols respectively. Neither organization is a standards organization, nor a legal entity of any kind, nor a membership organization. They are simply forums to allow information exchange and cooperative effort relating to the LEAP protocols and technology.

Both organizations maintain several mailing lists, to which any interested person or organization may subscribe. The ESRO and EMSD websites and mailing lists are presently hosted by Neda equipment and network resources, and managed by Neda personnel.

In particular, each organization hosts a Working Group mailing list for active development of the corresponding protocol. Mohsen Banan is the current chairperson of both Working Groups, with responsibility for coordinating the Working Group development effort.

For complete information, visit the appropriate website at either
http://www.esro.org/ or http://www.emsd.org/.

Free Protocols Foundation. The Free Protocols Foundation is a non-profit organization whose mission is to prevent the inclusion of patented components within protocols. The FPF has established a set of policies and procedures for protocol development that is designed to ensure that the resulting protocol is patent-free. The LEAP protocols conform fully to these policies and procedures. Free Protocols Foundation board members include Mohsen Banan and Richard Stallman.

For more information see the Free Protocols Foundation website at http://www.FreeProtocols.org.

1.11 About The LEAP Manifesto

The purpose of The LEAP Manifesto is to provide a complete description of the LEAP protocols and their intended role in the development of the Mobile Messaging industry. The Manifesto includes:

- An overview of the Mobile Messaging industry, and a description of the essential factors that are required for its long term success and growth.
- A technical description of the LEAP protocols themselves.
- A description of the process used to develop the LEAP protocols, and how and why this differs from the conventional development process.
- Technical descriptions of key aspects of the LEAP protocols, including their efficiency, and their implementation on Windows CE devices and Palm OS devices.
- An analysis of several closed Mobile Messaging solutions (e.g. WAP), and a description of LEAP's superiority to these closed solutions.
- A description of our strategy for encouraging widespread usage of the LEAP protocols, including the distribution of open-source software implementations of the protocols, and the availability of free subscriber services.

1.11.1 Manifesto Organization

The LEAP Manifesto is organized as a series of largely independent articles. Each of these articles stands on its own, and can be read and understood independently of the others. Together, these articles provide a complete picture of the Mobile Messaging industry and the role of the LEAP protocols. Since each article is intended to be self-contained, some material is duplicated in more than one article.

The LEAP Manifesto consists of the following articles:

- **Executive Summary.** An overview summary of the entire LEAP Manifesto. The Executive Summary provides a brief description of all the major elements of the manifesto.
  First Published: 2000/8/4 Last Updated: 2000/12/5
  Article formats: [HTML] [PDF] [PS] [Text Only]
- **Part I: The LEAP Protocols**
  - **Overview of the LEAP Protocols.** A general overview description of the LEAP protocols.
    First Published: August 4, 2000
    Last Updated: August 8, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **The LEAP Protocol Development Model.** A description of the processes used to develop the LEAP protocols, and how and why these differ from conventional development processes. This article also includes a criticism of the IETF protocol development processes.
    First Published: August 4, 2000

Last Updated: June 16, 2000
Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

  - **Free Protocols Foundation Policies and Procedures** A description of the Free Protocols Foundations processses to ensure the development and maintenance of patent-free protocols.
    First Published: March 29, 2000
    Last Updated: June 26, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **ESRO: A Foundation for the Development of Efficient Protocols.** A technical description of ESRO, the transport mechanism component of LEAP.
    First Published: August 4, 2000
    Last Updated: August 9, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **EMSD: The LEAP E-Mail Component.** A technical description of EMSD, the e-mail component of LEAP.
    First Published: August 4, 2000
    Last Updated: July 14, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **Efficiency of EMSD.** A technical paper analyzing the efficiency characteristics of EMSD and comparing its efficiency to other e-mail protocols.
    First Published: October 23, 1996
    Last Updated: August 16, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **A Brief History of LEAP.** A summary of the major events in the evolution of the LEAP protocols.
    First Published: August 4, 2000
    Last Updated: September 20, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **The Future of LEAP.** A description of the planned future development of LEAP, including descriptions of several LEAP-based products and services which are currently under development.
    First Published: August 4, 2000
    Last Updated: June 14, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

- **Part II: LEAPing Over Closed Solutions**
  - **The WAP Trap.** A detailed criticism of a set of specifications called the Wireless Application Protocol, or WAP. This article demonstrates that WAP is entirely unfit to play the role of a Mobile Messaging industry standard.
    First Published: April 3, 2000
    Last Updated: May 26, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **LEAP: One Alternative to WAP.** A point-by-point comparison of the LEAP protocols to the WAP specifications. This article demonstrates that LEAP has all the desirable characteristics of an industry standard protocol that WAP lacks.
    First Published: August 4, 2000
    Last Updated: December 6, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **WAP Scraps.** A discussion of what can be salvaged from what remains of WAP.
    First Published: August 28, 2001
    Last Updated: August 28, 2001
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

  - **Operation Whiteberry.** A description of how equivalent functionality to the closed BlackBerry mobile messaging solution can be implemented based on a completely open model, using existing open-source software implementations of LEAP, and existing off-the-shelf hardware components.
    First Published: February 27, 2001
    Last Updated: November 3, 2002
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

- **Part III: Making LEAP Widespread**

  - **Strategy for Making LEAP Widespread.** A description of our strategy for encouraging widespread usage of the LEAP protocols, including the distribution of open-source software implementations of the protocols, and the availability of free subscriber services.
    First Published: August 4, 2000
    Last Updated: August 8, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **EMSD on Windows CE.** A technical paper describing the architecture and implementation of EMSD on Windows CE devices.
    First Published: March 3, 1997
    Last Updated: August 16, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **LEAP on Palm OS.** A technical paper describing the architecture and implementation of LEAP on Palm OS devices.
    First Published: September 27, 2001
    Last Updated: September 27, 2001
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **LEAP in JAVA.** A technical paper describing the architecture and implementation of LEAP in JAVA.
    First Published: February 4, 2003
    Last Updated: February 4, 2003
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

    LEAP on Linux Based PDAs. A technical paper describing the architecture and implementation of LEAP on Linux Based PDAs.
    First Published: September 27, 2001
    Last Updated: September 27, 2001
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **Trying out LEAP.** A step-by-step, hands-on demonstration of how the LEAP protocols can be used to turn any Windows CE device into a fully functional Mobile Messaging device.
    First Published: June 12, 1998
    Last Updated: June 12, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **WhiteBerry and Bluetooth.** A description of how WhiteBerry and Bluetooth can be used in combination to bring new and enhanced messaging capabilities to the mobile professional.
    First Published: July 27, 2001
    Last Updated: July 31, 2001
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]
  - **Use of EMSD for Mail Notification.** A description of how EMSD can be used to provide a general Mail Notification service.
    First Published: TBD
    Last Updated: TBD
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

  - **Lessons From History: Comparative Case Studies.** An analysis of the factors which lead to the success or failure of protocols, including discussions of several historical case studies.
    First Published: August 4, 2000
    Last Updated: July 7, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

- **Part IV: The Mobile Messaging Industry**

  - **The Mobile Messaging Industry.** An overview of the Mobile Messaging industry, and a description of the essential factors that are required for its long term success and growth.
    First Published: August 4, 2000
    Last Updated: August 10, 2000
    Article formats: [ONE-HTML] [SPLIT-HTML] [PDF] [PS] [Text Only]

1.11.2 Draft Articles

The LEAP Manifesto is a work in progress, and various additional articles are planned for future inclusion in the Manifesto.

Some of these future articles already exist in draft form, and are available for review in the Draft Documents section of the LEAP Forum website at http://www.leapforum.org/draft-leapManifesto/. As these and other articles are completed, they will be incorporated into the Manifesto.

1.11.3 Getting the Manifesto

The LEAP Manifesto and all of its component articles are available in multiple formats, including HTML, PDF, PostScript, and plain text. You can view or download the Manifesto in any of these formats from the LEAP Forum website at http://www.LeapForum.org/leap/index.html. The LEAP Manifesto is also available at the Free Protocols Foundation website at http://www.FreeProtocols.org/leap/index.html.

Part I

The LEAP Protocols

Chapter 2

Overview of the LEAP Protocols

2.1 Introduction

The key component of the Manifesto is a set of mobile messaging protocols called the **Lightweight & Efficient Application Protocols**, or **LEAP**. LEAP is a set of high-performance, efficient protocols which are ideal for mobile and wireless applications. This article provides a brief overview of the LEAP protocols; complete details are provided elsewhere in *The LEAP Manifesto* [**?**].

2.2 The Need for Efficiency

Engineering is the art of making intelligent trade-offs between conflicting requirements. A perennial engineering trade-off is that which must be made between the need for *simplicity*, and the need for *performance*. In the case of wireless data communications, performance means such things as data transfer speed, power efficiency, and bandwidth efficiency.

The 1980s and 1990s were the decades of simple protocols - protocols such as the very aptly named Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP). A great deal of the success of these and other Internet protocols can be attributed to their simplicity.

The first generation of network engineers and network operators were only able to view network communications in relatively simple terms. It was appropriate to cater to that simplicity with simple protocols. A key reason for the success of these early protocols is the lack of technical sophistication on the part of first-generation network engineers and operators.

Simple protocols are easier to make widespread than "good" protocols (meaning those which have better capabilities and performance), for the basic reason that network engineers and operators are able to adopt and implement simple protocols much more easily than "good" protocols.

However, things have changed. Network communications has now expanded into the wireless and mobile data communications arena, and *wireless applications demand efficiency*. The move to wide-area wireless has significantly shifted the location of the ideal engineering balance between simplicity and performance - moving it away from simplicity, and towards performance.

Wireless networks are constrained by bandwidth limitations, and the hand-held devices they serve are constrained by limitations such as display size, battery capacity, and memory capacity. These constraints place an extremely high premium on the efficiency of data transfer.

Existing Internet protocols do not provide the required efficiency. We therefore need a new generation of high-performance, efficient protocols, to cater to the demands of wireless applications. The point is sometimes made that the need for efficiency in the wireless arena is a temporary one – that advances in wireless engineering technology in the form of third generation (3G) systems will eliminate existing bandwidth limitations, obviating the need for efficient protocols. As long as the capacity of wireless networks remains finite, however, the need for efficiency will persist. Efficient usage is an inherent requirement for any finite resource, therefore the requirement for efficient bandwidth usage and battery longevity will remain.

2.3 Technical Overview of LEAP

The LEAP protocols are intended to be an enabling catalyst for the growth of the wireless-IP based Mobile Messaging industry, and have been designed with this goal in mind from the outset. They have been designed as a genuine enabling technology which will bring enormous benefits to the industry and the consumer. They are a sound engineering construction based on true openness and patent-freedom.

The LEAP protocols a general-purpose solution to the problem of efficient message transfer, and their use is not limited to any particular device type or network. In particular, LEAP is compatible with all wireless-IP networks. Examples of wireless networks which provide native support for LEAP are CDPD, GSM, packet CDMA, and PCS.

The basic organization of the LEAP protocols is shown in Figure 2.1.

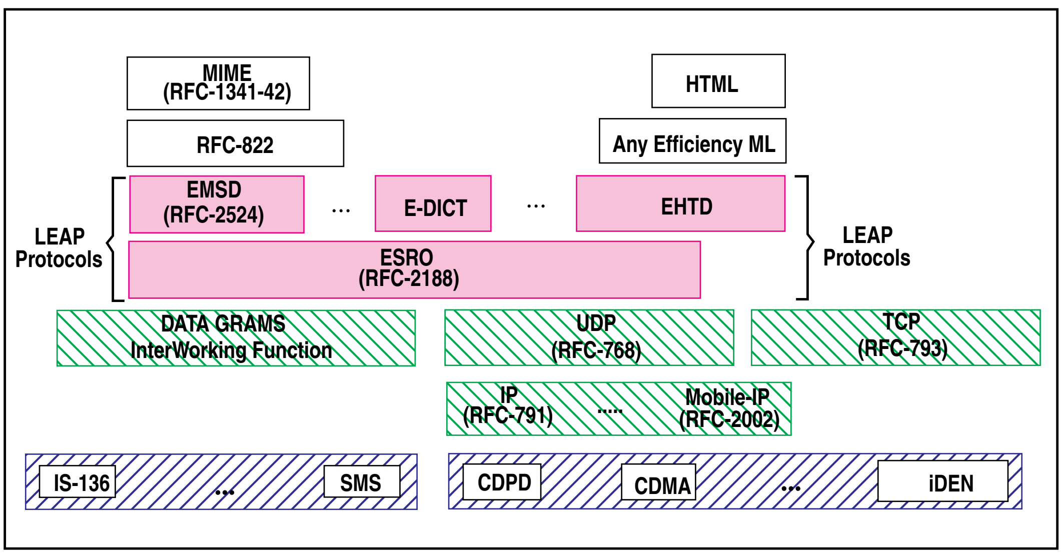


Figure 2.1: LEAP Protocol Organization

2.3.1 The ESRO Layer: Efficient Transport Services

As shown in Figure 2.1, the LEAP protocols are layered. The lower layer is called Efficient Short Remote Operations, or ESRO. The ESRO layer provides reliable connectionless transport services which can be used for a variety of applications. For example, in addition to mobile messaging services, ESRO can also be used as a transport service for credit card verification applications and efficient micro browsers.

For more information on ESRO see the article *ESRO: A Foundation for the Development of Efficient Protocols* within *The LEAP Manifesto*, or visit the ESRO website at http://www.esro.org/.

2.3.2 The EMSD Layer: Efficient E-Mail

One of the efficient application layers built on top of ESRO is called Efficient Mail Submission & Delivery, or EMSD. EMSD is the component of LEAP that addresses the Mobile Messaging application.

EMSD is a specialized native Internet messaging protocol. It defines a similar set of services to the existing SMTP protocols. It defines a complete set of rules for message submission (end-user device to server) and message delivery (server to end-user device). EMSD meets or exceeds the level of functionality, reliability and security provided by the existing SMTP protocols.

Though its use is not limited to wireless networks, EMSD has been designed specifically to address the requirements of wireless networks, such as CDPD, Wireless-IP, Mobile-IP. In particular, EMSD has been designed with a very strong and clear emphasis on efficiency.

EMSD is highly optimized for the submission and delivery of short (typically 4 kilobytes or less) Internet e-mail messages, and is therefore extremely well suited to the wireless environment. EMSD improves on existing messaging protocols by optimizing the exchange between the server and the end-user device, both in terms of the number of bytes transferred and the number of transmissions. Because of the required timeliness of the messages, mailbox access protocols like POP and IMAP are not used. EMSD is the only truly open messaging protocol that is specifically designed for the wireless network environment.

EMSD is a natural extension of the existing Internet e-mail environment, and accommodates the two-way paging model of usage, in which time-critical messages are ”pushed” to the recipient.

Any network or network operator which faces significant bandwidth and capacity limitations can benefit from the use of EMSD. Any user of a network who must bear high costs for measured network usage can benefit from the use of EMSD.

The initial use of EMSD is expected to be primarily to provide Mobile Messaging services over IP-based wireless networks. However, EMSD can also function as an adjunct to Mail Access Protocols for ”Mail Notification Services.”

For more information on EMSD see the article *EMSD: The LEAP E-Mail Component* within *The LEAP Manifesto*, or visit the EMSD website at http://www.emsd.org/.

2.3.3 The EHTD Layer: Efficient Web Browsing

The Efficient Hyper Text Delivery (EHTD) layer is a hypertext transfer protocol which is optimized for the efficient transfer of short markup pages. EHTD is the component of the LEAP protocols which facilitates web browsing. Along with EMSD, EHTD also benefits from the reliable efficient services of ESRO. A multiplicity of efficient markup languages can be used in conjunction with EHTD. Development of the EHTD protocol is currently in progress.

2.3.4 Other Efficient LEAP Applications

Various other efficient application protocols are either under development, or anticipated for future development. One of these is the Efficient Dictionary protocol, or E-DICT, which will enable efficient access to dictionaries and other look-up data structures. A starting point for the E-DICT protocol is currently being created. In developing E-DICT, we intend to build on the existing work already done in the context of the DICT protocol.

We anticipate that additional protocols will be needed for a variety of future applications, not all of which can be foreseen at this time. These applications will include such things as efficient implementations of ESRO-based instant messaging, chat, white pages, and others.

2.4 Efficiency Characteristics of LEAP

All LEAP protocols are designed with efficiency in mind. In this section we describe the efficiency characteristics of EMSD, the LEAP e-mail protocol. Other LEAP protocols deliver similar efficiency benefits.

Most existing Internet e-mail protocols are designed with simplicity, and continuity with SMTP traditions, as two of the primary design requirements. These requirements are in conflict with efficiency of data transfer, and for this reason most existing Internet e-mail protocols are not efficient.

EMSD, on the other hand, has been designed with efficiency as its primary requirement. For this reason, EMSD is a great deal more efficient than existing Internet e-mail protocols.

A detailed efficiency study of the LEAP protocols is provided in the article entitled *Efficiency of EMSD* [**?**] within *The LEAP Manifesto*. That article presents various efficiency studies which compare the efficiency of EMSD to other e-mail protocols such as SMTP, POP and IMAP, and which demonstrate the efficiency advantages of EMSD.

In this section we provide a brief summary of EMSD's efficiency characteristics. A comparison of the efficiency of the EMSD protocol to other messaging protocols is provided in Figure 7.3, which shows the delivery traffic overhead for EMSD and three other e-mail protocols: SMTP, IMAP and POP.

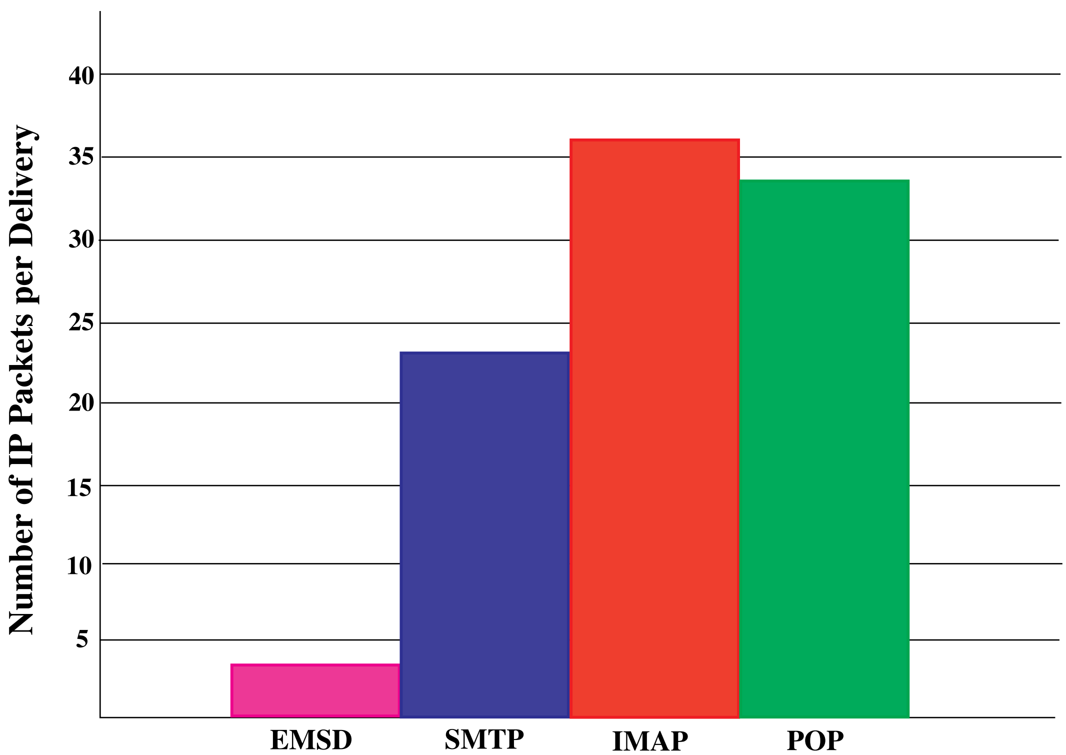


Figure 2.2: Protocol Efficiency Comparison

As the figure shows, EMSD is much more efficient than SMTP, POP and IMAP. For submission and delivery of short e-mail messages, EMSD is up to five times more efficient than the ubiquitous SMTP e-mail messaging protocols, both in terms of the number of packets transmitted, and in terms of number of bytes transmitted. Even with pipelining and other possible optimizations of SMTP, EMSD remains up to three times more efficient than SMTP, both in terms of the number of packets transmitted, and in terms of number of bytes transmitted.

By minimizing the network traffic required to send and receive messages, EMSD meets the needs of the mobile communicator. The extreme efficiency of the EMSD protocol translates into bandwidth efficiency, which in turn translates into:

- Efficient use of carrier bandwidth, and therefore increased capacity for network operators
- Longer battery life for mobile phones, PDAs and other wireless Internet devices
- Cheaper network usage costs for the end-user
- Reduced latency for the end-user
- Improved support for marginal coverage areas

2.5 LEAP: A Basis for Convergence

An illustration of how LEAP works is shown in Figure 22.4. As the figure shows, LEAP provides complete openness of interoperability among Mobile Messaging devices, message centers, and wireless networks.

LEAP will thus have the effect of unifying the entire Mobile Messaging industry under a set of open Internet Protocol (”IP”) standards and protocols so that, in the manner of the World Wide Web, all of the Mobile Messaging networks will effectively operate as one.

In order to achieve this convergence, it is not sufficient for the Mobile Messaging industry merely to adopt a set of common protocols. Many would claim that WAP is in fact just such a set of common protocols. However, a further essential attribute of the required protocols is that they must be a truly integral, “end-to-end” part of the Internet, as opposed to “gateways” which accommodate unnecessary gatekeepers and middlemen.

LEAP is based on the concept of the Internet end-to-end model, in which direct communication between the client and the server assumes that the role of the network service provider is merely that of a pipe – i.e. a passive communication conduit. The Internet end-to-end model assumes that both ends are under the control and choice of the user, and that the servers are widespread, from a variety of providers, and under no specific administration or control. The Internet end-to-end model is in sharp contrast to the traditional phone company and telecommunications approach, which inserts gateways between the two ends, and creates control and exploitation opportunities for the telecommunication operators.

Bearing in mind that the natural convergence of all wireless networks to IP at Layer 3 is well under way and rapidly progressing, the key remaining requirements are: efficiency, lightweightness, miniaturization, and conformance to the Internet end-to-end model. LEAP fulfils all of these requirements. By serving as the necessary missing link, LEAP will become the ultimate basis for convergence.

The mobile e-mail component of LEAP is EMSD. In the spirit of the Internet end-to-end model, the EMSD protocol will facilitate the convergence of the IP-based two-way paging industry, and Internet e-mail, in a natural and transparent manner.

2.6 The End-User’s Experience

The entire LEAP family of protocols bring efficiency and functionality benefits to the user of miniaturized mobile devices. In this section we describe the user’s experience of an EMSD-enabled device.

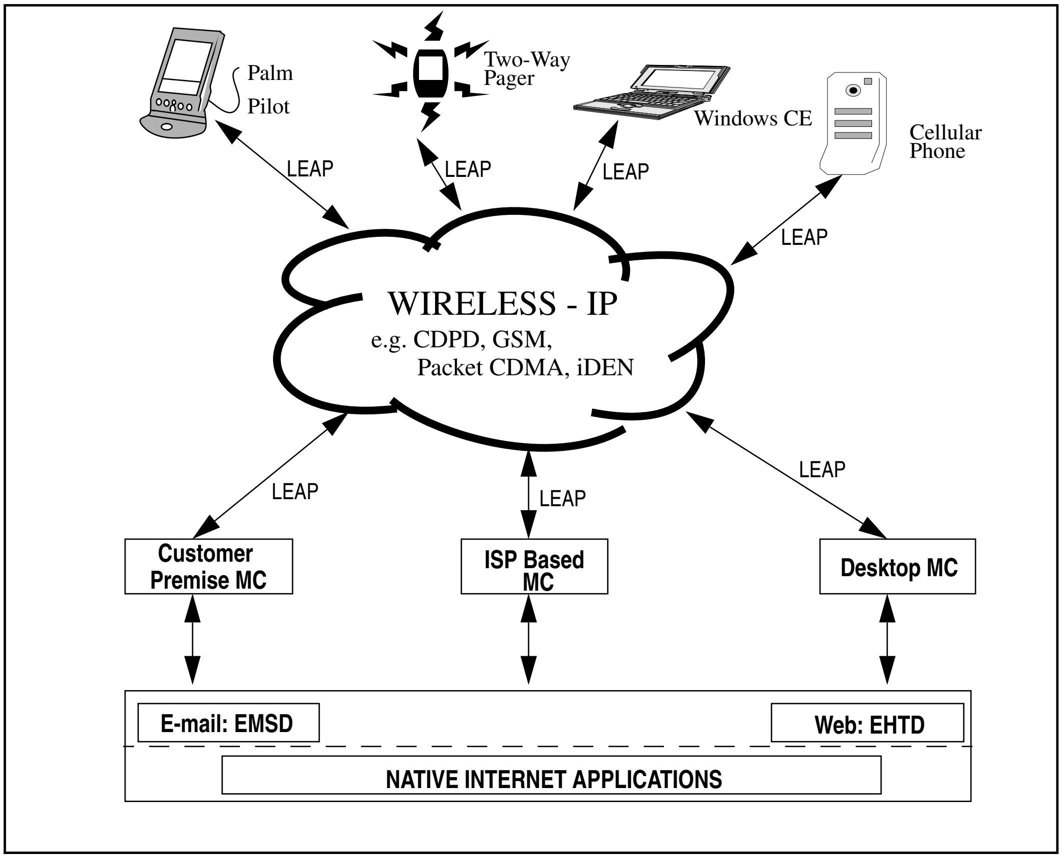


Figure 2.3: Open Mobile Messaging

Mobile users may not always have the benefit of a wired connection, because of their frequent mobility. They may have a permanent computing system elsewhere, at which they can review large messages at their leisure (for example, messages containing Word documents, Excel spreadsheets, images, etc.). While on the move, however, they need to be kept apprised of important information that requires their immediate attention. Such information cannot wait for them to find the time to set up a laptop and dial in to check for messages. They must be able to accept messages immediately, at any time, and on a device that they can carry anywhere.

The experience of the end-user in using LEAP-based Mobile Messaging technology is illustrated in Figure 13.2.

The user equips him/herself with an EMSD device. The EMSD device could be a dedicated two-way pager, or a hand-held device (such as a PalmPC) with a wireless (for example CDPD) modem. While the device can be turned off, the modem will remain on at all times to accept incoming messages.

Anyone with access to the Internet can now send a message to this user. The EMSD Service Provider accepts the message from the Internet e-mail system via standard Internet protocols, then delivers the message to the user's device via EMSD protocols. Since the modem is always on, the message can be accepted at any time, and the user can be notified immediately (in any of the ways commonly used for pager notification) that a message has arrived. The user will then activate the EMSD device and read the message.

To send a message the user enters the message, then submits it to the EMSD Service Provider via the EMSD protocols. The Service Provider then acts like a standard Internet Service Provider and sends the message to its destination.

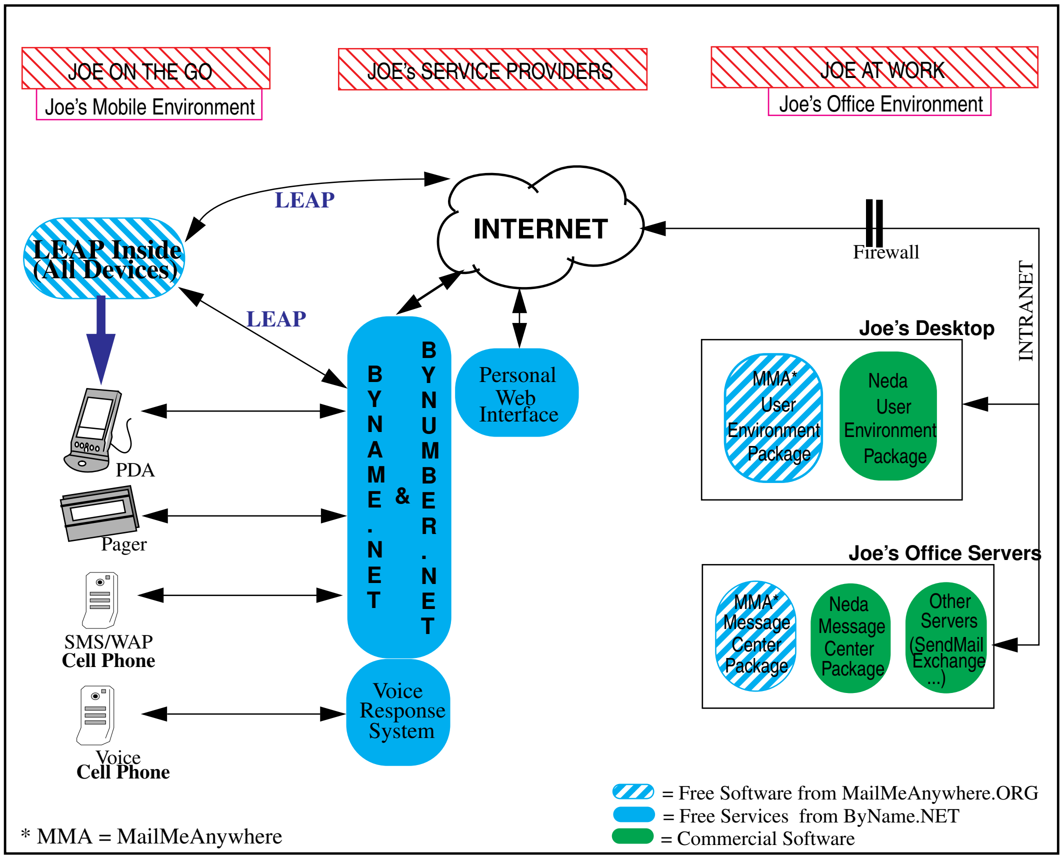


Figure 2.4: The End-User's Experience

The end-user device may have a limited display area and a limited keyboard. This is very much the case for today's cell phones, for example. If so, both the end-user and his/her correspondents may wish to make use of canned messages to facilitate their communication. These canned messages may be defined by the system or end-user device, or they may defined by the message originator as embedded multiple-choice responses.

Figure 13.2 illustrates how the Mobile Messaging needs of a typical user (we'll call him Joe) are provided by the LEAP technology. This figure includes all the required technological components, and shows how they interoperate to satisfy Joe's needs. The figure includes three major components:

1. Joe requires some form of handheld mobile device, such as a cell phone or a PDA. This component is shown on the left side of the figure. The device must include the appropriate LEAP device software, allowing it to use the LEAP protocols to communicate with LEAP-enabled Message Centers, either directly over the Internet, or via a Subscriber Service system.

2. Joe requires a set of Subscriber Services to support his Mobile Messaging capability. This component is shown in the center of the figure.

3. Joe may also wish to have LEAP-based Mobile Messaging capability on a Personal Desktop system at home, or on a Corporate Intranet system at his office. These components are shown on the right side of the figure.

If Joe receives a generic (i.e. non-LEAP) e-mail message over the Internet, then this will be fielded by his Subscriber Service provider, then forwarded to Joe's mobile device using the LEAP protocols.

Meanwhile, e-mails for Joe may be received in either his home or office mailbox systems. Joe may configure either of these systems to forward certain e-mails to his mobile device on a selective basis. If so, the qualifying e-mails will be forwarded to him directly over the Internet, using the LEAP protocols. The Subscriber Services system need not be involved in the transmission of these forwarded e-mails, since they are being sent from one LEAP-enabled system to another.

In summary, the end-user experience described above represents a superset of the capabilities of the RIM BlackBerry [tm] system. The market success of BlackBerry clearly demonstrates the large user demand for this kind of service. By providing the same functionality of BlackBerry in a completely open fashion, the benefits to the consumer will be that much greater. For further discussion, see the article *Operation WhiteBerry* in *The LEAP Manifesto*.

2.7 The LEAP Development Process

The LEAP protocols are intended to be open in the fullest sense of the word; they are intended to be freely and permanently available, subject to public review and revision, and without usage restrictions of any kind. Therefore the processes and procedures used throughout the development and maintenance of the LEAP protocols have been such as to endow them with these characteristics, and to ensure their integrity as public protocols.

A detailed description of the LEAP development process is provided in the article entitled *The LEAP Protocol Development Model* within *The LEAP Manifesto*. In the following sections we provide a brief summary of the major development principles.

2.7.1 Patent-Freedom

The development and maintenance of the LEAP protocols conforms fully to the policies and procedures of the Free Protocols Foundation. In particular, Neda has declared to the Free Protocols Foundation that the LEAP protocols are patent-free to the best of its knowledge, and that it intends to keep them patent-free permanently. For more information see `http://www.FreeProtocols.org`.

2.7.2 RFC Publication

Both protocols have been published as Internet RFCs; ESRO in September 1997 as RFC-2188 [**?**], and EMSD in March 1999 as RFC-2524 [**?**]. RFC publication is the mainstream Internet publishing procedure, ensuring that the protocols are freely, easily and permanently accessible to anyone who wishes to use them.

2.7.3 Open Maintenance Organizations

To provide an open forum for the continued development and maintenance of the LEAP protocols, Neda has established a public organization for each protocol.

The ESRO and EMSD protocols are maintained, respectively, by ESRO.org at http://www.esro.org/, and by EMSD.org at http://www.emsd.org/.

Each of these organizations allows public review of the respective protocol, and provides mechanisms for enhancement of the protocol as a result of collective experience.

Any interested person may participate in the further development of the protocols. Participation in the development process is entirely open and non-exclusive; there are no membership fees.

WAP	LEAP
Subject to known patent restrictions	Patent-free
Self-published by the WAP Forum	Published as Internet RFCs
Revisions subject to change without notice	All revisions permanently fixed
Maintained by the WAP Forum	Maintained by open working groups
Re-invention of existing protocols	Efficiency-optimizing extensions to existing protocols
Tailored to mobile phone user interface characteristics	User interface independent
Inherent security vulnerability	Imposes no security assumptions
Inconsistent protocol number assignment	Consistent protocol number assignment
Poor technical design	Good technical design
Initial focus: web browsing	Initial focus: messaging
Treats wireless as a special case	Treats wireless as an extension of Internet

Table 2.1: WAP versus LEAP

2.8 LEAPing over WAP

A set of specifications called the **Wireless Application Protocol**, or **WAP**, exists already, and purports to do the same things that LEAP does. However, the WAP specifications are entirely unfit for their claimed purpose, and are doomed to technological and political failure. A detailed criticism of WAP and justification of these statements is provided in an article called *The WAP Trap* [**?**] within *The LEAP Manifesto*.

LEAP is an alternative to WAP, that does in fact what WAP does only in fiction. For a point-by-point comparison of LEAP to WAP, see the article entitled *LEAP: One Alternative to WAP* [**?**] within *The LEAP Manifesto*.

Those characteristics of WAP that make it wholly unfit to be the industry standard are summarized in Table 11.1, along with the corresponding characteristics of the LEAP protocols.

2.9 A Brief History of LEAP

LEAP originated in 1994 as part of the research and development initiatives of McCaw Cellular's wireless data group (now AT&T Wireless Services). The development work that would eventually lead to LEAP was initially undertaken in the context of the CDPD network; its scope was later expanded to include the Narrowband PCS network also.

By 1996 McCaw Cellular was fully committed to paging, had recently purchased two nationwide narrowband wireless PCS licenses, and wished to develop an efficient wireless message transport and delivery system. Neda Communications, Inc., an independent consulting company working under contract to McCaw Cellular, played a significant role in the development of the required system. Neda Communications had also been involved from the outset in the development of the CDPD specification.

In 1997 however, soon after the purchase of McCaw Cellular by AT&T Wireless, the latter company abandoned the wireless messaging project. Prior to this event, Neda had secured from AT&T the necessary rights to continue independent development of the protocols. Therefore, recognizing the eventual future need for these protocols, Neda then undertook to continue development of them independently of AT&T. They were eventually completed by Neda, published as RFCs, and now form the basis of the LEAP protocols.

Prior to abandoning wireless messaging, AT&T Wireless Services invested several million dollars in related development work. In creating LEAP, therefore, Neda was able to build upon a large abandoned investment by AT&T

Wireless.

Chapter 3

The LEAP Protocol Development Model

3.1 Introduction

Protocols come in all shapes and sizes, and from a variety of sources. Some are proprietary, intended for use exclusively by their developer. Others may be "open" in some sense, indicating that they are intended for more general, public usage. In this context, the word "open" can mean any one of several different things. It may mean nothing more than that the protocol has been published by its developer. The protocol may still be very tightly controlled: revision of the protocol may remain the exclusive right of the developer, the protocol may be protected by patent or copyright restrictions, and use of the protocol may require a licensing agreement. This is a very narrow, and to our mind misleading, use of the word "open."

At the other extreme, the protocol may be open to a very high degree of public accessibility: it can be published by an open mechanism such as RFC publication, undergo revision by means of public working groups, and be entirely free of usage restrictions. A protocol satisfying all these criteria can be said to be "open" in the broadest sense. Protocols are often referred to as "open" to imply that they are open in a broad sense, whereas in fact they are open only in the narrowest sense.

To a large extent, the character of a protocol is defined by the processes used to develop it. This article is about a particular set of protocols called called **LEAP**, or the **Lightweight & Efficient Application Protocols**. LEAP is a set of high-performance, efficient protocols intended for mobile and wireless applications.

The LEAP protocols are intended to be open in the absolutely broadest sense of the word; they are intended to be freely and permanently available, subject to public review and revision, and without usage restrictions of any kind. The processes used to develop the LEAP protocols have been such as to ensure that they have these intended characteristics.

In this article we describe the LEAP development process. In the next section we provide a general description of the various phases of development that a protocol may go through. Then in subsequent sections we describe the specific processes used for the LEAP protocols at each phase of development.

In this article we also discuss the enormous economic influences that protocols can exert, and we point out the potential for corruption that inevitably accompanies this. We describe our general philosophy regarding protocol development, and we present what we consider to be the key principles which must be upheld in order to maintain protocol integrity in the face of corrupting influences. Finally, we provide justification for some of the protocol development choices we have made which others may consider to be unconventional or controversial.

3.2 Protocol Phases of Development

Over its lifetime, a protocol typically goes through a number of developmental phases. In general, from conception to decease, a protocol may go through some or all of the following phases:

1. **Initial Development**. A protocol may be initially developed in a variety of ways, e.g. by a standards organization, a private business, or the academic community.
2. **Global Parameter Assignment**. If necessary, global parameters must be assigned to the protocol, for example by the IANA.
3. **Publication**. If the protocol is intended for public usage, as opposed to exclusive in-house usage, then it must be made publicly available, for example by being published as an Internet RFC.
4. **Patent-Freedom**. If the developers of a protocol intend it to be patent-free rather than proprietary, then they must take appropriate steps to work towards a patent-free result.
5. **Industry Usage**. The ultimate test of a protocol is whether or not it becomes widely accepted and implemented in the industry at large. This is an aspect of protocol evolution which is not under the direct control of its developer.
6. **Maintenance and Enhancement**. Protocols frequently undergo revision and enhancement as a result of experience and/or changing industry requirements. Depending on the intended character of the protocol, this may take place by closed and proprietary processes, or by open and public ones.
7. **Endorsement by a Standards Body**. Once a protocol has become accepted as an industry standard, it may receive the formal sanction of a standards body.

Depending on its purpose, nature, and history, a protocol may undergo some, all, or none of these phases. Also, a protocol may iterate through phases (3) to (7) multiple times, as it undergoes maturation via repeated revision and re-publication.

In general, the developers of a protocol define and control the policy and processes to be applied to the protocol at every phase of development except (5) and (7). The developers have no direct control of (5) at all, and only minimal influence over (7).

The processes used at each phase of development determine the character of the resulting protocol. In the following sections of this article we will describe the processes applied to the LEAP protocols at each phase of development except (5).

3.2.1 Initial Protocol Development

The conception and early development of a protocol can take place in a variety of ways. A traditional source of Internet protocols is the IETF/IESG. Other sources of protocols are private businesses, the academic community, or even a single individual.

In the case of LEAP, initial development of the protocols was carried out by Neda Communications, Inc., a private business.

We believe that there is a tendency among established standards bodies to regard their own, officially sanctioned protocols as authoritative, while any other protocol is of questionable worth or validity. However, the history of protocol development does not support this view. Small groups of individuals have created protocols with far-reaching consequences (e.g. the World Wide Web), just as established standards bodies have created protocols which failed to achieved industry acceptance. For this reason we do not regard any one source of protocols as necessarily superior to any other.

3.2.2 Global Parameter Assignment

It may be necessary to assign global parameters to a protocol.

In the case of LEAP, the necessary UDP port assignments for ESRO and EMSD have been made through the IANA.

3.2.3 Protocol Publication

If a protocol is intended to be an open protocol, as opposed to an in-house or proprietary one, then it must be made publicly available. This can be accomplished in various ways; the protocol can be self-published by the developer, or it can be published through an independent agency such as the Internet RFC Editor.

In common with other Internet protocols, the LEAP protocols have been published in the form of Internet RFCs [**?**], [**?**]. RFC publication provides a number of significant advantages:

- **World-Wide Access.** RFC publication ensures world-wide access to the published protocol. There are numerous Web and FTP sites world-wide that carry the complete series of RFC documents – if a particular site is busy or unavailable, a person seeking an RFC can simply go to another.
- **Unrestricted Access.** A user may download an RFC publication completely freely, without incurring any legal restrictions whatsoever. All that is required to acquire the full text of an RFC is its number or title. There are no copying or other distribution restrictions attached to RFCs.
- **Permanence.** RFC publication is permanent. Even if the creator of a protocol should go out of business or otherwise become defunct, the RFC itself will remain in existence.
- **Stability.** Once published, an RFC is *fixed* – it cannot undergo change. If a new revision of the standard must be issued, this is handled by issuing the revision under a new RFC number.
- **Quality Assurance.** The RFC Editor maintains publication quality control, and will decline to publish a document which, in the Editor's opinion, falls below the required technical and/or editorial RFC standards.

For these reasons, RFC publication is the traditional, mainstream publication mechanism for Internet protocols, and virtually all Internet-related protocols are published this way. RFC publication of the LEAP protocols ensures that they are freely, easily and permanently accessible to anyone who wishes to use them.

3.2.4 Patent-Freedom

If a protocol is intended to be open rather than proprietary, then the protocol developers may take steps to work towards a patent-free result.

In the case of a public protocol, the presence of patented components within the protocol is very undesirable. A public protocol provides a well-defined interoperability specification for an industry, so that businesses and other organizations can create products and services independently, which can then be relied upon to function and interoperate correctly. The protocol is a positive, enabling force for the entire industry.

In order for the protocol to play this role, anyone who wishes to implement and use it must be able to do so freely. The presence of patented components within the protocol places restrictions on its usage, and therefore undermines this purpose. Furthermore, the presence of the patent brings an enormously unfair market advantage to the patent holder. By exercising his patent rights, the patent holder can gain financial benefit from competing companies who wish to use the protocol, or can stifle competition altogether by denying the competing company the right to use the protocol at all.

Software patents thus pose a significant danger to protocols. In some cases patents become included in protocols by accident – that is, without deliberate intentionality on the part of the protocol developer. In other cases, however, an unscrupulous company or organization may deliberately introduce patented components into a protocol, in an attempt to gain market advantage via ownership of the protocol.

In either case, the protocol can then be held hostage by the patent-holder, to the great disadvantage of anyone else who may wish to use it. Patented protocols thus have the effect of inhibiting free and open competition, and are therefore highly detrimental to the industry and the consumer.

The LEAP protocols are intended to be fully open and without usage restrictions of any kind. We have therefore exercised great diligence throughout their development to ensure their patent-freedom.

The Free Protocols Foundation

Various mechanisms exist for working towards a patent-free result. In developing the LEAP protocols, we have adopted a set of procedures defined for this purpose by the **Free Protocols Foundation**.

The Free Protocols Foundation, or FPF, is an independent, non-profit organization whose mission is to prevent the inclusion of patented components within protocols. The FPF has established a set of policies and procedures for protocol development that is designed to ensure, insofar as this is possible, that the resulting protocol is patent-free.

In general, there may be both an **Author** and a **Working Group** involved in the development of a protocol. The Author is the company, organization or other entity that has primary responsibility for the protocol. In some cases, the protocol may also undergo development at the hands of a **Working Group** – a set of independent individuals or companies who work cooperatively on the protocol, usually via public mailing lists. The FPF defines procedures for both Authors and Working Groups, allowing both to participate in the development of a protocol, while still maintaining the patent-free character of the protocol.

For example, both an Author and a set of Working Groups are involved in the development of LEAP. As noted above, the LEAP protocols were initially developed by Neda Communications, Inc., and this company continues to take primary responsibility for their maintenance. In addition, as described in Section 3.2.5, continued enhancement and revision of the protocols takes place by means of various Working Groups.

Both the Author and the Working Group may wish to make public declarations regarding the patent-freedom of the protocol. The Author may wish to make an initial declaration that the protocol is intended to be patent-free; and the Working Group may wish to make a declaration that it operates according to the FPF Working Group procedures, thereby maintaining the patent-free nature of the protocol.

In addition to defining protocol development procedures, the FPF also acts as an independent, public forum in which Authors and Working Groups may make these declarations. Such declarations which are submitted to the FPF are published on its website.

For complete information on the Free Protocols Foundation and its procedures, see the FPF website at http://www.FreeProtocols.org.

LEAP's Adherence to the FPF Procedures

Development of the LEAP protocols conforms in all regards to the processes and procedures of the Free Protocols Foundation. LEAP's compliance with these processes ensures, insofar as this is possible, that the LEAP protocols are, and will remain, patent-free.

We have adopted the FPF processes because they are available for use by any protocol developer, without the requirement that the developer be part of a formal standards organization. Various standards organizations include processes within their protocol development procedures which work towards patent-freedom. In general, however, these patent-free processes are created for the standards organization's internal use, and are not available for a protocol which is not officially sanctioned by the organization.

The FPF, on the other hand, is entirely egalitarian, providing the same level of support for patent-freedom to any protocol developer. To our knowledge, the FPF is unique in this regard.

Author’s Patent-Free Declaration

Neda Communications, Inc., the original Author of the LEAP protocols, has made a declaration to the FPF that the LEAP protocols are intended to be patent-free. This declaration includes statements to the effect that:

- To the best of the Author’s knowledge, the protocol is not subject to any patent restrictions.
- It is the Author’s intention to maintain the protocol patent-free.
- In the event that the Author becomes aware of any patent restrictions relating to the protocol, or if a patent right assertion is made with respect to the protocol, the Author undertakes to make prompt disclosure of this fact to the FPF.

The complete text of the declaration can be seen on the FPF website at http://www.FreeProtocols.org.

3.2.5 Maintenance and Enhancement

Protocols are usually not static, but instead typically undergo continued revision and enhancement. Depending on the character of the protocol, this may take place by closed, in-house processes, or by open and public ones. Since LEAP is intended to be a fully open protocol, it is maintained by fully open processes.

Open Maintenance Organizations

The LEAP protocols are developed and maintained through a set of public organizations. At present there are three of these:

- **LeapForum.org**. The LEAP Forum is a central clearing-house for information and pointers relating to the LEAP protocols in general. For complete information, visit the LEAP Forum website at http://www.leapforum.org/.
- **ESRO.org**. ESRO.org is an open organization for the development and maintenance of the ESRO protocol. For complete information, visit the ESRO website at http://www.esro.org/.
- **EMSD.org**. EMSD.org is an open organization for the development and maintenance of the EMSD protocol. For complete information, visit the EMSD website at http://www.emsd.org/.

Additional maintenance organizations are expected to be created as the LEAP protocol family grows.

None of these are standards organizations. They are simply forums to allow information exchange and cooperative effort relating to the LEAP protocols and technology.

Participation in these organizations takes place via various mailing lists which are hosted by each organization. Any interested company, organization or individual may participate by joining one or more of these mailing lists. Neither the organizations nor their mailing lists have any membership structure; nor does participation require any membership fees or other financial obligation.

For more information, and to subscribe to one or more mailing lists, visit the How To Participate area at any of the above websites.

Working Groups

In particular, ESRO.org and EMSD.org each host a Working Group mailing list. It is through the Working Group mailing lists that active development of the protocols takes place. Anyone may participate in the development of the ESRO and EMSD protocols by subscribing to the corresponding Working Group mailing list.

To subscribe to the ESRO Development Working Group mailing list, visit http://www.esro.org/joinESROmailingList/esroSpec.html.

To subscribe to the EMSD Development Working Group mailing list, visit http://www.emsd.org/joinEMSDmailingList/emsdSpec.html.

This open development process ensures that commentary and participation is open to any industry constituency that may be affected by the LEAP protocols.

Maintaining Patent-Freedom

Since the LEAP protocols are intended to be patent-free, the Working Groups must function in a way which preserves their patent-freedom. All LEAP protocol Working Groups therefore conform fully to the Free Protocols Foundation policies and procedures for Working Groups. These procedures ensure, insofar as possible, that a protocol remains patent-free despite undergoing collective development by a public Working Group. For more information, refer to the FPF website at http://www.FreeProtocols.org.

All Working Group contributors are required to adhere to these procedures. The Working Group imposes adherence to these procedures by requiring that all contributors agree to them as a condition of participation.

Both the ESRO Development Working Group and the EMSD Development Working Group have made declarations to the FPF that they conform fully to its patent-free policies and procedures. The text of these declarations can be seen on the FPF website.

3.2.6 Endorsement by a Standards Body

The ultimate arbiter of any particular protocol is the industry itself, in which a multitude of individual decisions leads to the acceptance or rejection of the protocol. The acceptance of a protocol as a standard is therefore something that occurs independently of formal endorsement by a standards body.

Standards organizations such as the IETG/IESG certainly have no monopoly on innovation, creativity, or competence. Any company, organization or individual is capable of creating protocols that become successful, far-reaching industry standards, without the sponsorship or sanction of a standards body.

For example HTTP, the protocol which forms the basis of world-wide Internet communications, was developed and achieved prominence entirely independently of any formal standards body. The same is true of Pretty Good Privacy (PGP), now a world-wide encryption standard. These and many other protocols became industry standards despite lack of any official endorsement.

For these reasons we believe that official endorsement of a protocol as a standard has little meaning, and that genuine legitimacy for a protocol derives from its industry usage. We therefore have no immediate plans to subject the LEAP protocols to any formal standards track process. If and when they become widespread within the industry, we may at that time place them on standards track and seek their formal endorsement as a standard.

3.3 Economic Consequences of Protocols

In general, protocols are a positive, enabling force for an industry. Protocols define an agreed-upon expected behaviour, allowing businesses to create products and services independently of one another, which can then be relied upon to interoperate correctly.

It may happen that one or more rival protocols arise, which address more or less the same industry need. Under these circumstances the rival protocols will effectively compete with one another, just as products and services do. It is usually most beneficial to the industry for there to be a single commonly-accepted protocol, so it is often the case that one protocol eventually displaces all others, thereby becoming an industry “standard.”

In the early stages of formation of a new industry, this competition among protocols is of great benefit to the industry. As in the case of products and services, competition is a mechanism for selection of the best. Free and open competition allows the success or failure of protocols to be determined on the basis of their merits, and their fitness to serve the overall interests of the industry.

In an ideal world, the success of protocols would be determined exlusively on this basis. However, this is not an ideal world, and forces can arise which interfere with the free and fair competitive process. In particular, the adoption of a protocol by an industry can have enormous economic consequences. And where there is money, there is the potential for corruption.

Businesses may attempt to exercise control over protocols in various ways. During protocol development, they may seek to introduce patented components into the protocol, that they can subsequently exploit to their advantage. They may seek to control the protocol publication process, thereby allowing them to impose copyright restrictions on the protocols, or even make unilateral, unannounced changes to the protocol specifications. They may develop and/or maintain the protocol by means of closed, exclusionary processes, which deny broad technical review of the protocol, and allow its design to be dictated by minority business self-interests.

3.3.1 Principles for Maintaining Protocol Integrity

We believe that there are three basic, fundamental principles for defending against these sorts of underhanded activities. These are:

- Patent-freedom
- RFC publication
- Maintenance by open organizations

Each of these provides a vital assurance of protocol integrity. Patent-freedom ensures that a patent-holder cannot subvert free-market competition among products and services based on the protocol. RFC publication ensures that the protocol is freely available to anyone who wishes to use it. And maintenance by open organizations ensures that development of the protocol takes place by democratic, rather than oligarchic, processes.

This trilogy of principles represent the most basic guarantees of the integrity of a protocol. If any one of these things is missing, then this means that some attempt is being made to control or limit access to the protocol. In the case of a public protocol, there is no valid reason for doing this.

3.4 Standards Organizations: Do They Mean Anything?

In addition to its general, industry-wide economic benefits, the adoption of a protocol also brings a unique set of benefits to its developer. These benefits consist of name recognition, and first-mover advantages in terms of development and sales of services and products based on the protocol. There may also be other significant benefits, such as those resulting from ownership of software patents related to the protocol.

These benefits can result in a huge financial windfall for the protocol developer. For this reason, the developer has an enormous vested interest in the adoption of his protocol in preference to any others. And this may lead the developer to promote his protocol in ways which subvert the free and fair competitive process among protocols. One of the ways a business may do this is by seeking to exert inappropriate influence or control over the activities of standards organizations. In particular, the developer may seek to have his protocol labelled as a "standard," while denying this label to other protocols. Alternatively, a business or group of businesses may form their own "standards organization" as an exclusive promotional vehicle for their own protocol. In either case the standards organization is in effect in the pocket of Big Business, and their discriminatory labelling of their own protocol as a "standard" represents another form of corruption. Though this serves the self-interest of the developer, it is likely to cause the industry to adopt the "wrong" protocol – i.e. not the one that best serves its overall needs. This is enormously detrimental to the industry at large and the consumer.

The incentives for businesses to indulge in these underhanded tactics is in direct proportion to the size of the industry and the financial stakes. And in the wireless data communications industry, the stakes are very high indeed.

As a result of this, we are currently seeing an enormous amount of standards-related activity in the wireless arena. Since 1998 a large number of self-proclaimed standards organizations relating to wireless data have been created, are claiming legitimacy, and are attempting to impose their own protocols on the industry. Among these recently-formed groups are:

- Wireless Application Protocol (WAP) Forum
- Mobile Internet Phone Services (MIPS) Forum
- Global Mobile Commerce Forum (GMCF)
- CDMA Development Group (CDG)
- Universal Wireless Consortium (UWC)
- GSM Alliance
- Global TDMA Forum
- Mobile Data Initiative
- Portable Computer & Communications Association (PCCA)

Each of these organizations is an independent entity, with its own claim of legitimacy, its own protocol publication mechanism, and its own set of restrictions and limitations on participation. Beyond the realm of specific wireless subnetworks, there is no reasonable need for this large number of independent standards organizations.

Many of these organizations are simply an instrument of Big Business. They are the result of a group of businesses forming an industry association or forum, for the purpose of branding their own protocol as a "standard."

At best, this labelling of some protocols but not others as "standards" is meaningless; it is a semantic shell game. And at worst, this labelling is an attempt by Big Business to market one set of protocols in competition with another. Marketing has its place in the promotion of a company's own products and services. But an industry protocol represents a public trust, and business marketing has no place in this arena.

We believe that the ultimate criterion of the legitimacy of a protocol is its acceptance and usage in the industry at large. And the industry at large is entirely capable of establishing its own winners and losers by means of free and fair competition among protocols. This arbitrary standards labelling serves only to corrupt this competitive process.

Our philosophy regarding protocol development requires only that the developer make sure that he adheres to the basic trilogy of principles described above. Beyond that, free and fair competition will do the rest.

By making it clear that these self-promoting consortia have no genuine legitimacy, we seek to reduce their influence and the harm they do to the industry.

3.5 Our Independence of the IETF

We are developing the LEAP protocols independently of the IETF, and we have not sought out their formal endorsement by the IETF.

Our decision to work independently of the IETF is a result of our previous experiences with IETF. Based on our interactions with the IETF, we have concluded that the illegitimate influence that the IESG and the IAB exert over non-IETF protocol specifications is counterproductive. We believe that the IESG and IAB are both prejudiced against externally-generate protocols (a frame of mind sometimes referred to as Not Invented Here syndrome), and that they engage in the active supression of competing protocols which originate outside their own domain.

The IETF has become an autocratic organization that acts to suppress innovation rather than encourage it. Indeed, on its own mailing lists, the IETF acronym has been referred to as the "Innovation Extermination Task Force," and the IETF/IESG/IAB has been referred to as a "cult." We do not consider either of these characterizations as being in the least inappropriate.

Our conclusion is that Big Business and political interests have now taken over the critical decision making processes within the IETF/IESG/IAB. We have come to this conclusion after many years of attempting to work within the system to bring the IETF back to it its original intended purpose. Our interactions with the IETF relating to several issues are publicly available:

- The complete record of our communications with the IESG and RFC Editor relating to publication of RFC-2188 is available at http://www.esro.org/communicationRecord/index.html.
- The complete record of our communications with the IESG relating to the IESG invitation to place RFC-2188 on standards track is available at http://www.esro.org/noStdTrack/main.html.
- Our complaint regarding the delays in publication of RFC-2188 is available at http://www.esro.org/iesgNote/index.html.
- The complete record of our communications with the IESG and RFC Editor relating to publication of RFC-2524 is available at http://www.emsd.org/communicationRecord/index.html.

Based on the above records we have come to the conclusion that the IESG is now characterized by irresponsibility, incompetence and arrogance.

3.5.1 Do We Need the IETF?

We believe that a lesser IETF will be a better IETF. The IETF has been a source of new protocols in the past, and there is no reason why it cannot continue to create and develop new protocols in the future.

But traditionally, the scope of IETF has gone far beyond this; the IETF has also claimed responsibility for judging, labelling and ratifying protocols. We believe that these IETF functions are unnecessary and inappropriate. All the functions that the IETF claims to provide can be accomplished by means of:

- The Free Protocols Foundation

- The RFC publication process
- Maintenance by open Working Groups
- Free and fair competition among protocols

The first three items above are sufficient to ensure that a protocol is patent-free, is freely available, and is open to democratic and egalitarian development processes.

And the fourth item above provides a more than adequate mechanism for determining which protocols become enduring industry standards, and which fall by the wayside.

So what positive thing can the IETF offer that is not better provided elsewhere? The answer is: not much.

We have no objection to the IETF existing as an entity, developing protocols, and making them available for use. What we demand is that other protocols have exactly the same opportunities as those of the IETF. These consist of the opportunity to be made public, and the opportunity to be judged *by means of open competition*, and *on the basis of their merits*.

What we object to most strenuously, is the IETF/IESG/IAB arrogating to itself the power to quash protocols that it considers to be in competition to its own. This is not the job of the IETF/IESG/IAB; this is the job of fair competition.

We believe that the collective intelligence and expertise of the Internet technical community is adequate protection against widespread adoption of "wrong" protocols, and we believe that the market and the consumer can collectively make the right eventual decisions. The actions of irresponsible entities such as IESG and IAB, claiming illegitimate authority to select winners and losers at the time of initial publication of protocols, on the pretext of protecting the network and the consumer, in fact does more harm than good.

Chapter 4

Free Protocols Foundation Policies and Procedures

4.1 Introduction

4.1.1 The Patent Debate

At the time of writing, there is an on-going debate within the software industry regarding software patents. Like many others within the industry, at the Free Protocols Foundation we regard the historical tradition of patents as being entirely inappropriate for software.

We consider software patents to have the effect of inhibiting free and open competition within the software industry, and to be extremely detrimental to the industry and the consumer. A complete discussion of the software patent issue is outside the scope of this document. For more information on this subject see [**?**] or [**?**].

4.1.2 How Patents Affect Protocols

Patents are applied to software, not to protocols. It is not possible to patent a protocol; in general only a process or an algorithm can be patented. However, a protocol may include a patented algorithm as an integral part of its specification. In this case, *any software implementation of the protocol requires the use of patented software*. That is, a patented process is an inherent part of the protocol.

Even if a protocol does not explicitly decree the use of a specific patented software process, it may still be the case that any practical implementation of the protocol requires the use of patented software components. The protocol could in principle be implemented in a way which avoids the use of patented software; in practice, however, the result would be a significantly inferior implementation, for example in terms of efficiency.

In either case, the protocol effectively implies the use of patented software. We will refer to any such protocol as a **patented protocol**. That is, a patented protocol is any protocol whose practical implementation requires the use of patented software components.

We will use the term **patent-free protocol**, or just **free protocol**, to refer to a protocol which is functionally free from software patents. By "functionally free from software patents," we mean either that the protocol is truly free from patents, or if the protocol does imply the use of patented software, that the patent-holder has granted non-restrictive rights to include the patented software components in implementations of the protocol.

In either case, the result is that the protocol can be freely implemented and used by anyone, without encountering significant restrictions.

4.1.3 Difficulties Relating to Software and Protocol Patents

Because of the current state of affairs regarding software patents, it is no longer possible to be absolutely certain that any particular protocol is patent-free. Whether or not a new invention or technology violates any existing patents has traditionally been determined by means of a patent-search – a direct search and examination of all relevant pre-existing patents. In the case of a physical technology, this yields a more or less definitive answer as to whether or not the technology violates any existing patents.

In the case of software, however, it is not possible to answer this question with the same degree of certainty. The basic reason for this lies in the very high degree of subtlety and complexity of modern software systems. A software system may contain hundreds or thousands of individual software constructs, any one of which may potentially violate an existing patent. Furthermore, it is very difficult to establish a taxonomy of existing software patents. A taxonomy is required to guide the patent-search process – the taxonomy defines which patents are "related" to the technology under search. Without a well-defined and meaningful taxonomy, it is not possible to define an appropriate scope of search.

In other words, a software system may contain a small universe of software constructs. Meanwhile, the Patent and Trademark Office has established a small universe of software patents. Unfortunately, there is no way to establish with certainty, that none of the elements of one universe also reside in the other universe.

To make matters worse, there can be room for dispute regarding whether or not a particular software construct violates a particular patent. The patent-holder may claim that it does, while the software developer claims that it does not. If the two parties are unable to come to agreement, the issue can only be resolved in the courts.

Finally, because of the delay between a company filing a patent application and the granting of the patent by the PTO, it is entirely possible for a company to conduct a software patent-search with negative results, only to discover subsequently that a patent has been granted after-the-fact, which now places its software in violation.

For all these reasons, it has now become essentially impossible to establish with certainty that a particular piece of software is, and will remain, truly patent-free. Likewise, it has become impossible to establish with certainty that a particular protocol is, and will remain, patent-free.

4.1.4 Terminology

Legal rights such as patents and copyright are sometimes referred to collectively as "Intellectual Property Rights." Although this is a very commonly used term, we will avoid using it in this document.

Along with other authors, we object to the use of this term on the grounds that it blurs the distinction between *intellectual* items and *material* ones. The law allows ownership rights to be asserted with regard to both types of item, and therefore both may be considered in some sense to be "property." However, we regard intellectual entities such as software and protocols to be fundamentally different in nature to material items, and worthy of entirely different legal treatment relating to questions of ownership. For more discussion about this point, see [**?**].

Where necessary, we will use explicit terms such as "patent," or "copyright," to refer to legal rights relating to intellectual constructs.

Definitions

Throughout this document, we will use the following terms with the meanings indicated:

- **Truly Patent-Free Protocol**. A protocol that can be implemented in the form of entirely patent-free software. A Truly Patent-Free Protocol contains no limitations whatsoever on its dissemination and use, and may be freely implemented by anyone, without restriction.
- **Functionally Patent-Free Protocol**. A protocol for which there are either no known software patent restrictions, or where software patent restrictions are known to exist, non-restrictive usage rights have been obtained from the patent-holder.
- **Free Protocol Specification**. A protocol that conforms to the policies and procedures of the Free Protocols Foundation relating to patents, copyright, and confidentiality. These procedures are set forth in Section 4.6.
- **Declared Patent-Free Protocol**. A protocol for which a declaration has been made, to the Free Protocols Foundation, that it conforms to the procedures of Section 4.6.
- **Author of a Protocol**. The individual, company or organization, or the set of individuals, companies or organizations, who are responsible for the creation, development, maintenance, or enhancement of the protocol specification.
- **Working Group**. An open-ended set of individuals or organizations who collectively work towards the creation, development, maintenance, or enhancement of the protocol specification.
- **Free Protocol Development Working Group**. A Working Group which has voluntarily elected to adhere to, and be bound by, the policies and procedures of the Free Protocols Foundation regarding patent-freedom.
- **Free Protocol Developer**. A contributor to a Free Protocol Development Working Group.

4.1.5 About the Free Protocol Processes and Procedures

This document establishes a set of policies and procedures for protocol development that is designed to ensure, as far as this is possible, that the resulting protocol is functionally patent-free. The purpose of these procedures is to create a resulting protocol that is either free from patent restrictions, or for which non-restrictive usage rights have been obtained from the patent-holder. These procedures are set forth in Section 4.6.

The procedures of Section 4.6 are based on a similar set of procedures defined by the IESG (Internet Engineering Steering Group). Specifically, the FPF procedures are an extension of Section 10, *Intellectual Property Rights*, of RFC 2026, *The Internet Standards Process – Revision 3* [**?**].

That section defines the procedures to be followed by the IETF (Internet Engineering Task Force) relating to patent-freedom. However, the scope of RFC 2026 is largely limited to the needs of the IESG itself; in particular, the patent-related procedures of Section 10 of RFC 2026 are limited to standards-track documents and other IETF-specific purposes. Thus, while these patent procedures may be entirely adequate for the needs of the IETF, they are not adequate to dealing with patent-freedom in a more general setting.

The processes and procedures in Section 4.6 of this document are therefore an extension of the RFC 2026 procedures, designed to address the need for patent-freedom procedures in general. They are intended to be a set of general-purpose processes which can be adopted by any protocol development organization.

4.1.6 About this Document

This document is available in several alternative formats at Free Protocols Foundation website (http://www.freeprotocols.org/freeProtocolProcess):

- ONE-HTML: Displays the entire document as a single web page.

- SPLIT-HTML: Displays the document as a set of linked web pages, starting from the Table of Contents.
- PDF: Provides the document in Adobe Acrobat format. Note that the Adobe Acrobat Reader is required to read this format. The Acrobat Reader can be downloaded from the Adobe website.
- PS: Provides the document in PostScript format for printing.
- Text Only: Provides the document in text-only format.

4.2 The Protocol Development Process

Protocols come in all shapes and sizes, and from a variety of sources. Some are proprietary, intended for use exclusively by their developer. Others may be "open" in some sense, indicating that they are intended for more general, public usage. In this context, the word "open" can mean any one of several different things. It may mean nothing more than that the protocol has been published by its developer. The protocol may still be very tightly controlled: revision of the protocol may remain the exclusive right of the developer; the protocol may be protected by patent or copyright restrictions; and use of the protocol may require a licensing agreement. This is a very narrow, and to our mind misleading, use of the word "open."

At the other extreme, the protocol may be open to a very high degree of public accessibility: it can be published by an open mechanism such as RFC publication; undergo revision by means of public working groups; and be entirely free of usage restrictions. A protocol satisfying all these criteria can be said to be "open" in the broadest sense. Protocols are often referred to as "open" to imply that they are open in a broad sense, whereas in fact they are open only in the narrowest sense.

Also, protocols can have widely differing periods of industry tenure. Some protocols never achieve widespread acceptance and usage, or else have very short lifetimes before becoming obsolete or being eclipsed by competing protocols. Other protocols become highly successful, and persist as long-term industry standards.

4.2.1 Phases of Development

Over its lifetime, a protocol typically goes through a number of developmental phases. In general, from conception to decease, a protocol may go through some or all of the following phases:

1. Initial development.
2. Global parameter assignment.
3. Publication.
4. Patent-free declarations.
5. Industry usage.
6. Maintenance and enhancement.
7. Endorsement by a standards body.

Depending on its purpose, nature, and history, a protocol may undergo some, all, or none of these phases. Also, a protocol may iterate through phases 3 - 7 multiple times, as it undergoes maturation via repeated revision and re-publication. As described later, the Free Protocols Foundation plays a role in only two of these phases. For completeness, however, in the following sections we provide a brief description of each phase, along with commentary on the FPF's philosophy regarding the protocol development process.

Initial Protocol Development

The conception and early development of a protocol can take place in a variety of ways. A traditional source of Internet protocols is the IETF/IESG. Other sources of protocols are private businesses, the academic community, or even a single individual.

We believe that there is a tendency among established standards bodies to regard their own, officially sanctioned protocols as authoritative, while any other protocol is of questionable worth or validity. However, the history of protocol development does not support this view. Small groups of individuals have created protocols with far-reaching consequences (e.g. the World Wide Web), just as established standards bodies have created protocols which failed to achieved industry acceptance.

At the Free Protocols Foundation, we do not regard any one source of protocols as necessarily superior to any other. We believe that any coordination of activities can generate useful protocols, and we are ready to provide the same support for patent-freedom regardless of the initial source of the protocol.

Global Parameter Assignment

If necessary, global parameters must be assigned to the protocol, e.g. by the IANA. The Free Protocols Foundation plays no role in this process.

Protocol Publication

If the protocol is intended to be an open protocol, as opposed to an exclusively proprietary one, then it must be made publicly available. This can be accomplished in various ways; the protocol can be self-published by the developer, or it can be published through an independent agency such as the Internet RFC Editor.

Ideally, the protocol should be published in a way which allows permanent and unrestricted access to the protocol by anyone wishing to implement it. In the case of Internet protocols, this is usually accomplished by RFC publication.

Patent-Free Declarations

Depending on their intentions, the developers of a protocol may take steps to work towards a patent-free result, and they may wish to make certain declarations to that effect.

In general, there may be both an **Author** and a **Working Group** involved in the development of a protocol. The Author is the person, company, or other entity that has primary responsibility for the protocol. In some cases, the protocol may also undergo development at the hands of a **Working Group** – a set of independent individuals or companies who work cooperatively on the protocol, usually via public mailing lists.

Both the Author and the Working Group may wish to make declarations regarding the patent-freedom of the protocol. The Author may wish to make an initial declaration that the protocol is intended to be patent-free. As described later in this document, it is not possible to make an absolute guarantee that a protocol is, and will remain, completely patent-free. The best an Author can do is make a good-faith declaration that:

- *To the best of his knowledge* the protocol is not subject to any patent restrictions.
- It is the Author's intention to maintain the protocol patent-free.
- If any patent restrictions relating to the protocol become known to the Author, he will make prompt disclosure of this.

Similarly, the Working Group may wish to make a declaration that:

- The protocol has been developed in such a way as to ensure that no patented components have been intentionally introduced into the protocol.

- If any patent restrictions relating to the protocol become known to the Working Group, it will make prompt disclosure of this.

One of the roles of the Free Protocols Foundation is to provide a public forum in which such declarations can be made. Any such declaration which is submitted to the FPF will be published on our website at http://www.FreeProtocols.org. Examples of previously submitted declarations may be seen at that location.

Industry Usage

The ultimate test of a protocol is whether or not it becomes widely accepted and implemented in the industry. If a protocol is largely unused, or eclipsed by a competing protocol, then it is largely irrelevant.

Maintenance and Enhancement

Protocols are usually not static, but instead typically undergo revision and enhancement in response to experience and/or changing industry requirements. Depending on the intentions of the Authors, this may take place by closed and proprietary processes, or by open and public ones. In the case of a truly open protocol, the development process should allow commentary and participation by all the constituencies that are affected by the protocol.

In some cases, continued development and enhancement of the protocol is accomplished by means of a public Working Group. Also depending on the Authors' intentions, the Working Group may function in a way which preserves the patent-freedom of the protocol, and the Working Group may wish to make a declaration to this effect.

Two things are required in order to achieve these goals. First, the developers must establish and follow a set of Working Group operating procedures that will have the effect of preserving the patent-freedom of the protocol. Second, the developers must make a public declaration that the Working Group follows these procedures.

A developer can achieve both of these things without the assistance of the Free Protocols Foundation. The development organization can establish its own set of Working Group operating procedures, and can independently announce that the Working Group follows them.

However, the Free Protocols Foundation provides a means of accomplishing these things which is external to, and independent of, the development organization itself. It is for this purpose that the FPF primarily exists. First, the FPF defines a clear and unambiguous set of Working Group processes and procedures which ensure, as far as possible, that the resulting protocol will remain functionally patent-free. Any development organization is free to adopt these procedures with regard to its own protocol. Second, the FPF provides an external forum in which the developer may declare publicly that its Working Group follows these procedures.

The FPF Working Group procedures are designed to:

- Ensure that no patented components can be intentionally introduced into the protocol as a result of the Working Group activities.

- Provide remedies in the case of unintentional introduction of patented components into the protocol. These remedies consist of prompt publication of the fact of the patented component, and an attempt to secure non-restrictive usage rights from the patent holder.

Endorsement by a Standards Body

The ultimate arbiter of protocols is the industry itself, in which a multitude of individual decisions leads to the acceptance or rejection of any particular protocol. The acceptance of a protocol as a standard is therefore something that occurs independently of formal endorsement by a standards body.

Nevertheless, once a protocol has become accepted as an industry standard, it is often the case that it receives the formal sanction of a standards body.

4.2.2 Role of the Free Protocols Foundation

It is sometimes the case that many of the above phases of development take place under the direction of a single institution, or a group of tightly coupled institutions. For example, when developing protocols the IETF/IESG/IAB traditionally claims authority for all aspects of development except for (5), over which, of course, it has no direct control.

At the Free Protocols Foundation, we consider it undesirable to place control of multiple aspects of the development process in the hands of any single institution. First, this can include built-in conflicts of interest. For example, if a standards body is responsible both for developing protocols and publishing industry protocols, the body may be inclined to favor publication of its own protocols in preference to competing protocols from other sources, or it may be inclined to place inappropriate commentary within competing protocols. The IETF/IESG has a history of doing both of these things.

As another example, if the Maintenance and Enhancement responsibility is closely-held by a developing company or group of companies, this process may be biased in favor of the companys' interests, rather than those of the industry at large.

Furthermore, a large spread of responsibility within a single institution can lead to bureaucratization of its activities; the energy of the organization can become directed towards maintaining its internal bureaucracy, rather than serving the needs of its consumers. In other words, the institution can become authority oriented, as opposed to responsibility oriented. The historical evolution of the IETF/IESG/IAB is a classic example of this.

For these reasons, the Free Protocols Foundation is in favor of decoupling, as much as possible, the responsibility for the various aspects of development. A separation of powers greatly lessens the potential for conflicts of interest. Furthermore, an organization with limited responsibility can be discarded, reformed, or replaced more easily than one with very broad responsibility. Separation of powers thus provides a greater degree of choice, and therefore competition, within the protocol development process.

The Free Protocols Foundation is therefore in favor of placing responsibility for the various phases and aspects of development in the hands of independent organizations with limited agendas. We favor delegating the Protocol Publication phase to a truly independent third-party entity, such as the Internet RFC Editor. We favor handling the Maintenance and Enhancement phase by means of a variety of truly open public working groups, not just the IETF. Both of these steps ensure unbiased processing of the protocol.

In the same spirit, we favor placing responsibility for working towards patent-freedom in the hands of an independent organization. It is primarily for this reason that the Free Protocols Foundation exists. The role of the Free Protocols Foundation is to place those aspects of the protocol development process which relate to patent-freedom in a common, central, public location. The various other aspects of the development process have been described only to place the role of the FPF in its proper context; the FPF plays no role in those aspects.

In our model of the protocol development process, the scope of FPF activities is limited to two items exclusively:

- (4) Patent-free declarations, and
- (6) Maintenance and enhancement,

and the Free Protocols Foundation has no agenda other than this.

Note that the role played by the FPF regarding protocol patents is somewhat analogous to the role played by the RFC Editor regarding protocol publication. RFC publication provides a means of publishing, via an independent agency, Internet protocols which have been produced by a variety of sources.

Similarly, the FPF represents a means of dealing with patent issues by an independent agency. Hitherto, this responsibility has been taken by multiple standards bodies, each of which has been obliged to define its own processes and procedures relating to patents. By adopting the FPF procedures, a standards body or development organization can make use of a set of general services established by an external agency.

4.2.3 Coordination of Activities

As noted in the previous section, the FPF is in favor of distributing responsibility for the various aspects of protocol development, rather than consolidating all these aspects under the umbrella of a single organization, such as the IETF.

The objection may be made, that this distribution of responsibility creates coordination difficulties. It can be argued that vertically-integrated organizations like the IETF play a valuable role in terms of coordination of activities, and that it is more difficult to coordinate the activities of multiple independent organizations. In particular, it can be said that the IETF assists in the logical and orderly development of multiple protocols, by establishing a common architecture and structure for sets of related protocols.

However, we believe that this objection is unfounded. It has been well demonstrated, for example by the development of Linux, that multiple independent entities can coordinate their development efforts extremely effectively. In any event, the advantages to be gained from a separation of powers certainly exceed the drawbacks.

4.3 The Free Protocols Foundation

4.3.1 General Philosophy

At the Free Protocols Foundation, we consider that patented protocols are very undesirable. A protocol is a general agreement within an industry that things will be done in a certain way. It represents an agreed-upon expected behavior, providing common ground for cooperation among a variety of disparate entities: private businesses, academic institutions, government, and individuals. The protocol allows any of these entities to create products, services, applications, etc., and provides the necessary assurances that these offerings will function and interoperate in a well-defined way. The protocol is a positive, enabling force for the entire industry.

In order for the protocol to play this role, anyone who wishes to implement and use it must be able to do so freely, and without restrictions. The presence of patented components within the protocol places restrictions on its usage, and therefore serves only to undermine this purpose. Furthermore, the presence of the patent brings an enormously unfair market advantage to the patent holder. By exercising his patent rights, the patent holder can gain financial benefit from competing companies who wish to use the protocol, or can stifle competition altogether by denying the competing company the right to use the protocol at all.

We have no objection to companies seeking market advantage by means which benefit the consumer and the industry. Such means include the creation of superior products and services, or the exercising of legitimate patents on physical processes. But we strenuously object to the seeking of market advantage by attempting to lay claim to the protocol itself.

The presence of a patent within a protocol is at best absurd, pointless, and self-defeating. And at worst, it represents an attempt by corrupt business interests to gain market advantage at the expense of the industry and the consumer.

4.3.2 Purpose, Activities and Scope

The Free Protocols Foundation is a nonprofit corporation, incorporated in the State of Washington.

The principal purpose of the Free Protocols Foundation is to act as an independent public forum for the support of patent-free protocols. We do this by means of the following major activities:

- By providing an independent, external forum in which an Author may make an initial declaration that a protocol is intended to be patent-free.
- By defining a set of patent-related Working Group procedures which, if followed, ensure that the resulting protocol will remain functionally patent-free. Any development organization is free to adopt these procedures with regard to its own protocols.
- By providing an independent, external forum in which a Working Group may make a public declaration that it follows these procedures.
- Whenever patent rights are asserted with respect to any protocol which has been declared patent-free to the FPF, by publishing a statement of the patent right assertion.
- Whenever patent rights are known to exist with respect to a protocol which has been declared patent-free to the FPF, by assisting in obtaining from the patent-holder a non-restrictive license to implement the patented process as part of the protocol.

The scope of activities of the Free Protocols Foundation is limited to those activities which provide support for free protocols, and which oppose the mis-use of patented protocols.

In particular, the Free Protocols Foundation does not develop protocols itself, nor does it participate in the development of the protocols of other organizations. The FPF does not evaluate or provide any commentary on the technical merits of protocols.

Also, the FPF does not make any independent verification of whether or not a developer actually adheres to the processes and procedures of Section 4.6. The FPF does no more than record the fact that the development organization *has made the declaration* that it conforms to those procedures.

4.3.3 Other Activities

In addition to its activities undertaken to support the development of patent-free protocols, the FPF may also take actions to oppose the creation and promotion of patented protocols. These actions may consist of any of the following:

- Acting as a clearing house for information and articles relating to protocol patent-freedom.
- Supporting the creation and development of patent-free alternative protocols to existing patented protocols.
- Alerting legislators of the harmful effects of software and protocol patents, and lobbying for changes in the way software patents are granted.
- Alerting the United States Patent and Trademark Office (PTO) of the harmful effects of software and protocol patents, and lobbying for changes in the way sofware patents are granted.
- Supporting fights against invalid software patents in the courts.
- Boycotting companies which misuse software and protocol patents.

We encourage others to join us in resisting the granting and abuse of software and protocol patents.

4.4 Free Protocol Development Working Groups

It is often the case that protocols are developed, or maintained and enhanced, by means of public Working Groups. A Working Group may enter into participation at various stages in the protocol's development.

In many cases, the creation and initial development of a protocol is done by a single Author or a small team of core developers. Once an initial basis for the protocol is in place, a larger public Working Group is then formed, which takes over responsibility for maintenance and enhancement of the protocol. In this case, the Working Group begins participating at stage (6), Maintenance and enhancement.

In other cases, a Working Group may be formed in order to do the initial creation and development work itself. In this case, the Working Group begins participating at the very beginning, at stage (1), Initial development.

Working Groups typically communicate amongst themselves by means of working group mailing lists, together with occasional physical meetings as necessary. Under these circumstances, it is possible for a large number of contributors to participate in the development of a protocol.

The processes and procedures of Section 4.6 include provisions whereby patent-freedom can be maintained for a protocol despite its being developed by a public Working Group. These provisions consist of a set of procedures that, if followed by the Working Group, will keep the protocol functionally patent-free. A key provision is that anyone who participates in the Working Group is required to adhere to these procedures. That is, the Working Group must impose adherence to its procedures on all Working Group contributors.

Note that the FPF does not manage Working Group mailing lists itself, nor does it maintain a list of Working Groups which have adopted the FPF's processes and procedures. Though the FPF does provide a mechanism whereby Working Groups may make a declaration that they conform to the FPF Working Group procedures, it does not provide a mechanism whereby individual contributors may make such a declaration.

The role of the FPF regarding Working Groups is limited to that of establishing a minimal set of policies and procedures that contributors may choose to adopt in order to maintain a patent-free protocol. This minimal set of policies relates only to patents, copyright and confidentiality. All other procedures are at the discretion of each individual Working Wroup.

4.5 Patent-Free Declarations

4.5.1 Author's Declaration

Any Author may make an initial patent-free declaration to the Free Protocols Foundation relating to a protocol. This declaration should include statements to the effect that:

- To the best of the Author's knowledge, the protocol is not subject to any patent restrictions.
- It is the Author's intention to maintain the protocol patent-free.
- In the event that the Author becomes aware of a patent right restriction relating to the protocol, or a patent right assertion is made with respect to the protocol, the Author will make prompt disclosure of this fact to the Free Protocols Foundation.

Any such statement provided to the FPF will be published on the FPF website. Examples of statements previously provided to the FPF can be found in the Author's Declarations section of the FPF website.

4.5.2 Working Group Declaration

If a Working Group participates in the development and/or maintenance and enhancement of a protocol, it may make a declaration that its protocol maintenance and enhancement procedures conform to the FPF Processes and Procedures regarding patent-freedom. A Working Group may do this by providing the Free Protocols Foundation with a statement that its protocol development process conforms to the processes and procedures set forth in Section 4.6.

Any such statement provided to the FPF will be published on the FPF website, and the corresponding protocol will be added to the list of those declared to conform to the FPF's patent-free policies and procedures.

Examples of statements previously provided to the FPF can be found in the Free Protocols Working Group Declarations section of the FPF website. The list of declared-conforming protocols is available in the List of Free Protocols Specifications and Their Declarations section of the FPF website.

4.6 Patents, Copyright and Confidentiality - Policy Statement

In this section we define a set of policies and procedures which ensure that a protocol will remain functionally patent-free. Any Working Group is free to make use of these procedures as part of their development process. A key component of these procedures is that every member of the Working Group is required to abide by the procedures.

4.6.1 Policy Statement Principles

Because of the difficulties relating to software patents described in Section 4.1.3, it is not possible to be absolutely certain that a protocol is truly patent-free. The scope of these policies and procedures is therefore limited to ensuring that a protocol is patent-free *as far as is practically possible*. The purpose of the procedures is to codify the following principles:

- **Author's Patent-Free Intent Declaration**. When a developer makes a patent-free declaration to the FPF, a key part of the declaration is that of *intent*. That is, the declarer is making the statement that *to the best of the declarer's knowledge* the protocol is patent-free, and that it is the declarer's intention to keep it so.
- **On-Going Patent-Free Contributions**. By becoming a member of a Working Group, every contributor to the on-going maintenance and enhancement of the protocol is required to adhere to principles and procedures which preserve patent-freedom of the protocol.
- **Working Group's Declaration** When a Working Group makes a declaration to the FPF, the effect of this declaration is that the Working Group's activities conform to a set of processes that ensure, as far as possible, that the resulting protocol specification is functionally patent-free.
- **Patent Assertion Disclosure** If a patent assertion is made subsequent to the declaration, the declarer undertakes to make prompt announcement of this to the Free Protocols Foundation. The Free Protocols Foundation maintains a record of all patent right assertions that have been made against any of its listed protocols. This record is available in the Notices of Claimed Rights section of the FPF website.
- **Obtaining Non-Restrictive Usage Rights**. In the event that the developer becomes aware of a patent restriction relating to the protocol, the developer will attempt to obtain non-restrictive usage rights for the protocol.

4.6.2 General Policy

In all matters of patent and confidentiality rights and procedures, the intention of the FPF is to benefit the Internet community and the public at large, while respecting the legal rights of others.

4.6.3 Confidentiality Obligations

No Free Protocol Developer shall make any contribution to the Free Protocol Working Group that is confidential. No contribution that is subject to any requirement of confidentiality or any restriction on its dissemination shall be included in any part of a Free Protocol Specification.

4.6.4 Rights and Permissions of All Contributions

By submission of a contribution, each person actually submitting the contribution is deemed to agree to the following terms and conditions on his own behalf, on behalf of the organization (if any) he represents and on behalf of the owners of any proprietary rights in the contribution. Where a submission identifies contributors in addition to the contributor(s) who provide the actual submission, the actual submitter(s) represent that each other named contributor was made aware of and agreed to accept the same terms and conditions on his own behalf, on behalf of any organization he may represent and any known owner of any proprietary rights in the contribution.

1. Some works (e.g. works of the U.S. Government) are not subject to copyright. However, to the extent that the submission is or may be subject to copyright, the contributor, the organization he represents (if any) and the owners of any proprietary rights in the contribution, grant an unlimited, perpetual, non-exclusive, royalty-free, world-wide right and license to the Free Protocols Foundation and the Free Protocol Working Group under any copyrights in the contribution. This license includes the right to copy, publish and distribute the contribution in any way, and to prepare derivative works that are based on or incorporate all or part of the contribution, the license to such derivative works to be of the same scope as the license of the original contribution.

2. The contributor acknowledges that the Free Protocol Working Group and the Free Protocols Foundation have no duty to publish or otherwise use or disseminate any contribution.

3. The contributor grants permission to reference the name(s) and address(es) of the contributor(s) and of the organization(s) he represents (if any).

4. The contributor represents that the contribution properly acknowledges major contributors.

5. The contributor, the organization (if any) he represents, and the owners of any proprietary rights in the contribution agree that no information in the contribution is confidential and that the Free Protocol Working Group and the FPF may freely disclose any information in the contribution.

6. The contributor represents that he has disclosed the existence of any proprietary or patent rights in the contribution that are reasonably and personally known to the contributor. The contributor does not represent that he personally knows of all potentially pertinent proprietary and patent, confidentiality and copyright rights owned or claimed by the organization he represents (if any) or third parties.

7. The contributor represents that there are no limits to the contributor's ability to make the grants acknowledgments and agreements above that are reasonably and personally known to the contributor.

4.6.5 FPF Role Regarding Free Protocol Specifications

1. Where any patents, patent applications, or other proprietary rights are known, or claimed, with respect to any Free Protocol Specification, and brought to the attention of the FPF, the FPF shall prepare a note, to be included in the next revision of the Free Protocol Specification, indicating the existence of such rights, or claimed rights.

2. The FPF encourages all interested parties to bring to its attention, at the earliest possible time, the existence of any patent rights pertaining to Free Protocol Specifications.

3. Where the FPF knows of rights, or claimed rights under (1), the FPF shall assist the Free Protocol Working Group in attempting to obtain from the claimant of such rights, a written assurance with respect to the relevant protocol specification(s), that any party will be able to obtain the right to implement, use and distribute the technology or works when implementing, using or distributing technology based upon the specific specification(s) under openly specified, reasonable, non-discriminatory terms.

Chapter 5

ESRO: A Foundation for the Development of Efficient Protocols

5.1 Overview of ESRO

All efficient applications have the requirement for an efficient transport mechanism. For this reason, part of the initial focus of the LEAP protocol development effort has been on creating a general efficient transport mechanism. The resulting protocol is referred to as **Efficient Short Remote Operations**, or **ESRO**. ESRO is the efficient transport layer protocol for several LEAP applications. ESRO is a reliable connectionless transport mechanism, forming the foundation for the development of efficient protocols when TCP is too much and UDP is too little.

The need for efficient protocols extends across all aspects of wireless data communications, including e-mail, web browsing, and other applications. The LEAP architecture accommodates many of these applications based on the use of ESRO.

ESRO was published as RFC-2188 [**?**] in September of 1997. The ESRO protocol is publicly maintained and enhanced by ESRO.org at http://www.esro.org/. Patent free declarations have been made with respect to ESRO through the Free Protocols Foundation.

5.1.1 The Need for ESRO

Considering that:

- Many vertical applications in the wireless environment need a *reliable*, *efficient* and *lightweight* data transfer mechanism.
- Existing Internet transport protocols (TCP/UDP) do not address these requirements adequately.
- Today, most vertical applications either implement their own *ad hoc* reliability service on top of UDP, or else use non-standard middleware.
- A reliable connectionless transport protocol based on open specifications is needed.
- Remote Operations is superior to reliable connectionless transport as an application development paradigm.

ESRO has been designed to address these specific needs.

The need to address the requirements that the ESRO protocol addresses has been well recognized for quite a long time. As early as 1995 such requirements were being discussed within the Internet research community. In particular, RFC-955 [**?**], entitled "Towards a Transport Service for Transaction Processing Applications," demonstrates recognition of the need for ESRO. A more recent document, RFC-2757 [**?**], entitled "Long Thin Networks," again recognizes the demand for such a protocol.

In the past, this unaddressed demand has given rise to a variety of product oriented proprietary solutions (as opposed to open protocols) which are often referred to as "Middleware Products." This has been particularly widespread in the wireless industry. Often these solutions are vendor specific and do not scale.

5.1.2 ESRO Requirements and Goals

The requirements and goals driving the ESRO protocol and system design include the following:

1. Provide reliability in an efficient manner for a wide range of vertical applications (e.g. wireless).
2. Specify an Internet open protocol.
3. Minimize the number of transmissions.
4. Minimize the number of bytes transmitted.
5. Be fast; minimize latency.
6. Be power efficient, and show respect for the resource limitations of mobile platforms, including memory, CPU, and battery power limitations.
7. Be lightweight; accommodate miniaturized devices.

5.1.3 Terminology

A description of ESRO includes references to a number of basic data communications concepts and ideas. Because of the informal, *ad hoc*, and often inconsistent terminology used within the Internet technical community, many of these concepts are referred to by a variety of different names throughout various documents and RFCs. Many of the RFCs mentioned below refer to ESRO-related concepts in an inconsistent manner.

In contrast to this, the ESRO specification uses the well defined and precise terminology of the "Open Systems Interconnection Reference Model" [**?**]. In this paper we will also adhere to the same precise terminology.

An example of the use of imprecise terminology is the term "Transaction Processing." Various other protocol specifications refer to the concept of "Remote Operations" as "Transaction Processing." In our terminology, however, Transaction Processing goes above and beyond Remote Operations, and also includes the concepts of Commitment, Concurrency and Recovery, and chained (related) Remote Operations which may be built on top of ESRO.

5.2 Other Related Protocols

The overall model of ESRO is similar to and consistent with many existing protocols. The distinguishing characteristic of ESRO is its efficiency. Also, the scope of ESRO is very narrow and well defined. The options and selections that it provides for are few and clear.

A brief comparison of ESRO to other similar protocols is provided in the sections below.

5.2.1 RPC

Remote Procedure Call (RPC) is specified in RFC-1831 [?] and RFC-1833 [?].

The RPC specifications define a remote procedure model that is essentially the same as ESRO. However, the notation of RPC uses a syntax which is quite different from that of ESRO. RPC can rely on a connection-oriented or a connectionless transport mechanism. When using the connectionless mechanism, the retransmission and reliability issues are considered to be beyond the scope of the RPC specification. RPC is usually used in combination with External Data Representation, XDR (RFC-1832) [?].

When using RPC over UDP, no meaningful reliable transport mechanism is defined. For this reason use of RPC over UDP has been limited to LANs.

5.2.2 ROSE

Remote Operations Services Element (ROSE) is specified in [?] and [?]. ROSE is a complete and heavyweight traditional OSI application which assumes availability of all of the underlying OSI layers.

The ESRO protocols can accomplish short operations with much less overhead than ROSE.

5.2.3 WAP's WTP

The Wireless Appliction Protocol (WAP) includes a layer which is similar to ESRO, called "Wireless Transaction Protocol" (WTP) [?].

The WTP specification was published after ESRO was published, and is similar in functionality to ESRO. In many ways WTP can be considered to be patterned after ESRO; however, WTP is in fact a step backwards.

The clear and simple Remote Operations model of ESRO is renamed as "Wireless Transactions" in an inappropriate context. The notation specification conventions are not used, and the state machines are not as robust.

More importantly, the WTP specifications are not stable, have not been published as Internet RFCs, and have not been declared to be patent free.

As early as 1995, those involved in the development of WTP were made aware of LSROS and ESRO. The only reason we know of for their re-specification of WTP, rather than re-use of ESRO, is the WAP Forum's desire for control.

More details on the problems of the WAP Forum's approach are provided in the article *The WAP Trap* [?].

5.2.4 T/TCP

Transaction/TCP is specified in [?] and [?]. T/TCP is a backwards-compatible extension of TCP which provides efficient transaction-oriented service in addition to virtual-circuit service.

Use of T/TCP often involves replacing existing TCP implementations. This non-evolutionary aspect of T/TCP is one of the reasons why T/TCP has not been widely adopted.

Various lessons can be learned from why T/TCP has not been widely adopted.

5.2.5 RDP

Reliable Data Protocol (RDP) is specified in [?] and [?].

RDP can be considered to be a specialized TCP, specified for particular circumstances in which some of TCP's facilities are needed.

One of the reasons why RDP has not been heavily used is because the set of specialized circumstances that it addresses do not justify the cost of implementing it. RDP allows for many options and selections, which makes its use difficult.

Various lessons can be learned from why RDP has not been widely adopted.

5.2.6 VMTP

Versatile Message Transaction Protocol (VMTP) is specified in [**?**].

VMTP can be considered to be a specialized transport protocol, targeted for what it calls the transaction model of communications.

One of the reasons why VMTP has not been heavily used is because it tries to address too broad of a software engineering-centric model. VMTP allows for many options and selections, which makes its use difficult.

Various lessons can be learned from why VMTP has not been widely adopted.

5.2.7 TCP

Transmission Control Protocol (TCP) is specified in [**?**] and [**?**].

TCP is mature, well understood and stable. Congenstion control and flow control mechanisms for TCP have been developed over the years, and incorporated into TCP implementations.

The simplest and shortest TCP interaction involves 5 PDU exchanges. For applications wishing to accomplish short and occasional communications in less than 5 PDU exchanges, ESRO is more efficient that TCP.

5.2.8 UDP

UDP (User Datagram Protocol) is specified in [**?**].

UDP is a very simple and thin layer on top of IP, which does not provide reliability or sequence ordering. Applications requiring a reliable connectionless transport need to build on top of UDP. ESRO provide an efficient and reliable transport service on top of UDP.

5.2.9 UDP Plus Ad Hoc Re-Transmissions

Various protocols have added their own specific re-transmission policies on top of UDP to make it more reliable.

Examples of such efforts include the Domain Name System (DNS) [**?**] [**?**], Network Time Protocol (NTP) [**?**], and NNTP for Usenet News Reading.

These efforts have all resulted in incomplete and limited solutions. The problems with such approaches were understood as early as 1985 [**?**].

5.3 The ESRO Protocol

The ESRO specification describes the service model, the notation and the protocol for Efficient Short Remote Operations (ESRO). The ESRO service is similar to and is consistent with other Remote Operation and Remote Procedure Call services. The emphasis of the ESRO service definition and the ESRO protocol is on efficiency. ESRO is designed specifically with wireless network (e.g. CDPD) usage in mind.

The ESRO protocol provides reliable connectionless remote operation services on top of UDP (or any other non-reliable connectionless transport service) with minimum overhead. ESRO supports segmentation and reassembly, concatenation and separation, as well as multiplexing for service users (applications).

ESRO allows for trade-offs between efficiency and reliability by specifying both 2-way handshake and 3-way handshake based protocols.

The ESRO specifications also define a notation and the services provided by an application-service element to support interactive applications in a distributed systems environment. A *Remote Operation* is invoked by one entity; the other entity attempts to perform the Remote Operation and then reports the outcome of the attempt.

Encoding mechanisms for presentation of the parameters of remote operations are outside the scope of ESRO. However, identification (tagging) of the encoding mechanism in use (e.g. XDR, BER, PER) is supported by ESRO.

A variety of applications can use the ESRO protocol. Some early applications which use ESRO include: efficient short message submission and delivery, credit card authorization, and white pages lookup.

5.3.1 Efficiency Characteristics of ESRO

The key requirement driving the design of ESRO is efficiency. Reliable transfer of a short message using TCP involves 5 transmissions at a minimum. Reliable transfer of a short message using ESRO involves only 3 or 2 transmissions.

For many applications in which optimization of efficency is desired, it is likely that elimination of the extra 2 transmissions which are inherent to TCP, justifies deviation from it and use of ESRO instead.

The efficiency premium realized by the use of ESRO rather than TCP can be very significant. For example, EMSD (a mail submission and delivery protocol that uses ESRO) can be upto 5 times more efficient than SMTP, while maintaining precisely the same level of reliability and security. The paper entitled “Efficiency Study of EMSD vs. SMTP/POP3/IMAP” [**?**] quantifies the efficiency of ESRO in comparison to traditional TCP based applications.

5.3.2 Why We Adopted the Remote Operations Model

ESRO is a reliable connectionless transport mechanism.

A reasonable question is: Why did we design ESRO’s service interface to be based on the Remote Operations model?

Many Internet protocols are ”text-based” on top of TCP. And the “here is some text” followed by “here is some more text” followed by “here is some text in response” has become the tradition of simple Internet protocols. Protocols designed on the basis of this ”text-based” approach have a good track record of acceptance throughout the Internet, primarily because they are simple to understand and simple to implement.

When efficiency matters, however, the traditional text exchange model can be better expressed by the client requesting a particular operation from the server, and the server responding with the results of that operation, thereby eliminating the traditional “text-based” chit-chat. With such an approach, the design of the protocol becomes a natural fit for the remote operations model.

For short interactions, a reliable connectionless transport mechanism and the Remote Operations model are simply the same thing. The formalism of Remote Operations is an asset that ESRO exploits.

ESRO provides a service which supports interaction of applications based on a remote operation model. A Remote Operation is invoked by one entity; the other entity attempts to perform the Remote Operation and then reports the outcome of the attempt. The ESRO protocol is designed to be able to support a large variety of applications.

5.3.3 RFC Publication of the ESRO Protocol

The ESRO protocol is completely open. It has been published as RFC-2188.

For information on how to obtain the ESRO protocol, visit the ”Base Protocol Specifications” section of http://www.esro.org/.

5.3.4 Maintenance of the ESRO Protocol via ESRO.org

ESRO.org is a public organization which enables and facilitates the development, maintenance and enhancement of protocols and related technologies which address the efficiency requirements of generic Internet applications.

Anyone interested in participating in the development of the ESRO protocol can join ESRO.org by visiting the ”Joining the ESRO.org and Related Mailing Lists” section of http://www.esro.org/.

Patent-free declarations have been made with respect to ESRO and RFC-2188 through the Free Protocols Foundation.

5.4 Use of ESRO

The ESRO protocol is designed to be able to support many applications, such as mobile messaging, directory services, micro-browsers, dictionary lookup (white and yellow pages), data sensors monitoring, telemetry, dispatch, and credit card authorization applications.

In this section we discuss various considerations that protocol developers and application designers should make when designing applications which use ESRO.

We then build on this by providing various examples.

5.4.1 Common ESRO Application Design Considerations

When designing applications which use ESRO services, a number of common issues often need to be considered. These include:

- Presentation – Syntax and Encoding
- Operation Reliability
- Idempotency – Duplication Detection
- Failure Detection and Re-Tries
- Segmentation/Re-Assembly
- Congestion and Flow Control
- Security Considerations: Authentication, Confidentiality, Authorization

We discuss each of these in some detail below.

Presentation – Syntax and Encoding

Using the remote operations model, Argument, Result and Error parameters are communicated between the invoker and the performer.

The application designer must choose and specify the particular syntax and encoding to be used.

Encoding mechanisms for presentation of the parameters of remote operations are outside the scope of ESRO. However, identification (tagging) of the encoding mechanism in use is supported by ESRO.

The choice of encoding mechanism by the developers often revolves around the efficiency requirements, the complexity of the application to be designed, the availability of tools (e.g. XML, XDR, ASN parser and compilers) and the familiarity of the developer with the tools.

ESRO can accomodate any syntax and encoding, including: plain text, XML, ASN.1 (BER, PER etc.) and XDR.

The ESRO specification introduces one notation in ASN.1 for representation of Operations. This in no way ties the use of ESRO to ASN.1. Any syntax and encoding can be used, and use of any notation for representation of Operations is not mandatory.

Operation Reliability

Different applications may need various levels of failure detection for various operations. Any one of the following three methods may be needed to accomplish the desired reliability and semantics of various aspects of applications:

- 2-Way Handshake
- 3-Way Handshake
- 3-Way Handshake with Verify

An example of the usage of all three of the above methods is provided by EMSD [**?**].

Idempotency – Duplication Detection

Some operations are idempotent in nature, that is, they can be performed more than once without causing harm. However, other operations are non-idempotent, and should therefore be performed only once. In the case of non-idempotent operations, the performer should be able to detect duplicate operations, and perform each non-idempotent operation only once.

Examples of non-idempotent operations are Submission and Delivery of messages, which should not be performed more than once. Examples of idempotent operations are enquiries of date and time, which can be performed more than once without harm.

ESRO Services do not detect duplicate invocation of operations. As a result, the application should detect duplication when the same operation instance is invoked more than once.

An example of how this can be accomplished in a coherent manner can be found in Section 4 of RFC-2524, entitled “Duplicate Operation Detection Support” [**?**].

Failure Detection and Re-Tries

Based on ESRO’s notification of Failures, the application must take necessary measures to re-try the operation.

Depending on the nature of the application, use of a general spooler which supports persistence may be reasonable.

Segmentation/Re-Assembly

ESRO provides Segmentation and Re-Assembly, as well as Concatenation and Separation, inside the protocol. However, an application may choose to provide segmentation and re-assembly above ESRO services.

The specific requirements of the application and the nature of the networks in use may justify such a design.

Congestion Control and Flow Control

The ESRO protocol includes flow control, and allows for various re-transmission timers policies to be implemented. Such policies may include exponential back-off and adaptive retransmission algorithms. These allow for the use of self-adjusting timers to determine and dynamically set timers to effectively adjust data traffic in the event the link is slower than usual due to congestion or other network or system conditions.

However, specification of retransmission timers policies was deliberately not included in the ESRO protocol. Often ESRO will be used on the edges of the Internet, where the characteristics of network behavior between the Invoker and the Performer are deterministic and stable. In such environments, a custom re-transmission timers policies is more effective than a generic one. For example, timer profiles specific to CDPD or GSM wide area networks which assume the servers are at the edges of the Internet can be tailored to optimize performance. In practice, early usages of ESRO are likely to be in such environments.

Congestion Control was also deliberately not included in the ESRO protocol, and is intended to sit on top of ESRO. This is for two reasons.

First, indications of congestion in the IP protocol suite violate layering principles, and use of such indications would have made the congestion control mechanism limited to IP. Use of ESRO over non-IP packet networks (such as GSM) is highly desirable in the short term.

Second, the indication and source congestion need not be limited to purely network resources, and indications of congestion may come from a variety of sources. In practice, ESRO based congestion control is primarily server driven. Design of ESRO applications should include Congestion Control mechanisms.

An example of how Congestion Control on top of ESRO may be designed can be found in SubmissionControl and DeliveryControl sections of RFC-2524, [**?**]. Such control facilities and design makes use of ESRO and EMSD well suited for the End-To-End Internet usage model.

Security

Different applications may need various levels of security facilities. It is the requirements and scope of the individual application that drives the nature of security facilities that are appropriate to use.

In many cases, the key required security facilities are Authentication, Confidentiality and Authorization. The trade-offs between complexity, efficiency and reliability are best made on a case-by-case basis.

In the specific case of Efficient Mail Submission & Delivery (EMSD) [**?**], the existing mainstream Internet security mechanism of Pretty Good Privacy (PGP) can be used to provide end-to-end security facilities.

As an underlying general facility, ESRO itself does not provide any generalized security facilities. At some point in time, it makes good sense to create a common security framework for LEAP which is consistent with the broader Internet security framework.

We will start incorporating common security features when they become widespread in the broader Internet. When Network Layer Security, Transport Layer Security, PKI, etc. are widespread, they will be incorporated as common facilities within LEAP.

5.5 Example Applications

A variety of applications can make use of the ESRO protocol. In this section we mention and provide references to a number of ESRO based applications. Based on the broadness of their usage and intended audience, we categorize them as either "Horizontal Applications" or "Vertical Applications."

5.5.1 Horizontal Applications

Various efficient horizontal applications have been developed or are being developed using ESRO. These include:

- EMSD: Mobile and Wireless Messaging, Modern Two-Way Paging
- EHTD: Efficient Micro-Browsing
- E-DICT: Efficient Dictionary Lookup

A brief description of each is provided below.

5.5.2 EMSD: Efficient E-Mail

One of the efficient application layers built on top of ESRO is called Efficient Mail Submission & Delivery, or EMSD. EMSD is the component of LEAP that addresses the Mobile Messaging application.

EMSD is highly optimized for the submission and delivery of short (typically 4 kilobytes or less) Internet e-mail messages, and is therefore extremely well suited to the wireless environment. EMSD improves on existing messaging protocols by optimizing the exchange between the server and the end-user device, both in terms of the number of bytes transferred, and in terms of the number of transmissions. For more information on EMSD see the article *EMSD: The LEAP E-Mail Component* within *The LEAP Manifesto*, or visit the EMSD website at http://www.emsd.org/.

5.5.3 EHTD: Efficient Web Browsing

The Efficient Hyper Text Delivery (EHTD) layer is a hypertext transfer protocol which is optimized for the efficient transfer of short markup pages. EHTD is the component of LEAP which facilitates web browsing. Along with EMSD, EHTD also benefits from the reliable efficient services of ESRO. A multiplicity of efficient markup languages can be used in conjunction with EHTD. Development of the EHTD protocol is currently in progress.

5.5.4 Other Efficient Horizontal Applications

Various other efficient application protocols are either under development, or anticipated for future development. One of these is the Efficient Dictionary protocol, or E-DICT, which will enable efficient access to dictionaries and other lookup data structures. A starting point for the E-DICT protocol is currently being created. In developing E-DICT, we intend to build on the existing work already done in the context of the DICT protocol.

We anticipate that additional protocols will be needed for a variety of future applications, not all of which can be foreseen at this time. These applications will include such things as efficient implementations of ESRO-based instant messaging, chat, white pages, and others.

5.5.5 Vertical Applications

Various efficient vertical applications have been developed or are being developed using ESRO. These applications include:

- Wireless credit card verification
- Wireless automated teller machine (ATM)
- Dispatch applications
- Telemetry
- Wireless T.V. ratings
- Vehicle tracking (GPS applications etc.)

There are a set of common architectural characteristics to most of these vertical applications. These common characteristics are illustrated in Figure 5.1. In this figure, the box labeled "Thin Reliability Layer" represents the position of ESRO.

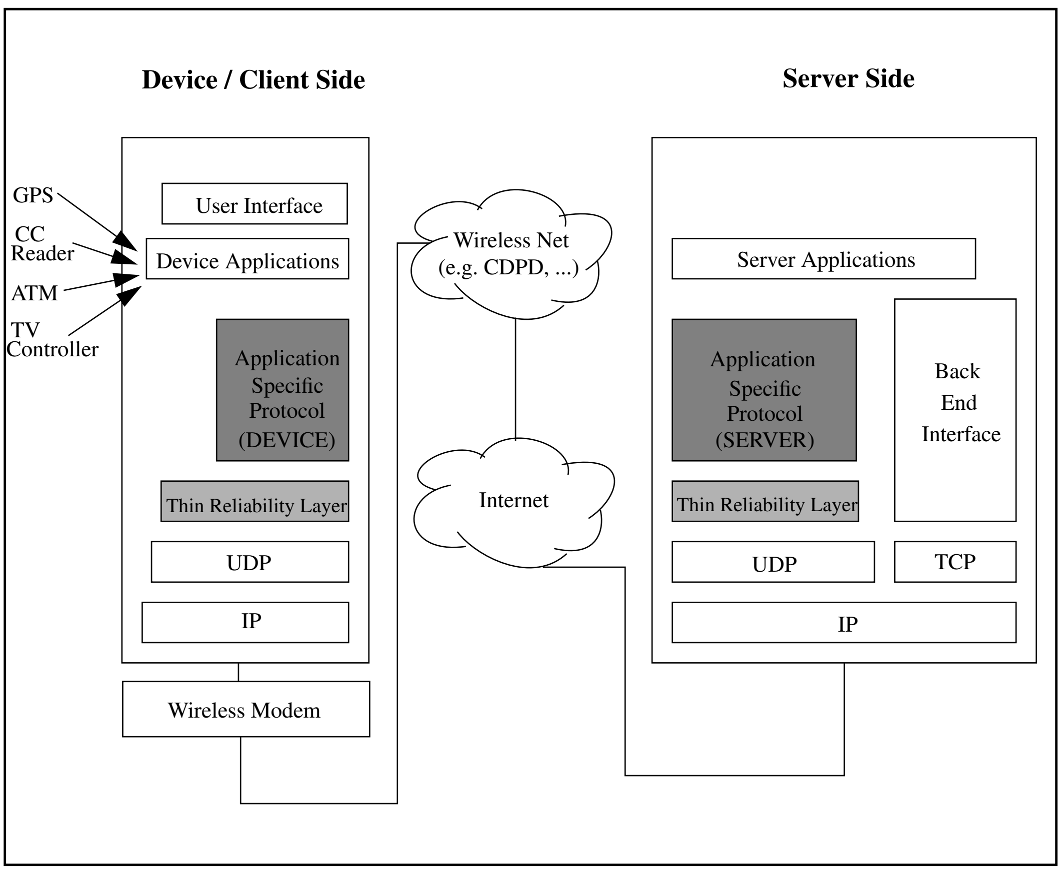


Figure 5.1: Anatomy of a Vertical Wireless Application

The combination of ESRO as the efficient transport mechanism, the systems modeling represented in Figure 5.1, and the methods described in Section 5.4.1 can greatly facilitate and expedite development of efficient vertical applications.

5.6 Existing Implementations of ESRO

A Reference Implementation of ESRO in the form of free software in open-source format is being made available at http://www.MailMeAnywhere.org/.

We expect to have the ESRO protocol engine software components subject to the GNU General Public License available at this location by September 2000.

On the device side, software will be made available for most generic platforms, including PalmOS, EPOC, WindowsCE, Windows9X, and Windows-NT. On the server side, software will be available on Solaris, Linux and NT.

In addition to open-source format, the ESRO Reference Implementation will also be made available for the above mentioned platforms as a ready-to-use Toolkit.

For a current list of ESRO free and commercial software products, visit the ”Products and Services” section of http://www.esro.org.

5.6.1 ESROS Application Programming Interface

The “ESRO API Specification” [**?**] defines the ESRO Application Programming Interface (API). This specification maps ESRO service primitives into C language function calls. This document is available at
http://www.esro.org/documents/API.html.

Chapter 6

EMSD: The LEAP E-Mail Component

6.1 Introduction

This article provides a description of the **Efficient Mail Submission & Delivery** protocol, or **EMSD**. EMSD is the e-mail component of the LEAP family of protocols.

The entire family of LEAP protocols has been designed with efficiency as a primary requirement, and each component protocol brings efficiency and functionality benefits to the users of miniaturized mobile devices. In particular, EMSD brings these efficiency benefits to the e-mail, or Mobile Messaging, application.

Mobile Messaging is just one of several applications for which there is a high efficiency premium in the mobile and wireless arena. Other applications which demand efficiency in the mobile environment are such things as web browsing, dictionary look-up, etc. From an industry-building perspective, however, e-mail is the most critically important of all wireless and mobile data communications applications. Mobile Messaging provides the user with a unique communications facility: the immediate delivery of important and time-critical information to the mobile recipient, wherever and whenever he/she happens to be. This is a capability which is not provided by phone, Fax, or any other tool in the modern user's array of communications options.

In fact, this capability represents the principal value of wide-area wireless networks to the end user, and this is why e-mail remains the dominant application for wide-area wireless networks. Furthermore, miniature hand-held mobile devices are extremely well-suited to the e-mail application. The same is not true for other applications, such as web browsing.

For all these reasons, e-mail is the key industry-building application, and this is why EMSD represents the right starting point for the LEAP family of protocols.

6.1.1 Terminology

Throughout this article, we will make use of the following terms and definitions:

- **MTS**. Mail Transfer System.
- **Mail Submission**. Mail submission refers to the process of putting mail into the Mail Transfer System (MTS).
- **Mail Delivery**. Mail delivery refers to the process of the MTS putting mail into a user's final mailbox.
- **SMTP**. Simple Mail Transfer Protocol.

- **MUA**. Mail User Agent.
- **EMSD-P**. EMSD Protocol.
- **EMSD-FS**. EMSD Format Standard.
- **EMSD-UA**. EMSD User Agent.
- **EMSD-SA**. EMSD Server Agent.

6.2 Existing Internet Mail Submission and Delivery

Mail transmission in the Internet did not arise as a result of well-planned engineering processes; rather, it grew and evolved in a more organic way.

At present, most mail submission and delivery throughout the Internet is done by means of the Simple Mail Transfer Protocol, or SMTP. SMTP was originally defined as a message *transfer* protocol – that is, a means to route (if necessary) and deliver mail by putting finished (i.e. complete) messages in a mailbox. Originally, users connected to servers from terminals, and all processing occurred on the server. Today, a split-MUA (Mail User Agent) model is common, in which MUA functionality occurs both on the user's own system, and the server.

In the split-MUA model, the process of getting a message to the user is accomplished by access to a mailbox on the server, using protocols such as POP and IMAP. Also, in the split-MUA model, the user's access to his/her message is based on a "Message Pull" paradigm, in which the user is required to explicitly poll his/her mailbox to retrieve mail. Message delivery based on a "Message Push" paradigm, in which mail is delivered directly to the user without polling, is presently not supported.

Despite its original definition as a message *transfer* protocol, in the split-MUA model, SMTP is often used for message *submission*. The widespread use of SMTP for submission has become a reality, regardless of whether this is a good or a bad thing.

6.3 Overview of EMSD

EMSD is a messaging protocol that is highly optimized for the submission and delivery of short Internet e-mail messages. The EMSD protocol addresses all the shortcomings in the existing Internet mail system described in the previous section. EMSD properly supports the Message Push mode of operation, and it provides an alternative mechanism to SMTP for message submission. And most important of all, it does this with a major emphasis on efficiency.

6.3.1 Protocol Layering

As shown in Figure 6.1, the LEAP protocols are layered. The lower layer, called Efficient Short Remote Operations (ESRO), provides efficient reliable connectionless transport services which can be used by a variety of applications. For example, in addition to Mobile Messaging services, ESRO can also be used as a transport service for credit card verification applications and efficient micro browsers.

EMSD is built on top of ESRO. The reliability requirements for message submission and message delivery in EMSD are the same as for existing e-mail protocols. The EMSD protocol provides reliable connectionless mail submission and delivery services.

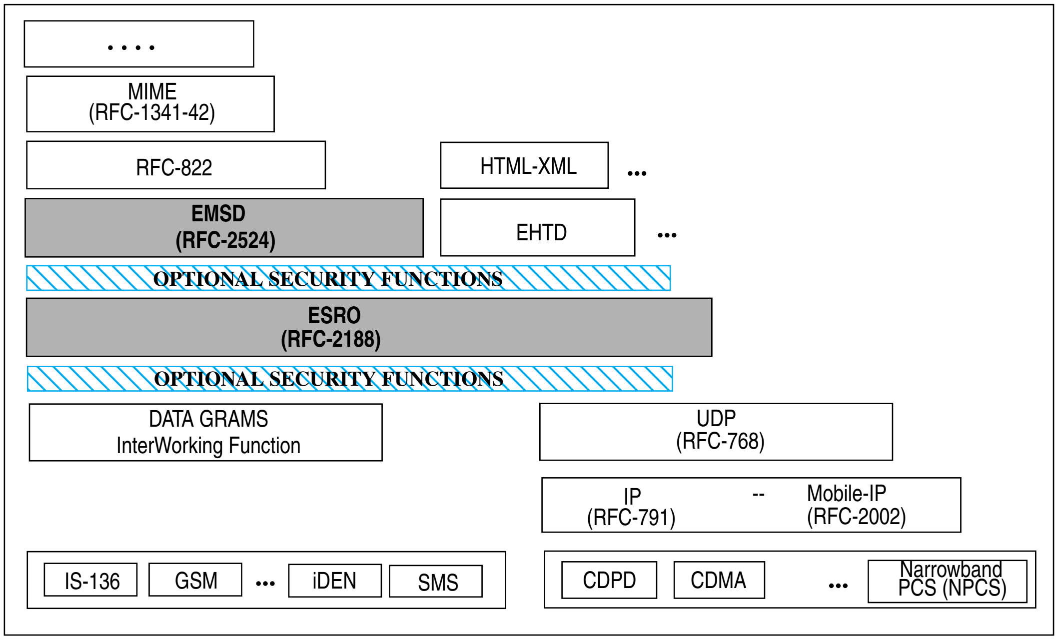


Figure 6.1: LEAP Protocol Stack

6.3.2 EMSD Protocol Components

EMSD consists of two independent components: the **EMSD Format Standard**, and the **EMSD Protocol**. These two components provide the following functions:

1. **EMSD Format Standard (EMSD-FS)**

   EMSD-FS is a non-textual form of compact encoding of Internet e-mail (RFC-822) messages, which facilitates efficient message transfer. EMSD-FS is used in conjunction with the EMSD-P (described below), but is not in any way a general replacement for RFC-822. EMSD-FS defines a method of representation of short interpersonal messages. It defines the ”Content” encoding (Header + Body). Although EMSD-FS contains end-to-end information, its scope is purely point-to-point. EMSD-FS relies on EMSD-P for the transfer of the content to its recipients.

2. **EMSD Protocol (EMSD-P)**

   EMSD-P is responsible for wrapping a limited size EMSD-FS message in a point-to-point envelope, and submitting or delivering it. EMSD-P performs the envelope encoding. EMSD-P relies on the services of Efficient Short Remote Operations (ESRO) as specified in RFC-2188 [**?**] for transporting the point-to-point envelope. Some of the services provided by EMSD-P include: message originator authentication, and optional message segmentation and re-assembly. EMSD-P is expressed in terms of abstract services using the ESRO notation.

Together, the EMSD Protocol and Format Standard define the protocols used to transfer messages between an EMSD Server Agent (EMSD-SA), for example a Message Center, and an EMSD User Agent (EMSD-UA), for example a Two-Way Pager.

Figure 6.2 illustrates how EMSD defines the communication between a specific EMSD-UA and a specific EMSD-SA. The Message Transfer System may include a number of EMSD-SAs, and each EMSD-SA may have any number of EMSD-UAs with which it communicates.

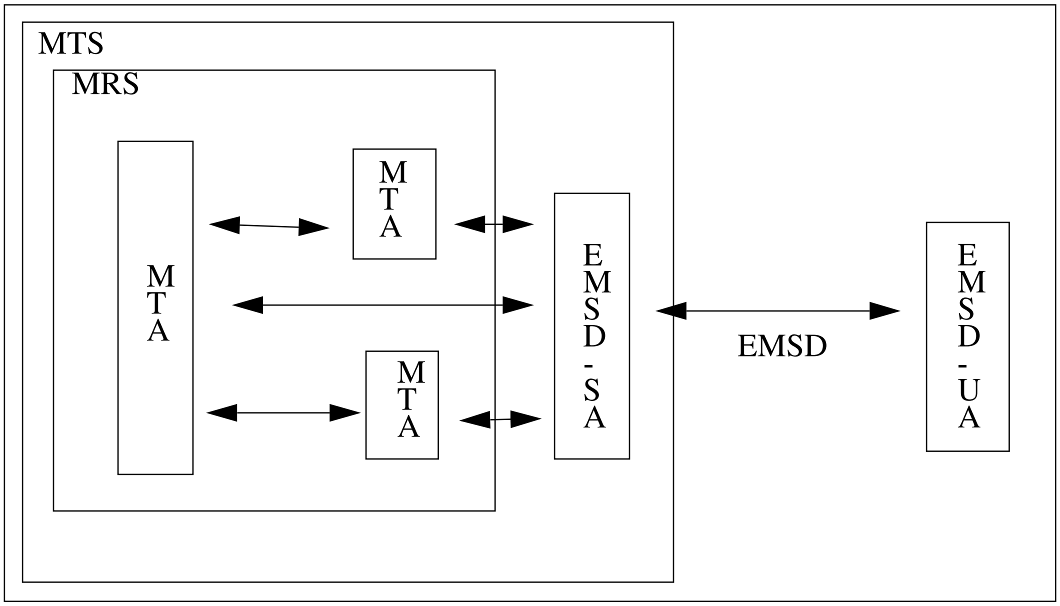


Figure 6.2: Efficient Mail Submission and Delivery Protocol

It is important to note that EMSD-P and EMSD-FS are not end-to-end, but instead focus on the point-to-point transfer of messages. The two points being referred to are the EMSD-SA and EMSD-UA. EMSD-P functions as an element of the Internet mail environment, providing end-to-end services (i.e. EMSD-User to any other Messaging Originator or Recipient).

The EMSD services use the Efficient Short Remote Operations (ESRO) services. They also use the Duplicate Operation Detection Support Functions. These functions guarantee that an operation is performed no more than once.

6.3.3 Efficient Short Remote Operations (ESRO)

The EMSD protocol specifications define the protocols between the EMSD Device and the EMSD Server. EMSD is built on top of, and requires the services of, ESRO (Efficient Short Remote Operations). This EMSD requirement was the major motivation for the development of ESRO; however, ESRO has been developed to be independent of EMSD.

ESRO defines a notation and the services provided by an application-service element to support interactive applications in a distributed systems environment. The scope of ESRO services is not limited to EMSD. ESRO is designed to be able to support other applications, such as finger/limited directory service.

The ESRO protocol provides reliable connectionless remote operation services on top of UDP (or any other non-reliable connectionless transport service) with minimum overhead. ESRO supports segmentation and reassembly, concatenation and separation, as well as multiplexing for service users (i.e. applications).

The ESRO service is similar to and is consistent with other Remote Procedure Call services. The major emphasis of the ESRO service definition and the ESRO protocol is on efficiency. ESRO has been designed specifically with wireless network (e.g. CDPD) usage in mind.

The service model, the notation, and the protocol for ESRO are fully specified in RFC-2188 [**?**]. The EMSD Protocol uses ESRO to accomplish reliable connectionless mail submission and delivery.

For more information on ESRO, see the article entitled *ESRO: A Foundation for the Development of Efficient*

Protocols within *The LEAP Manifesto*, or visit the ESRO website at http://www.esro.org/.

6.3.4 Anticipated Uses of EMSD

Any network or network operator which faces significant bandwidth and capacity limitations can benefit from the use of EMSD. Any user of a network who must bear high costs for measured network usage can benefit from the use of EMSD.

The initial use of EMSD is expected to be primarily to provide Mobile Messaging services over IP-based wireless networks. However, EMSD can also function as an adjunct to Mail Access Protocols for ”Mail Notification Services.”

Mail submission and delivery take place at the edges of the network. It is likely that multiple mail submission and delivery protocols will be developed, each addressing the specific requirements of a particular user's environment. Such diversity on the edges of the network is beneficial, and with the right protocols, this diversity does not adversely affect the integrity of the mail transfer system. EMSD is the basis for the mail submission and delivery protocol to be used when the user's environment demands efficiency.

6.4 EMSD Design Goals and Requirements

The EMSD protocols have been designed to accomplish three high-level goals:

1. Define the new ”world” of Efficient Mail Submission & Delivery
2. Define a remote operations service that can handle messaging and other standard networking applications
3. Make EMSD an extension of the existing Internetworking world

Based on these goals, EMSD has been designed to satisfy the following design requirements:

1. Support the submission of short mail messages with the same (or better) level of functionality as the existing Internet mail protocols.
2. Support the delivery of short mail messages with the same (or better) level of functionality as the existing Internet mail protocols.
3. Function as an extension of the existing mainstream Internet mail.
4. Minimize the number of transmissions.
5. Minimize the number of bytes transmitted.
6. Be quick: minimize the latency of message submission and delivery.
7. Provide the same level of reliability (or better) as the existing e-mail protocols.
8. Accommodate varying sizes of messages: the size of a message may determine how the system deals with the message, but the system must accommodate it.
9. Be power efficient and show respect for mobile platform resources, including memory and CPU levels, as well as battery power longevity. In other words, be client-light and server-heavy.
10. Be highly extensible. Different users will demand different options, so the solution cannot require every feature to be a part of every message. Likewise, usage will emerge that is not currently recognized as a requirement. The solution must be extensible enough to handle new, emerging requirements.

11. Be secure. Provide the same level of security (or better) as the existing e-mail protocols. Content confidentiality, originator/recipient authentication, and message integrity must be available options to users.

12. Be easy to implement: re-use existing technology as much as possible.

The EMSD protocols make extensive use of existing technology, including:

- RFC-822
- ASN.1
- Basic Encoding Rules
- Internet mail

By using these established technologies, the design of EMSD avoids the expense and other problems associated with "re-inventing the wheel." The above technologies have been thoroughly tested, and have proven to be reliable solutions for the problems they address (e.g. message format, reliable message delivery, encoding and compacting). The EMSD specifications cater to users who enjoy the advantages of this new technology, but at the same time want to be connected to the rest of the existing Internet e-mail world. Figure 6.3 shows how the Global and EMSD worlds complement one another.

The Internet e-mail community is shown in the lower half of the figure. This world is connected to the EMSD Internet e-mail system.

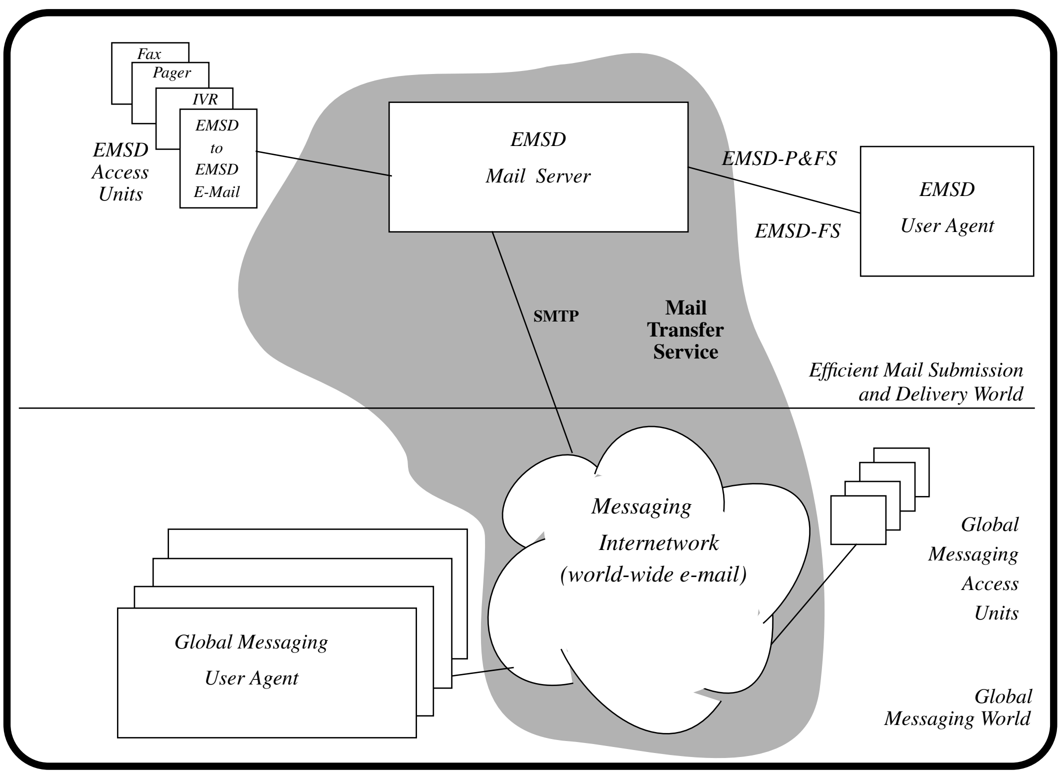


Figure 6.3: EMSD World and Global Messaging World

6.5 Rationale for Key Design Decisions

This section summarizes the rationale for the key design decisions that were made while developing the EMSD protocols.

6.5.1 Deviation from the SMTP Model

SMTP is the main mail transport mechanism used throughout the Internet. It is widely deployed and well understood by many engineers who specialize in Internet e-mail. For these reasons, protocols based on or derived from SMTP or more likely to become widely deployed throughout the Internet.

However, SMTP is highly inefficient for the transfer of short messages. SMTP is inefficient both in terms of the number of transmissions, and in terms of the number of bytes transmitted. Even when fully optimized with PIPELINING, SMTP remains significantly inefficient.

The submission of a short message using SMTP requires 15 transmissions. The submission of a short message with SMTP and PIPELINING requires 9 transmissions. The submission of a short message with EMSD (EMSD-P and ESRO) requires only 3 transmissions (in a typical case).

The key design requirement of EMSD is efficiency. Because of the 3 fold (at least) gain in efficiency, this justifies the deviation from the SMTP model.

Efficiency Comparison of SMTP and EMSD

Table 6.1 shows the number of N-PDUs exchanged for the transfer of a short Internet e-mail when using SMTP, SMTP with PIPELINING, QMTP, and EMSD. The names used for identifying the PDUs are informal names.

	SMTP	**SMTP + Pipelining**	**QMTP , QMQP**	**EMSD**
Client:	SYN	SYN	SYN	Submit.Req
Server:	SYN ok	SYN ok	SYN	Submit.Resp
Client:	HELLO	HELLO	message	ack
Server:	ok	PIPELINING	accept close	
Client:	MAIL	MAIL RCPT DATA	close	
Server:	ok	ok		
Client:	RCPT	message QUIT		
Server:	ok	accept ok close		
Client:	DATA	close		
Server:	ok			
Client:	message			
Server:	accept			
Client:	QUIT			
Server:	ok close			
Client:	close			

Table 6.1: Comparison of EMSD to Other Protocols

6.5.2 Use of ESRO Instead of TCP

In order to provide the same level of reliability that the existing e-mail protocols provide for short messages, it is clear that a reliable underlying service is needed. UDP [**?**], by itself, is clearly not adequate.

Use of TCP however, involves three phases:

1. Connection Establishment
2. Data Transfer
3. Disconnect

The reliable transfer of a short message using TCP involves a minimum of five transmissions, as is the case with QMTP.

Again, the key design requirement of EMSD is efficiency. Therefore deviation from TCP is justified, because this eliminates the two extra transmissions that are an inherent characteristic of TCP.

The ESRO protocol, as specified in RFC-2188 [**?**], provides reliable connectionless remote operation services on top of UDP [**?**] with minimum overhead. ESRO supports segmentation and reassembly, concatenation and separation.

The reliable transfer of a short message using ESRO involves 3 transmissions, as is the case with EMSD-P.

6.5.3 Use of the Remote Procedure Call (RPC) Model

Many Internet protocols are ”text-based.” On the other hand, few Internet protocols are RPC-based. Protocols designed on the basis of the ”text-based” approach have a better track record of acceptance throughout the Internet.

However, considering that message submission and delivery in EMSD involves no more than two data exchanges, the text-based model becomes the same as an operation. Furthermore, the RPC model is the natural way of using ESRO.

6.5.4 Use of ASN.1

In order to minimize the number of bytes transfered, efficient encoding mechanisms are needed.

Among today’s encoding mechanisms, ASN.1 has the unique feature of separating the abstract syntax from the encoding rules. By selecting ASN.1 as the notation used for expressing EMSD’s information objects, EMSD has the flexibility of using the most efficient encoding rules, such as Packed Encoding Rules (PER), when they are available.

Efficient encoding can always be better performed when the syntax of the information is known. In general, encoding and compression techniques which use the knowledge of the syntax of the information produce better results than those compression techniques that work on arbitrary text.

6.6 Relationship of EMSD to Other Mail Protocols

EMSD is designed to be a companion to existing Internet mail protocols. It is designed to fit within the many protocols already in use for messaging, as well as those already in use for networking.

Figure 6.4 shows how EMSD fits in with the other major messaging protocols. The RFCs referenced in the figure are current at the time of this writing, but could be updated or made obsolete at any time.

The various Internet mail protocols provide different sets of capabilities for mail processing. Table 6.2 summarizes the capabilities of SMTP, IMAP, POP and EMSD in the following areas of functionality:

- Mail Submission

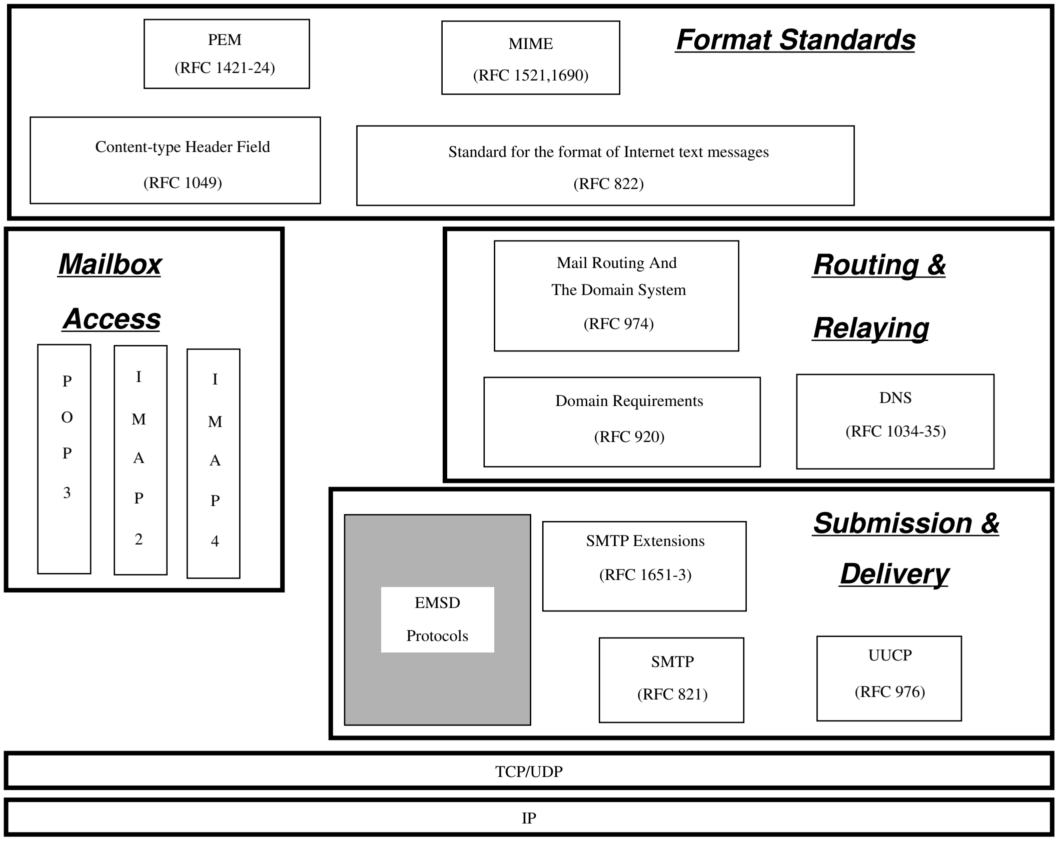


Figure 6.4: Messaging Communication Stack and EMSD

- Mail Delivery
- Mail Routing (Relay)
- Mail Retrieval
- Mailbox Access
- Mailbox Synchronization

The number of X's in each cell of the table denotes the extent to which a particular function is supported by a particular protocol.

Table 6.2 clearly demonstrates that combinations of these protocols can be used to complement one other in providing rich functionality to the user. For example, a user interested in highly mobile messaging functionality can use EMSD for the submission and delivery of time-critical and important messages, and use IMAP for comprehensive access to his/her mail-box.

Protocol Function	**SMTP**	**IMAP**	**POP**	**EMSD**
Submission	XX			XXX
Delivery	XXX			XXX
Relay (Routing)	XXX			
Retrieval		XXX	XXX	XX
Mailbox Access		XXX	X	
Mailbox Sync.		XXX		

Table 6.2: Messaging Protocol Functionality

6.7 Obtaining the EMSD Protocols

For complete instructions on how to obtain the EMSD protocols, visit the ”Base Protocol Specifications” section of http://www.emsd.org/.

Chapter 7

Efficiency of EMSD

Preface

This article was originally published in October 1996. It is being included in the Manifesto, essentially unchanged from its original form.

7.1 Introduction

We provide here an overview of both SMTP and EMSD, to compare and contrast their features and to lay the groundwork for analysis of the experimental results in Sections **??** and **??**.

SMTP

According to RFC 821[**?**], the objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently. The SMTP design is based on the following model of communication:

As the result of a user mail request, the sender-SMTP establishes a two-way transmission channel to a receiver-SMTP, which may be either the ultimate destination or an intermediate.

Following this, the sender-SMTP sends a MAIL command indicating the sender of the mail. If the receiver-SMTP can accept mail it responds with an OK reply. The sender-SMTP then sends a RCPT command identifying a recipient of the mail. If the receiver-SMTP can accept mail for that recipient it responds with an OK reply; if not, it responds with a reply rejecting that recipient (but not the whole mail transaction). The sender-SMTP and receiver-SMTP may negotiate several recipients. When the recipients have been negotiated, the sender-SMTP sends the mail data, terminating with a special sequence. If the receiver-SMTP successfully processes the mail data it responds with an OK reply. Note that the dialog is purposely lock-step, one-at-a-time.

SMTP provides two mechanisms for the transmission of mail: directly from the sending user's host to the receiving user's host when the two host are connected to the same transport service, or via one or more relay SMTP-servers when the source and destination hosts are not connected to the same transport service. The mail commands and replies have a rigid syntax. Replies also have a numeric code.

Thus it can be seen that for the exchange of any one message with SMTP, a number of transactions must be completed. EMSD attempts to improve efficiency by cutting down on this number and simplifying the process for the case of short messages.

7.1.1 Efficient Mail Submission & Delivery

The EMSD specifications define the protocols between an EMSD Device and an EMSD Server. EMSD requires ESROS (Efficient Short - Remote Operation Services) [**?**]. The EMSD-P&FS consist of two independent components: Efficient Mail Submission & Delivery Protocol (EMSD-P) and EMSD Format Standards (EMSD-FS).

EMSD-FS is responsible for defining the format of a limited size interpersonal message. It defines the "Content" encoding (Header + Body) and the end-to-end envelope. It relies on EMSD-P for the transfer of the content to its recipients.

EMSD-P is responsible for wrapping a limited size message in a point-to-point envelope and submitting or delivering it. EMSD-P performs the envelope encoding and relies on the services of ESROS for transporting the envelope. Some of the services of EMSD-P include message originator authentication and optional message segmentation and re-assembly. The Efficient Mail Submission & Delivery Protocols are designed with three high level goals:

- Define the new "world" of Efficient Mail Submission & Delivery
- Define a remote operations service that can handle messaging and other standard networking applications
- Make Efficient Mail Submission & Delivery an extension of the existing internetworking world

These goals will prevent, whenever possible, the expense and associated problems of "re-inventing the wheel." The EMSD Protocols make heavy use of existing technology:

- RFC-822
- ASN.1
- Basic Encoding Rules
- X.400 and Internet e-mail

These technologies have been thoroughly tested and have proven to be reliable solutions for the problems they address (e.g. message format, reliable message delivery, encoding and compacting). The EMSD Specifications allow for users who enjoy the advantages of this new technology and at the same time want be connected to the rest of the existing messaging world.

7.2 Study Overview

We have chosen to compare the efficiency of using EMSD to the efficiency obtained by other submission and delivery protocols in this study. While it is debatable whether EMSD can be placed at the same level as the test protocols, we nonetheless feel that a study such as this is quite useful and provides a common denominator to discuss various aspects of EMSD performance.

The experiments cover both submission (from a mobile unit) and delivery (to a mobile unit). Under submission we looked at comparing EMSD and SMTP. The delivery experiments tested EMSD vs. SMTP, POP, and IMAP. In all cases a single uniform test message was relayed between two devices (a recipient or sender, and a mail server) and the data traffic recorded. Although you cannot compare EMSD directly to any one messaging protocol, because each protocol is designed to perform a specific function, you can compare the results obtained by each messaging solution. The following table illustrates the functions supported by each protocol. Note that EMSD is the most like SMTP.

Protocol	Submission	Delivery	Relay	Retrieval	Mailbox Access	Mailbox Sync
SMTP	X	X	X	X		
IMAP				X	X	X
POP				X		
EMSD	X	X		X		

Table 7.1: Messaging Protocols

7.3 Submission

Please refer to Figure 7.1 below, which shows the setup for the following two submission experiments in detail. The experimental setup involves:

At Site One:

- A ”sender”: Toshiba Laptop running Windows 3.1 and Chameleon Winsock TCP/IP stack from NetManage
- A ”receiver”: Sun Sparc running Solaris 2.4
- A ”mail server”: Sun Sparc running Solaris 2.4
- A sniffer that monitors packet movement at the juncture of the above three devices, recording two-way traffic

At Site Two:

- A Message Test Center: Sun Sparc running Solaris 2.4.

The two setups are linked to each other over a number of routers across the Internet.

In both cases below, we are interested exclusively in analyzing the recorded data between the sender laptop and the Unix mail server (in the case of SMTP submission), or the EMSD Message Test Center (in the case of EMSD submission). Thus the ”receiver” shown below, although necessary to submit the message, does not enter into our study picture directly.

7.3.1 SMTP Submission from PC to Unix

Message Submission Process

Message was submitted from the Laptop to the Unix mail server. To submit a message from the laptop, Netscape's Mail User Agent on Windows 3.1 was utilized. From the file menu on Netscape, ”New Mail Message” was selected, popping up a mail window. The message was typed in, a recipient (the ”receiver” in Figure 7.1) was specified, and ”Send” was then clicked. The sniffer recorded the exchange of data between the sender and the mail server that was ¡nwestmail.nwest.airdata.com¿ implementing sendmail.

Protocol Trace

The following is the protocol trace recorded by the sniffer. After TCP synchronization and acknowledgment, a virtual circuit is established between the sender's Netscape Mail User Agent and sendmail on the mail server, and data is exchanged after specifying the sender and recipient addresses.

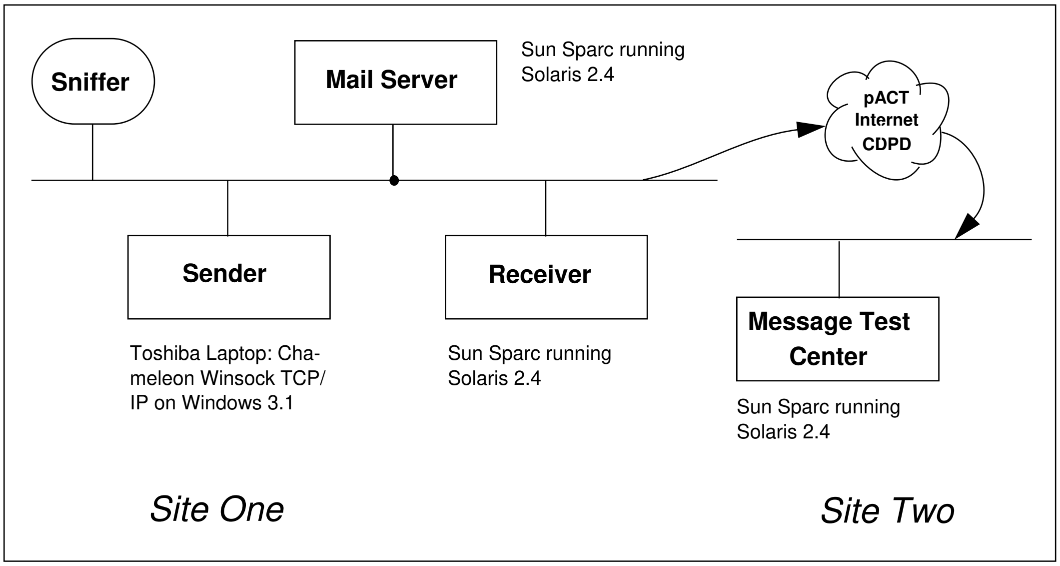


Figure 7.1: Experimental Setup for Submission

```
IP_PDU      MailServer             UA   DATA   TCP     IP
--------------------------------------------------------------------
1  TCP   .<------- TCP SYN -------- .     0     24     44
2  TCP   . ------- TCP SYN ack ---->.     0     24     44
3  TCP   .<------- Push ACK ------- .     0     20     40  (128)
4  SMTP  . ----220 server ready --->.   116    136    156
5  TCP   .<------- Push ACK ------- .     0     20     40  (196)
6  SMTP  .<------- HELO <client>--- .    36     56     76
7  SMTP  . ----250 server Hello --->.   111    131    151
8  TCP   .<------- Push ACK ------- .     0     20     40  (267)

9  SMTP  .<--MAIL FROM:<sender> --- .    32     52     72
10 SMTP  . ----250 ... Sender ok--->.    39     59     79
11 TCP   .<------- Push ACK ------- .     0     20     40  (191)
12 SMTP  .<--RCPT TO:<rcpt>-------- .    33     53     73
13 SMTP  .<----250...Recipient ok-- .    45     65     85
14 TCP   .<------- Push ACK ------- .     0     20     40  (198)

15 SMTP  .<------ "DATA" ---------- .     6     26     46
16 TCP   . ------- ACK ------------>.     0     20     40   (86)
17 SMTP  . ----354..end with "."--->.    50     70     90
18 TCP   .<------- Push ACK ------- .     0     20     40  (130)
19 SMTP  .<--Mail header+body ----- .   437    457    477
20 SMTP  .<------ . --------------- .     5     25     45
21 TCP   . ------- ACK ------------>.     0     20     40  (562)

22 SMTP  . ------- 250 Ok --------->.     8     28     48
23 TCP   .<------- Push ACK ------- .     0     20     40
24 TCP   .<------- Push Reset ----- .     0     20     40  (128)
```

```
----------------------------------------------------------------------
```

Measurement Results

```
Total IP Packet bytes: 1886
Message Length (header + body): 437
Total Overhead (TCP header + IP header): 1449
3.1.4 Message as Received
Message-ID: <32249501.46FD@airdata.com>
Date: Wed, 28 Aug 1996 11:50:41 -0700
From: Jia-bing Cheng <jcheng@airdata.com>
Organization: AT&T Wireless Services
X-Mailer: Mozilla 2.0 (Win16; U)
MIME-Version: 1.0
To: j.cheng@pocketnet.net
Subject: test3
X-URL: file:///c:/netscape/jbc.htm
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

123456789012345678901234567890
12345678901234567890
1234567890
```

7.3.2 EMSD Submission from PC to Unix

Message Submission Process

The message was submitted from the laptop using Neda's EMSD Mail User Agent version 0.9 on Windows 3.1, to Neda's EMSD Message Test Center. EMSD-Pine version 3.91 was used to submit the message from the laptop. After invoking Pine, and typing "C", a new message was composed and then sent via ¡CTRL X¿. A direct connection was then established between the EMSD Mail User Agent on the laptop and the EMSD Message Test Center, and the message was relayed. The sniffer recorded exchange of data between the sender and Neda's EMSD Message Test Center which was ¡emsd.neda.com¿ implementing ESROS.

Protocol Trace

The following is the protocol trace recorded by the sniffer. As compared to the SMTP protocol trace in section 7.3.1, you can see the exchange is quite brief.

```
IP_PDU  MailServer            UA   DATA   UDP     IP
---------------------------------------------------------------
1  UDP   .<--Invoke header+body --- .   206    214    234
2  UDP   . -----Response ---------->.    15     23     43
3  UDP   .<------- Ack ------------ .     2     10     30
---------------------------------------------------------------
```

Measurement Results

Total IP Packet bytes: 307

Message Length (header + body): 206

Total Overhead (EMSD header + UDP header + IP header): 101

Total IP bytes in the case of EMSD submission as compared to SMTP submission is down by a factor of 6; the header count is down by a factor of 2.6; and total overhead is down by a factor of 14, representing major savings in data traffic.

Message as Received

The # text below is provided as comments and does not appear in the received message.

```
P.!.0. ... 0..0z@."333. 333"<test1@emsd. neda.com>          # FROM:
010/@)Jia-bing-pn Cheng<j.cheng@pocketnet.net>             # RCPT:
......test4                                                # Subject:
....text/plain; charset=us-ascii                           # content-type
0C.A                                                       #

123456789012345678901234567890.                            # BODY:
12345679801234567890.
1234567890
```

7.4 Delivery

Similar to the submission experiments above, we also conducted analogous delivery tests. The first experiment on SMTP delivery is essentially the complement of the SMTP submission experiment described above, and uses the same setup as in Figure 7.1. The second and third delivery experiments are with POP and IMAP servers and are described in their corresponding sections below. The final experiment is on EMSD delivery and also uses the same setup as in Figure 7.1. We then compare the performance of EMSD delivery versus the other three delivery methods.

7.4.1 SMTP Delivery from Unix to Unix

Please refer to Figure 7.1 above for this experiment.

Message Delivery Process

The message was delivered to the Unix receiver from the Unix mail server. Both were implementing sendmail and the message was delivered via standard SMTP. The sniffer recorded the exchange of data between the recipient and the mail server, which was ¡nwestmail.nwest.airdata.com¿.

Protocol Trace

The following is the protocol trace recorded by the sniffer. After TCP synchronization and acknowledgment, a virtual circuit is established between the recipient's Mail User Agent and sendmail on the mail server, and data is exchanged after specifying the sender and recipient addresses.

```
IP_PDU  MailServer           bluejeans   DATA   TCP     IP
-------------------------------------------------------------------
1  TCP   .<------- TCP SYN -------- .     0     20     40
2  TCP   . ------- TCP SYN ack ---->.     0     20     40
3  TCP   .<------- Push ACK ------- .     0     20     40

4  SMTP  . ----220 server ready --->.   116    136    156
5  SMTP  .<------- HELO <client>--- .    16     36     56
6  SMTP  . ----250 server Hello --->.    95    115    135
7  SMTP  .<--MAIL FROM:<sender> --- .    29     49     69
8  SMTP  . ----250 ... Sender ok--->.    39     59     79
9  SMTP  .<--RCPT TO:<rcpt>-------- .    44     64     84
10 SMTP  .<----250...Recipient ok-- .    57     77     97

11 SMTP  .<------ "DATA" ---------- .     6     26     46
12 TCP   . ------- ACK ------------>.     0     20     40
13 SMTP  . ----354..end with "."--->.    50     70     90
14 SMTP  .<--Mail header+body ----- .   301    321    341
15 TCP   . -------- ACK ----------->.     0     20     40
16 SMTP  .<------ . --------------- .     5     25     45
17 TCP   . ------- ACK ------------>.     0     20     40
18 SMTP  . ------- 250 Ok --------->.     8     28     48

19 SMTP  .<------- QUIT ----------- .     6     26     46
20 SMTP  .--221 closing connection->.    46     66     86
21 TCP   .<------- FIN ACK -------- .     0     20     40
22 TCP   . -------- ACK ----------->.     0     20     40
23 TCP   . ------- FIN ACK -------->.     0     20     40
24 TCP   .<-------  ACK ----------- .     0     20     40
```

Measurement Results

Total IP Packet bytes: 1778

Message Length (header + body): 301

Total Overhead (TCP header + IP header): 1477

Message as Received

```
Received: by bluejeans. (SMI-8.6/SMI-SVR4)
<09>id PAA24890; Fri, 13 Sep 1996 15:34:53 -0700
Date: Fri, 13 Sep 1996 15:34:53 -0700
From: jcheng@bluejeans
Message-Id: <199609132234.PAA24890@bluejeans.>
To: dnakano@griffey.nwest.airdata.com
Subject: test1

1234567890 1234567890 1234567890
1234567890 1234567890
1234567890
```

7.4.2 Message Delivery via POP Mailbox

Please refer to Figure 7.2 below, which shows the setup for the following two delivery experiments in detail. The experimental setup at Neda Communications involves the following:

- A POP Server: Sun Sparc running Solaris 2.4.
- An IMAP Server: Sun Sparc running Solaris 2.4.
- A ”receiver”: IBM Thinkpad Laptop running Microsoft TCP/IP on Windows 95
- A sniffer that monitors packet movement at the juncture of the above three devices, recording two-way traffic

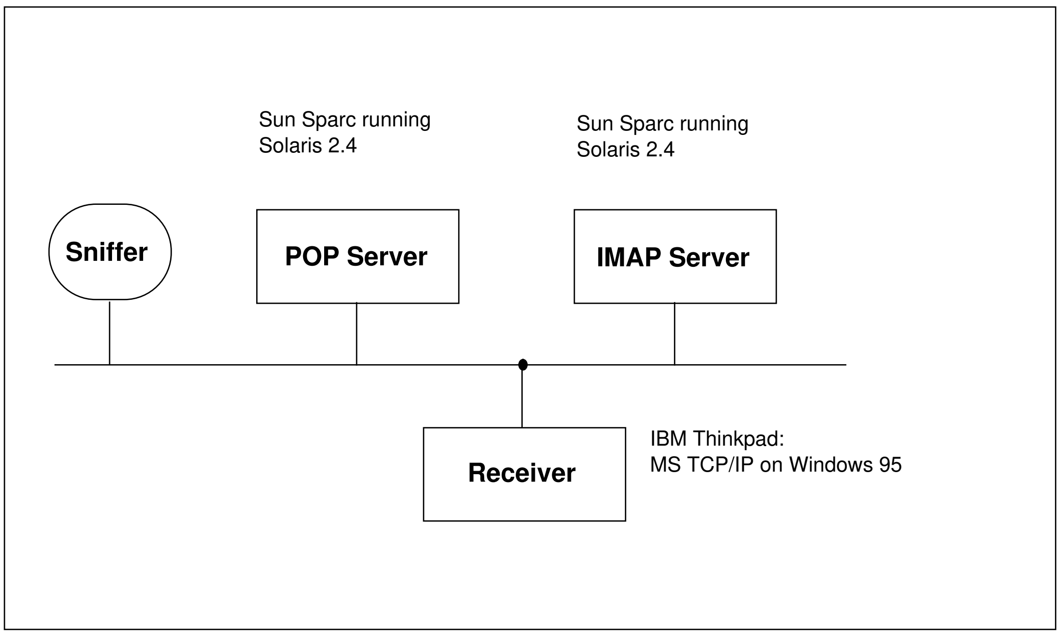


Figure 7.2: Experimental Setup for Delivery

Message Delivery Process

The message was delivered to the laptop from the POP server. After invoking Microsoft's Internet Explorer 3.0 on the laptop and bringing up MS Internet Mail, the new message was automatically retrieved from the POP server. The sniffer recorded traffic data between the POP server and the recipient.

Protocol Trace

```
        (arash)                (vader)
IP_PDU  Mailbox                 Client DATA   TCP      IP
----------------------------------------------------------------
1  DNS   *<-- DNS Query ----------- .                   (dns)
2  DNS   * -- DNS Reponse---------->.
```

```
3  TCP   .<-- SYN req-------------- .     0    24     44 (conn)
4  TCP   . ---SYN ACK ------------->.     0    24     44
5  TCP   .<-- ACK ----------------- .     0    20     40

6  TCP   . ---POP3 server OK ------>.   117   137    157 (auth)
7  TCP   .<-- ACK ----------------- .     0    20     40
8  TCP   .<-- AUTH twinkie -------- .    14    34     54
9  TCP   . ---unknown command ----->.    45    65     85
10 TCP   .<-- USER test-1 --------- .    13    33     53
11 TCP   . ---user acpt,password? ->.    41    61     81
12 TCP   .<-- PASS ****** --------- .    13    33     53
13 TCP   . ---+OK ----------------->.     0    20     40

14 TCP   . ---+OK mbox open 1 msg-->.    30    50     70 (trans)
15 TCP   .<-- STAT ---------------- .     6    26     46
16 TCP   . ---+OK 1 542------------>.    11    31     51
17 TCP   .<-- UIDL 1 -------------- .     8    28     48
18 TCP   . ---unknown command ----->.    43    63     83
19 TCP   .<-- TOP 1 0 ------------- .     9    29     49
20 TCP   . ---+OK Top of msg ------>.   503   523    543 (_header)
21 TCP   .<-- LIST ---------------- .     6    26     46
22 TCP   . ---+OK scan listing----->.    44    64     84
23 TCP   .<-- RETR 1 -------------- .     8    28     48
24 TCP   . ---+OK 542  msg body---->.   561   581    601 (_body)
25 TCP   .<-- DELE 1 -------------- .     8    28     48
26 TCP   . ---+OK msg deleted ----->.    21    41     61
27 TCP   .<-- ACK ----------------- .     0    20     40

28 TCP   .<-- QUIT ---------------- .     6    26     46
29 TCP   . ---+OK ----------------->.     6    26     46
30 TCP   . ---Sayonara ------------>.    14    34     54
31 TCP   .<-- FIN ACK ------------- .     0    20     40
32 TCP   . ---+OK sa--------------->.     6    26     46
33 TCP   .<-- FIN ACK ------------- .     0    20     40
34 TCP   . ---ACK ----------------->.     0    20     40
---------------------------------------------------------------
```

Measurement Results

Total IP Packet bytes: 2731

Message Length (header+body): 561

Total Overhead: 2170

Message as Received

```
+OK 542 octets..
Return-Path: <muratd@neda.com>..
Received: from vader.neda.com by arash.neda.com (5.0/SMI-SVR4)...
```

```
  id AA04601; Wed, 18 Sep 1996 16:35:39 +0800..
Date: Wed, 18 Sep 1996 16:35:11 -0700 ()..
From: Murat Divringi <muratd@neda.com>..
To: test-1@arash.neda.com..
Subject: test6..
Message-Id: <Pine.WNT.3.95.960918163418.-158025A-100000@vader.neda.com>..
X-X-Sender: muratd@zahak.neda.com..
Mime-Version: 1.0..
Content-Type: TEXT/PLAIN; charset=US-ASCII..
Content-Length: 66..
Status:   ..
..
012345678901234567890123456789 ..
01234567890123456789 ..
0123456789 ..
..
```

7.4.3 Message Delivery via IMAP Mailbox

Please refer to Figure 7.2 above for the experimental setup.

Message Delivery Process

Message was delivered to the laptop from the IMAP server. After invoking PC-Pine, the new message was automatically retrieved from the IMAP server. The sniffer recorded traffic data between the IMAP server and the recipient.

Protocol Trace

```
        (zahak)         (198.62.92.35)
IP_PDU  Mailbox                  Client DATA   TCP     IP
---------------------------------------------------------------
1  DNS   *<-- DNS Query ----------- .                   (dns)
2  DNS   * -- DNS Reponse---------->.

3  TCP   .<-- SYN req-------------- .     0    24     44 (conn)
4  TCP   . ---SYN ACK ------------->.     0    24     44
5  TCP   .<-- ACK ----------------- .     0    20     40

6  TCP   . ---IMAP2 server OK ----->.    78    98    118 (auth)
7  TCP   .<-- ACK ----------------- .     0    20     40
8  TCP   .<-- LOGIN test-1 **** --- .    28    48     68
9  TCP   . ---ACK ----------------->.     0    20     40
10 TCP   . ---LOGIN completed ----->.    27    47     67

11 TCP   .<-- A001 SELECT INBOX --- .    21    41     61
12 TCP   . ---ACK ----------------->.     0    20     40
13 TCP   . ---A001 cmp 1 EXISTS --->.   110   130    150
14 TCP   .<-- A002 NOOP ----------- .    13    33     53
15 TCP   . -- A002 NOOP cmp ------->.    26    46     66
```

```
16 TCP   .<-- A003 FETCH 1:1 ALL -- .    22    42     62
17 TCP   . -- A003 FETCH evlp cmp-->.   364   384    404 (())
18 TCP   .<-- A004 NOOP ----------- .    13    33     53
19 TCP   . -- A004 NOOP cmp ------->.    26    46     66
20 TCP   .<-- ACK ----------------- .     0    20     40
21 TCP   .<-- A005 FETCH 1:1 FULL-- .    23    43     63
22 TCP   . -- A005 FETCH 1:1 cmp--->.   431   451    471 (())
23 TCP   .<-- A006 FETCH 1 RFC822hdr.    30    50     70
24 TCP   . -- A006 FETCH 1 cmp hdr->.   708   728    748 (_header)
25 TCP   .<-- A007 FETCH 1 body-----.    24    44     64
26 TCP   . -- A007 FETCH 1 cmp body>.   125   145    165 (_body)
27 TCP   .<-- ACK ----------------- .     0    20     40
28 TCP   .<-- A008 SEARCH DELETED --.    23    43     63
29 TCP   . -- A008 SEARCH cmp ----->.    38    58     78

30 TCP   .<-- A009 LOGOUT --------- .    15    35     55
31 TCP   . ---ACK ----------------->.     0    20     40
32 TCP   . -- A009 LOGOUT cmp ----->.    80   100    120
33 TCP   .<-- FIN ACK ------------- .     0    20     40
34 TCP   . ---ACK ----------------->.     0    20     40
35 TCP   . -- FIN ACK ------------->.     0    20     40
36 TCP   .<---ACK ----------------- .     0    20     40
```

Measurement Results

Total IP Packet bytes: 3593

Message Length (header+body): 833

Total Overhead: 2760

Message as Received

```
* 1 FETCH(RFC822.HEADER {646}..
Received: from arash.neda.com (arash [198.62.92.10]) by zahak.neda.com
  (8.6.10/8.6.10) with SMTP id QAA16710 for <test-1@zahak>;
  Wed, 18 Sep 1996 16:42:24-0700..
Received: from vader.neda.com by arash.neda.com (5.0/SMI-SVR4)...
  id AA04617; Wed, 18 Sep1996 16:41:42 +0800..
Message-Id: <9609182341.AA04617@arash.neda.com>..
From: "test-1" <test-1@neda.com>..
To: <test-1@zahak.neda.com>..
Subject: test6..
Date: Wed,18 Sep 1996 16:41:13 -0700..
X-Msmail-Priority: Normal..
X-Priority: 3..
X-Mailer: Microsoft Internet Mail 4.70.1155..
Mime-Version: 1.0..
Content-Transfer-Encoding: 7bit..
Content-Type: text/plain; charset=ISO-8859-1..
Content-Length: 66....)..
```

```
A00006 OK FETCH completed..
* 1 FETCH (BODY[1] {70}..
012345678901234567890123456789 ..
01234567890123456789 ..
0123456789..
..
)..
A00007 OK FETCH completed..
```

7.4.4 EMSD Delivery from Unix to PC

Please refer to Figure 7.2 above for this experiment.

Message Delivery Process

The message was delivered to the laptop, running Neda's EMSD Mail User Agent version 0.9 on Windows 3.1, from Neda's EMSD Message Test Center. The sniffer recorded exchange of data between the recipient and Neda's EMSD Message Test Center which was ¡emsd.neda.com¿ implementing ESROS.

Protocol Trace

```
IP_PDU   UA                MailServer   DATA   UDP     IP
---------------------------------------------------------------
1  UDP   .<--Invoke header+body --- .   299    307    327
2  UDP   . -----Response ---------->.     2     10     30
3  UDP   .<------- Ack ------------ .     2     10     30
---------------------------------------------------------------
```

Measurement Results

Total IP Packet bytes: 387

Message Length (header+body): 299

Total Overhead: 88

Comparing EMSD delivery with SMTP delivery we see that total IP packets in the case of EMSD delivery is down by a factor of 4.6, and total overhead is down by a factor of 16.8.

In the case of POP retrieval, total IP bytes in the case of EMSD delivery is down by a factor of 7, and total overhead is down by a factor of 24.7.

Finally for IMAP delivery, total IP packets in the case of EMSD delivery is down by a factor of 9.3, and total overhead is down by a factor of 31.4.

Message as Received and Decoded

```
From jcheng@airdata.com Tue Oct 01 16:16:31 1996
Date: 14 Sep 96 05:48:55 GMT
From: jcheng@airdata.com
Subject: TEST Subject
```

	EMSD	SMTP
Total number of IP packets	3	24
Total IP bytes	307	1886
Total MSG length (mail hdr+ mail body)	206	437
Total overhead	101	1449

Table 7.2: Comparison of Submission Traffic overhead for EMSD and SMTP

	EMSD	SMTP	IMAP	POP
Total number of IP packets	3	24	36	34
Total IP bytes	387	1778	3593	2731
Total MSG length (mail hdr+ mail body)	299	301	833	561
Total overhead	88	1477	2760	2170

Table 7.3: Comparison of Delivery Traffic Overhead for EMSD, SMTP, IMAP and POP

```
To: 333.333@
Message-ID: <199609132148.OAA24774@bluejeans.>
Content-Length: 66
X-Homepage: Visit our home page at http://www.airdata.com/
     id OAA24774; Fri, 13 Sep 1996 14:48:55 -0700

1234567890 1234567890 1234567890
1234567890 1234567890
1234567890
```

7.5 Results Summary

The following paragraphs summarize the results obtained above. Results indicate that EMSD compares very favorably to other message transfer mechanisms.

7.6 Conclusion

Results of the experiments show the dramatic efficiency gain of EMSD over all the other protocols under test.

However, it should be noted that EMSD was specifically designed for efficient short messaging in the context of mobile wireless devices, and thus from inception was meant to be more efficient than protocols designed to handle a wider variety of messages. Deployment and use of EMSD similarly is geared towards messaging scenarios that are a subset of the current global messaging world, such as palmtop devices exchanging messages over an airlink. At the other extreme, in a traditional office environment, concerns like efficient use of communications infrastructure and maximizing the battery life of the devices do not necessarily apply to tethered devices plugged to a standard wall outlet and a high speed (non-air) networking infrastructure.

Within its own domain, EMSD does its job efficiently and admirably and as is clear from the results of this study, is a highly competitive alternative to other messaging protocols.

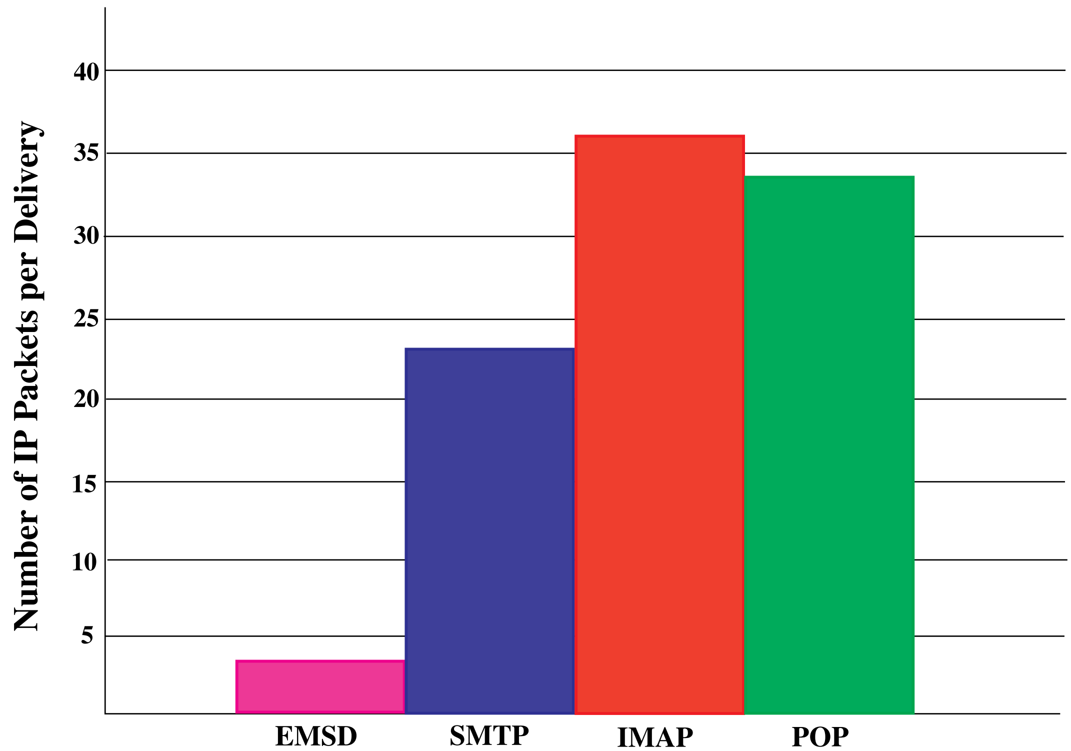


Figure 7.3: Packets Per Delivery

7.7 Acknowledgments

This study was performed with the support and assistance of AT&T Wireless Services.

Chapter 8

A Brief History of LEAP

8.1 Overview

The origins of the LEAP protocols go back to 1994, when they originated as part of the research and development initiatives of McCaw Cellular's wireless data group (now AT&T Wireless Services). The development work that would eventually lead to LEAP was initially undertaken in the context of the CDPD network; its scope was later expanded to include the Narrowband PCS network also.

By 1996 McCaw Cellular was fully committed to paging, had recently purchased two nationwide narrowband wireless PCS licenses, and wished to develop an efficient wireless messaging system. Neda Communications, Inc., an independent consulting company working under contract to McCaw Cellular, played a key role in the development of the required system. Neda Communications had also been involved from the outset in the development of the CDPD specification.

In 1997 however, soon after the purchase of McCaw Cellular by AT&T Wireless, the latter company abandoned the wireless messaging project entirely. Prior to this event, Neda Communications had secured from AT&T the necessary rights to continue independent development of the protocols. Therefore, recognizing the eventual future need for these protocols, Neda then undertook to continue development of them independently of AT&T. They were eventually completed by Neda, published as RFCs, and now form the basis of the LEAP protocols.

8.2 Time-Line History

A time-line history of the significant events relating to LEAP is provided below. Note that the name LEAP is relatively new; this acronym was coined in early 2000. Prior to 1997, the research and development work which would eventually lead to the creation of LEAP was referred to by the general name of Limited Size Messaging (LSM).

Much of the LSM work was sponsored in various ways by McCaw Cellular, then later by AT&T Wireless Services (AWS). Two divisions of AWS are referred to in the time-line below. First, the Wireless Data Division (WDD) of AWS led much of the LSM-related development work. WDD was the division of AWS which had major responsibility for the CDPD (Cellular Digital Packet Data) network.

Later, the Messaging Division of AWS also made use of the LSM technology in the context of their Narrowband PCS (NPCS) network. In the context of Narrowband PCS, LSM was referred to by the general name of pACT (personal Air Communications Technology).

Summer 1994: The basic concept of providing wireless e-mail services over the CDPD network was first analyzed.

January 1995: AWS began creating the LSM protocol specifications. This work was carried out as a joint effort between the Wireless Data Division, and the NPCS Group within the Messaging Division.

January 1995: AWS began development of the reference implementation of the LSM protocols for both Message Centers and Devices.

June 1995: WDD submitted the LSM specifications to the CDPD Forum. The WDD made various LSM-related direction statements, and produced several press releases. This resulted in significant press coverage of LSM. Early development of the WAP protocols had the benefit of seeing this public release of LSM technology, and was based in part upon it.

December 1995: Neda's reference implementation of LSM was completed and ready for demonstration and testing.

December 1995: AWS sent out Requests For Proposal to potential large-scale Message Center suppliers.

February 1996: Neda's LSM device implementation interoperated with Aldiscon's Message Center.

March 1996: Sema Group UK was selected as the production Message Center supplier by AWS.

April 1996: The pACT Vendor Forum was formed. The initial forum members included Ericsson, PCSI, Aldiscon, AT&T, Casio, NEC, Novatel, Research in Motion, and Sema Group UK.

July 1996: Neda completed interoperability tests against the PCSI pACT pager.

August 1996: AWS issued the equivalent of a VAR agreement to Neda for development and distribution of the LSM software.

September 1996: Neda supplied LSM technology (in the form of source code) to Sema Group UK, and assisted Sema in the development of Message Center products for AWS.

November 1996: AWS changed the LSM strategy for pACT from two-way to "mostly one-way plus."

December 1996: Neda's palmtop LSM became ready for general testing.

January 1997: Sema Group UK delivered its first release of the LSM Message Center product.

January 1997: The Messaging Division of AWS licensed Neda's LSM product set.

February 1997: Neda's LSM implementation interoperated with Sema Group UK's LSM implementation.

February 1997: WDD terminated funding of LSM-related work, and focussed instead on early development of WAP.

March 1997: On March 17, AWS terminated the two-way paging project entirely. The NPCS Group of the Messaging Division was abruptly shut down, all employees were reassigned, and all vendor work terminated. Later the same year, the two nationwide Narrowband PCS licenses belonging to AWS were sold.

April 1997: Neda began development of EMSD and the Enhanced Two-Way Paging (ETWP) products.

September 1997: Efficient Short Remote Operations (ESRO) protocol was published as Internet RFC 2188.

June 1998: The Windows CE efficient e-mail implementation was publicly released by Neda.

August 1998: ETWP Subscriber Services and web access were made available.

November 1998: Open maintenance organization EMSD.org was established to support public enhancement of the EMSD protocol.

January 1999: Open maintenance organization ESRO.org was established to support public enhancement of the ESRO protocol.

February 1999: Efficient Mail Submission & Delivery (EMSD) protocol was published as Internet RFC 2524 by Neda.

March 2000: Neda made patent-free declarations to the Free Protocols Foundation with respect to RFC 2188 and RFC 2524.

This brings us up to the present. Our plans for the future of LEAP are described in a separate article in this Manifesto, entitled *The Future of LEAP*.

8.3 Acronym Apology

We live in the age of the acronym. Our language is now littered with more acronyms than at any other time in history, with more being added every day. We are inundated, swamped, awash with acronyms. We even have an acronym (TLA) for referring to acronyms.

The right acronym can make all the difference between the success and the failure of a product or idea, and for this reason considerable effort sometimes goes into creating just the right acronym.

In some cases the result is an acronym which is in good harmony with what it represents: PAWS (Progressive Animal Welfare Society); MADD (Mothers Against Drunk Driving). In other cases the acronym has good force and immediacy, but is not especially relevant to what it represents: NOW (National Organization of Women). On the other hand, the striving for a catchy acronym can lead to a labored and contrived construction: DARE (Drug Abuse Resistance Education).

In some cases no thought at all is given to the aesthetics of the acronym, resulting in one which is both pointless and clumsy: AFTRA (American Federation of Television and Radio Artists); or even worse, one which carries an actively negative connotation: SAG (Screen Actors' Guild).

As can be imagined, the search for the right acronym for our protocols has given rise to a protracted and sometimes emotional debate. Prior to 1997, while the protocols were undergoing development at AT&T Wireless Services, they were referred to as LSM, standing for Limited Size Messaging. When Neda began independent development of the protocols in 1997, the early, working name for the protocols was EAPS, standing for the Efficient Application Protocol Suite. This of course, is a purely engineering construction, describing the basic nature and purpose of the protocols perfectly. However, the word EAPS makes those of us with more aesthetic sensibilities physically ill.

The working name EAPS was eventually displaced by the name WHIP, standing for the Wireless High-Performance Internet Protocols. WHIP has a good strong personality, and is therefore more likely to remain in the mind of the hearer. An important component of our Manifesto strategy is capturing mindshare. In today's deafeningly noisy Internet environment, any assistance is welcome.

The price to be paid for this, of course, is that WHIP is contrived and inaccurate. First, the use of the word "Wireless" is inappropriate. There is nothing about the protocols which restricts them to wireless applications. Certainly, they have been designed with the needs of wireless applications in mind, but their major defining characteristic is their efficiency, which makes them appropriate for use in many applications, wireless and otherwise. Also, the hyphenated phrase "High-Performance" has been deliberately chosen to provide the coveted "H." A more accurate word would be Efficient, but there is nothing remotely memorable about "WEIP."

For these reasons, the name WHIP caused considerable distress among the engineering segment of the development team. Nevertheless, despite its contrived nature, WHIP persisted as a very strong candidate. This is because WHIP provides a compelling and very hard-to-resist payoff: it allows us to say, with spectacular alliterative effect:

WHIP will whip WAP!

However, first and foremost we are engineers, not marketeers, and in the end the inherent inaccuracy of WHIP proved to be intolerable, and regretfully, we abandoned it.

Our final choice is LEAP, standing for the Lightweight and Efficient Application Protocols.[1] This leaves both the engineers and the linguists equally dissatisfied. It has some personality, though not as much as WHIP. And it is reasonably accurate, though not as accurate as EAPS. But it is a choice we can all live with.

WHIP is dead. *Long live LEAP!*

[1] We are grateful to Warren Sly for proposing this name

Chapter 9

The Future of LEAP

9.1 Where We Are Today

At the time of writing in June 2000, the basic structure of the LEAP protocols is complete and in place. The key component protocols have been published as Internet RFCs, and public support organizations for the continued development and maintenance of the protocols have been created. All aspects of the LEAP development and maintenance processes conform fully to the basic trilogy of principles that we espouse: patent-freedom, RFC publication, and openness of maintenance.

Our next major challenge will be to promote the usage of LEAP throughout the Mobile Messaging industry. We will facilitate and encourage the adoption of LEAP by the following mechanisms:

- We will provide free, open-source software implementations of the LEAP protocols for major device platforms such as PalmPilot and Windows CE. This free software will be distributed through http://www.MailMeAnywhere.org.
- We will provide free, open-source software implementations of LEAP which are fully integerated with major Message Center platforms such as Sendmail and Qmail. This free software will also initially be distributed through http://www.MailMeAnywhere.org.
- We will provide free Subscriber Services. These free services will initially be provided through http://www.ByName.net.

The underlying purpose of this is to eliminate all economic and legal hindrances which might otherwise inhibit the adoption and usage of LEAP. We accomplish this by means of the patent-freedom of the protocols themselves, the availability of free, open-source software implementations the protocols, and the availability of free support services. The result of this is that the costs of implementing LEAP, other than the associated overhead costs, are zero.

By means of this strategy, we intend to make LEAP widespread throughout every segment of the Mobile Messaging industry. Our eventual goal is for LEAP to become the natural choice for Mobile Messaging applications.

9.2 Invitations to Participate

This is an ambitious goal, and cannot be accomplished without the cooperation and participation of others within the industry. We invite others to participate in the following arenas:

Invitations to Protocol Developers

- Anyone who is interested in ESRO and EMSD is invited to participate in their development through ESRO.org and EMSD.org.
- Additional protocols are needed to enable efficient web browsing. A starting point for this, called EHDP (Efficient Hypertext Delivery Protocol), is currently being created. In developing EHDP we intend to build on the work of WAP and the W3 Consortium, and re-use the ESRO technology. We invite others to join us in this development effort.
- Additional protocols are needed to enable efficient access to dictionaries and other look-up data structures. A starting point for this, called EDICT (Efficient Dictionary protocol), is currently being created. In developing EDICT, we intend to build on the existing work already done in the context of the DICT protocol. We invite others to join us in this development effort.
- We anticipate that additional protocols will be needed for a variety of future applications, not all of which can be foreseen at this time. These applications will include such things as efficient implementations of ESRO-based instant messaging, chat, white pages, and others.

Invitations to Software Developers

- Based on the existing open-soure software implementations for end-user devices, we invite others to enhance the PalmPilot and Windows CE implementations, and to port LEAP to additional general-purpose device platforms.
- We invite telephone manufacturers and wireless data modem manufacturers to include the LEAP protocols as an integral part of their next generation devices.
- We invite Message Center software developers to enhance and better integrate the Message Center LEAP software.

Invitations to Subscriber Services Providers

We invite Subscriber Services providers such as AOL, Yahoo, MCI and AT&T to participate in the general concept of providing free services based on free protocols and open-source software, and to integrate LEAP into their suite of Subscriber Services. A model of this concept is provided by our own free service at www.ByName.net.

Invitations to Systems Integrators

Each of the LEAP protocols is a component which must be integrated into larger solutions. In particular, we invite the developers of customer-premise Message Centers to incorporate LEAP into their existing products.

9.3 Preview of Coming Attractions

Several LEAP-based products and services are currently under development. These include MailMeAnywhere, ByName and ByNumber.

9.3.1 MailMeAnywhere.org

In order to make use of the LEAP protocols convenient and widespread, we are providing implementations of the protocols as free and open-source software. Binary formats of the software for a variety of platforms are available. In order to provide complete solutions, the LEAP protocol components are integrated with various other free software components, forming consistent and coherent bundles. Since the initial LEAP components are oriented towards interpersonal messaging, the initial software distribution takes place through http://www.MailMeAnywhere.org.

MailMeAnywhere is a distribution center for free and open-source software which relates to LEAP, or which facilitates use of the LEAP protocols. Device implementations are available for a large number of general-purpose device platforms. Message Center implementations have been integrated with Qmail and Sendmail.

To learn more about MailMeAnywhere, see the website at http://www.MailMeAnywhere.org.

9.3.2 ByName.net and ByNumber.net

In order to make use of LEAP protocols convenient and widespread, we are also providing an initial free subscriber service which integrates the LEAP protocols into a variety of other services. We are delivering these services through the ByName.net domain. ByName provides a set of free services, based on free protocols which have been implemented as free software. The ByName services are highly personalized and are based on the user's identity – ByName is based on the user's name, while ByNumber is based on a numerical user ID.

A conventional e-mail account typically provides the user with a single address, usually of the form "someName@someDomain.com." This provides the user with a single mailbox, to which all mail for that address is sent.

This becomes inconvenient when the owner uses the account for multiple types of incoming e-mail. For example, the user may use the account for both personal and work-related mail, to subscribe to various mailing lists, and to participate in usenet groups. Over time the user may get onto a large number of mailing lists, resulting in an incoming e-mail stream spanning a very wide dynamic range of importance, from urgent personal e-mail, all the way down to meaningless spam.

E-mail applications typically deal with this by providing the user with tools to manage and prioritize mail. These consist of inbox sorters and filters to eliminate spam and prioritize incoming messages based on the originator or subject.

The ByName.net service provides a better way. ByName provides you with multiple mailboxes and addresses, each of which can be dedicated to a particular type of e-mail. Furthermore, these various addresses have a simple and uniform naming scheme, based on the one symbol that is most dear and personal to you: your own name. ByName includes your name in the domain part of the address, and appends various selectors in front of the @ sign. For example, a particular subscriber might have the following addresses and mailboxes:

```
 personal@homer.simpson.1.ByName.net
   office@homer.simpson.1.ByName.net
   urgent@homer.simpson.1.ByName.net
   public@homer.simpson.1.ByName.net
   mobile@homer.simpson.1.ByName.net
    pager@homer.simpson.1.ByName.net
      fax@homer.simpson.1.ByName.net
emergency@homer.simpson.1.ByName.net
```

This provides our anti-hero with a consistent set of e-mail boxes that he can use for different purposes – one address for personal mail, a different one for work-related mail, and so on. Homer now has control over the routing of his e-mail without having to use a mail sorter or filters.

Your home page is also based on your name; Homer's is **http://homer.simpson.1.ByName.net**.

To learn more about the ByName service and to apply for an account, see the website at http://www.ByName.net.

The ByNumber.net service provides a complementary service to ByName, based on numbers rather than letters. ByNumber enables devices with digit-only origination capability (e.g. conventional telephone keypads) to send e-mail messages, and provides a unified way of sending messages to pagers, two-way pagers, faxes and e-mail accounts.

To learn more about the ByNumber service, see the website at http://www.ByNumber.net.

Part II

LEAPing Over Closed Solutions

Chapter 10

The WAP Trap

10.1 Introduction

The new Internet reality is that of wireless networks, providing service to legions of miniaturized, hand-held mobile devices. This places an entirely new set of requirements on the underlying communications protocols – they must provide the power efficiency demanded by hand-held wireless devices, together with the bandwidth efficiency demanded by wide area wireless networks.

At some point, the wireless data communications industry must agree on a common set of standard protocols that satisfies these requirements. Unfortunately, the road to an industry standard is a rocky one. The wireless industry is populated by a number of disparate constituencies and self-interests. Among these constituencies are the technical community, with its fundamental mandate to create sound engineering solutions, and the business community, ultimately driven by the pursuit of profit and marketplace advantage. The differing agendas of these constituencies frequently bring them into conflict.

In this confusing environment it can be very difficult to distinguish between developments that are genuine, enabling technologies, and those that are ill-conceived wild-goose chases.

10.1.1 The Wireless Application Protocol (WAP)

Into this chaotic arena comes WAP. On April 30 1998, a group of business interests published a set of specifications referred to as the **Wireless Application Protocol**, or **WAP**. WAP is a specification for wireless data communications using hand-held devices such as mobile phones and palmtop computers. Use of the WAP specification allows mobile devices to communicate with the Internet or an intranet, providing the users of these devices with mobile data communications capabilities such as web-browsing and e-mail.

The WAP specification was developed by the **WAP Forum**, an industry association of wireless device manufacturers, service providers, and software companies. The WAP Forum was founded in June 1997 by three mobile phone manufacturers (Ericsson, Motorola and Nokia), together with the US software company Phone.com (formerly Unwired Planet). The WAP specification is largely the product of these four founding companies. Further information about the WAP Forum can be found at its website at http://www.wapforum.org/.

The Wireless Applications Protocol purports to be just what the doctor ordered: a set of standards that will unify the wireless data applications industry. WAP presents itself as an open, license-free standard for wireless Internet access. It claims to be a well-designed engineering construction, allowing free interoperability among wireless industry vendors. It claims to be an enabling technology that will catalyze the development of the wireless industry, to the ultimate benefit of the industry and the consumer.

As we will argue in this article, however, WAP satisfies none of these claims.

10.1.2 Characteristics of Successful Protocols

Industry standards do not usually come about as the result of an orderly design process. Especially in the early stages of industry development, protocols and standards arise organically, and without the benefit of hindsight. Because of this, early protocols are frequently very much less than ideal. As Bill Joy, the founder of Sun Microsystems, puts it,

> ”Sometimes when you fill a vacuum, it still sucks.”

Though his phrasing leaves much to be desired, his point is beyond debate: the most appalling solution is better than no solution at all.

However, history has shown that successful protocols tend have certain characteristics in common. By a ”successful” protocol, we mean one which becomes accepted as an industry standard in the face of competing protocols, endures as an standard in the long term, and serves to promote the growth of the industry.

The key characteristics of a successful protocol are:

1. **Adequate Technical Design.** It should address the basic technical requirements of the industry. This means that the protocol must primarily be an engineering construct, not a business one.
2. **Open Development and Maintenance Process.** Some form of mechanism should exist for public commentary on the protocol. The protocol should be maintained by a process that allows the participation of the various constituencies that are affected by the protocol.
3. **Open Availability Process.** It should be published and made available in a way that ensures that it is freely, easily and permanently accessible to anyone who wishes to use it.
4. **Freedom from Usage Restrictions.** There should be no restrictions on the use of the protocol. Anyone who wishes to base an application on the protocol should be able to do so without legal or financial hindrance of any kind.

Not all successful protocols have all these attributes. Nevertheless, as the history of protocol development demonstrates, the more a protocol conforms to these attributes, the more likely it is to become an enduring industry standard. An analysis of several successful and failed protocols, supporting this conclusion, is provided in the article entitled *Lessons from History: Comparitive Case Studies* within *The LEAP Manifesto* [**?**].

WAP claims to have all four of the above attributes. In fact, it has none of them. WAP has claimed center stage not because it fulfills the needs of the industry, but because thus far, no viable alternative has been presented.

10.1.3 About this Document

In this article we will show that WAP is entirely unfit for the purpose being claimed for it. We will show that it is handicapped as a result of the processes the WAP Forum has used to develop it, and that it includes numerous serious technical design errors. Our conclusion will be that the WAP specification is essentially a marketing construct, rather than an engineering one. It is designed to provide short-term financial benefit to a minority of the WAP Forum members, rather than providing long-term benefit to the industry at large and the consumer.

We will then enumerate and analyze the steps that can be taken to prevent the harm of WAP. One of the most crucial steps will be to identify alternatives to WAP, and eventually adopt one of these in place of WAP.

Finally, we will propose one alternative to WAP, namely LEAP, the Lightweight & Efficient Application Protocols. We provide a brief description of LEAP, and provide references to where further information on LEAP can be found.

This article is one of a series of articles we have written that analyze the current status of the wireless data communications industry, criticize WAP, and present our view of what is truly needed to promote the growth of the industry. Other articles in this series are:

- *LEAP: One Alternative to WAP* [**?**]. A companion paper to this one, taking over where this leaves off. The scope of the current paper is largely limited to a critique of WAP. This companion paper describes a specific alternative to WAP.
- *The LEAP Manifesto* [**?**]. This document provides a complete analysis of the industry, and a detailed description of the LEAP protocols.

Alternative Formats

In addition to this English-language version, this document has also been translated into French under the title *Le WAP a la Trappe*. Both English and French versions are available in several alternative formats at the Free Protocols Foundation website
(http://www.FreeProtocols.org/wapTrap):

- ONE-HTML: Displays the entire document as a single web page.
- SPLIT-HTML: Displays the document as a set of linked web pages, starting from the Table of Contents.
- PDF: Provides the document in Adobe Acrobat format. Note that the Adobe Acrobat Reader is required to read this format. The Acrobat Reader can be downloaded from the Adobe website.
- PS: Provides the document in PostScript format for printing.
- Text Only: Provides the document in text-only format.

Acknowledgments

We gratefully acknowledge the assistance of the following persons in the preparation and review of this document: Andrew Hammoude, Richard Stallman, Bill Frezza, Rob Mechaley, and Pinneke Tjandana.

Conflict of Interest Disclosure

The authors of this article were also the initial developers of LEAP, and therefore have a vested interest in the success of LEAP over WAP.

However, we are also active participants in the Free Protocols Foundation (FPF), under whose auspices this article is being written. As participants in the FPF, we are fully committed to its patent-free principles; principles which WAP violates completely. The mission of the Free Protocols Foundation is to provide support for patent-free protocols. Part of this mission is to provide support for patent-free alternatives to patented protocols such as WAP. It is in the spirit of this mission that this article is being written.

The purpose of this article is not to promote LEAP or any other particular alternative to WAP. The purpose of this article is to expose the harm of WAP and describe the steps that can be taken to prevent it. Any other viable alternatives to WAP that are brought to our attention, and that conform to the principles of the Free Protocols Foundation, will be promptly referenced at the FPF website, and included in subsequent versions of this article.

The most recent version of this article, describing all known alternatives to WAP, will be maintained on the FPF website at http://www.FreeProtocols.org.

10.2 WAP - A Procedural Fraud

There are two distinct sets of problems relating to WAP: procedural, and technical. In this section we will describe the procedural problems. The procedural problems lie in the processes that the WAP Forum has used to develop and disseminate the WAP specifications.

10.2.1 Not Open in Terms of Development and Maintenance

A highly desirable attribute of an industry standard protocol is that it be the result of an open design process. What this means is that, somewhere along the line, the various constituencies that have a stake in the protocol be given a voice in its development.

This does not mean that the protocol must be conceived and built from the ground up by industry-wide consensus. Indeed, this is usually impractical, and of necessity, the first drafts of any protocol are usually created by a small group, functioning autonomously.

An open design process means only that *at a certain point*, ownership of the protocol must become public. Beyond that point, the various industry constituencies must be given an opportunity to participate in its design, and the design process must include mechanisms for reaching consensus among conflicting lobbies.

Openness of design has two important benefits. First, it allows the protocol to be subjected to adequate technical review, ensuring that it is a sound engineering solution. Second, it prevents the design of the protocol from being excessively influenced by minority business self-interests. An open design process provides vital assurance of the integrity of the resulting protocol, in both engineering and business terms.

The WAP specification is in complete violation of these principles. The specification has been developed exclusively by the WAP Forum, entirely behind closed doors, and without the benefit of a single public mailing list for discussion and review. The WAP Forum permits no outside influence over the specification; the only institutions that may participate in its development and maintenance are the WAP Forum members themselves.

The WAP Forum claims that the specification *is* open, on the grounds that any company or organization is free to join the forum. While technically this is true, membership in the Forum carries a $27,000 entrance fee (as of February 2000). In practice, this fee effectively excludes most small businesses, and virtually all academic institutions.

The WAP Forum is therefore a highly restricted subset of the business and academic community, and influence over the specification is limited to those companies that can afford this entrance fee. The specification is thus the creation of a limited constituency of the telecommunications world; specifically, it is primarily the creation of a set of telephone manufacturers. Though important, the telephone manufacturers represent just a single component of the world of data communications. In creating WAP, grossly inadequate consideration has been given to other important disciplines and constituencies, such as the Internet engineering community, the academic community, and the small business community.

10.2.2 No Assurance of Availability and Stability

An essential attribute of an industry standard protocol is that it be freely and permanently accessible to anyone who wishes to use it. In the Internet world, this is traditionally accomplished by means of RFC publication. RFC publication provides the following important benefits:

- **World-Wide Distribution.** RFC publication ensures unrestricted world-wide distribution of the published protocol. There are many Web and FTP sites world-wide that carry the complete series of RFC documents - if any one site is busy or unavailable, someone seeking an RFC can simply go to another.
- **Unrestricted Distribution.** A user may download an RFC publication completely freely, without incurring any legal restrictions whatsoever. All that is required to acquire the full text of an RFC is its number or title.
- **Permanence.** RFC publication is permanent. Even if the creator of a protocol should go out of business or otherwise become defunct, the RFC will remain in existence.
- **Stability.** Once published, an RFC is *fixed* – it cannot undergo change. If a new revision of the standard must be issued, this is handled by issuing the revision under a new RFC number.

The WAP specifications do not carry these same assurances of free and permanent availability. Rather than being published as an RFC or by another third-party agency, the specifications are self-published by the WAP Forum. As a result of this, each of the above benefits of RFC publication is in some way diminished:

- **Limited Distribution.** The specifications are only available from a single source: the WAP Forum itself. If the WAP Forum web site should happen to be down, the specifications are simply unavailable.
- **Restricted Distribution.** In order to download any particular WAP specification, a user must agree to a license agreement.
- **Diminished Permanence.** The permanence of the WAP specifications is only as good as that of the WAP Forum itself. If the WAP Forum should ever become defunct and cease operations, all its specifications will become orphaned.
- **Diminished Stability.** The WAP Forum does not guarantee the stability of its specifications. As of February 2000, each WAP specification carries on its title page the disclaimer, “This document is subject to change without notice.”

This last item is particularly worrisome. A key attribute of a protocol is that any particular revision of it be *fixed* – indeed, this is the very definition of a standard. The WAP Forum’s disclaimer gives it the power to modify individual revisions of the specification at will – an extraordinary power to hold over something that purports to be an industry standard. This is not a purely theoretical concern – the WAP Forum has already exercised this power inappropriately [**?**].

Of overriding significance is the fact that the WAP Forum has declined to publish its specifications as RFCs. For all the above reasons, Internet-related protocols are always published as RFCs – this is the mainstream, industry-standard, method for publication of Internet protocols. RFC publication is well-understood and accepted within the Internet community, and fully represents the spirit of cooperation which is characteristic of this community. Quite simply, there is no good reason to do otherwise.

Yet the WAP Forum has done otherwise. Our question is: Why? We can think of only three plausible reasons:

1. The specifications are so technically deficient that they do not meet the minimum standards required for RFC publication.
2. The WAP Forum wishes to retain full control over the specifications, including the ability to change them at will, without regard to the resulting loss of stability.
3. The WAP Forum wishes to impose copyright restrictions on the protocols beyond those provided by RFC publication.

Whatever the reason, the WAP Forum evidently does not subscribe to the spirit of openness and cooperation represented by RFC publication and practiced by the Internet engineering community at large.

10.2.3 Not Patent-Free

An essential attribute of an industry building protocol is that there must be no restrictions on its usage. In particular, there must be no patent restrictions on the use of the protocol.

The WAP specification, however, is burdened with several patent restrictions. These include patents held by certain members of the WAP Forum itself, most notably Phone.com and Geoworks. Patent infringement claims have already been made by the holders of the following patents:

- U.S. Patent # 5,327,529 (Geoworks). Process of designing user's interfaces for application programs.
- U.S. Patent # 5,809,415 (Phone.com, formerly Unwired Planet). Method and architecture for an interactive two-way data communication network.

More patent infringement claims can almost certainly be expected in the future.

One of the benefits of a standard protocol is that it does not provide any advantage to one industry player over another. Anyone who wishes to develop products and/or services based on the protocol may do so, and may then compete with similar products and/or services in a fair, open, free-market environment. In such an environment products succeed or fail based on their merits, and the benefits to the consumer are those traditionally resulting from free-market competition: better products at lower prices.

The inclusion of patents in a standard entirely corrupts this process. The patent provides the patent-holder with a sustained and unfair market advantage. The losers are the industry as a whole, small companies, and the consumer.

Patents thus pose a significant danger to protocols. For this reason, the protocol development process must include mechanisms to address and eliminate patent restrictions. Such mechanisms exist, and are well-known and understood within the Internet community. For example, we refer the reader to RFC 2026, *The Internet Standards Process – Revision 3* [**?**]. Section 10 of this document, *Intellectual Property Rights*, describes the policies and procedures followed by the IESG (Internet Engineering Steering Group) in this regard. Among other things, this section describes their policy regarding:

- Strong preference for the adoption of patent-free technology wherever possible.
- The prompt, forthright disclosure of any actual or potential patent rights which may affect the protocol.
- The steps to be taken to secure from the patent owner an unlimited, perpetual, non-exclusive, royalty-free license to include the patent as part of the protocol.

The IESG policy is an example of the effort typically made within the Internet community to work strenuously and diligently towards the goal of a patent-free protocol.

As another example, the Free Protocols Foundation publishes a set of policies and procedures for protocol development that ensures, as far as possible, that the resulting protocol is functionally patent-free. These procedures are fully documented in the *Free Protocols Foundation Policies and Procedures, Version 1.0* [**?**]. Any development organization is free to adopt these procedures with regard to its own protocols.

The WAP Forum, clearly, has not followed the example set by the Internet community at large. Procedures such as those of the IESG and the Free Protocols Foundation are well understood within the Internet community, and by following such procedures the WAP Forum could have ensured patent-freedom of the WAP specification, had it wished to do so. However, it has not adopted such procedures, and as a result the WAP specification is in total violation of the principles of patent-freedom.

More than any other factor, it is the failure of the WAP Forum to work towards a patent-free specification that leads us to characterize the WAP specification as a trap. Two of the companies that participated in the development of the WAP protocol (Phone.com and Geoworks) own patents which they silently included in the protocol design. They remained silent until use of the protocol began to spread throughout the industry, and only then did they announce their patent ownership and demand royalties. In effect, these companies have booby-trapped the WAP specification with their patents.

The WAP Forum clearly does not share the Internet community's commitment to patent freedom. Indeed, the WAP Forum's attitude to patents appears to be diametrically opposed to this. To quote Ben Linder, VP of Marketing for Phone.com [**?**],

> "By virtue of building a standard, each company contributes a small part of it. Then you trade with each other to keep implementing the standard."

Mr. Linder seems to view patent rights as something to be traded among companies like baseball cards. Our question is: How are companies without any cards expected to participate in this trade?

The spirit of a healthy protocol is that it be open and free, a spirit which the WAP specification violates completely. The WAP Forum's characterization of WAP as an "open, license-free standard" is utterly preposterous.

10.2.4 No Legitimacy as a Standard

Throughout its web site and printed materials, the WAP Forum repeatedly refers to its specification as a "standard." The use of this term is inappropriate and misleading.

In common engineering usage, the term "standard" means certain specific things. It means a protocol or other specification which

(a) is supported and approved by an established, professional standards organization, and

(b) has achieved industry-wide acceptance and usage.

The WAP specification enjoys neither form of legitimacy. It is approved by no organization other than the WAP Forum itself, which has no standing as a professional standards body. Also, despite the claims made in the WAP Forum's promotional material, it has achieved little acceptance in the marketplace. Though marketing projections for use of WAP are very impressive, its actual usage in the U.S. remains limited.

The WAP Forum's use of the word "standard" implies that their specification carries some kind of formal authority; in fact, it has none. The WAP Forum's use of this term is marketing hype, not an objective statement of fact.

Any group of business interests can form a members-only club, generate a specification, publish it themselves, and call it whatever they wish. Regardless of the name they choose to attach to it, however, this does not make the result a "standard" in the conventional sense.

10.3 WAP - A Technical Failure

In addition to its procedural flaws, the WAP specification also includes serious technical deficiencies.

A detailed technical criticism of the WAP specification is beyond the scope of this document, and in this section we will do no more than provide a brief summary of the major issues. For a detailed analysis of WAP, the reader is referred to the article *W* Effect Considered Harmful* [**?**], in which author Rohit Khare argues the shortcomings and ultimate non-viability of the WAP specification.

The deficiencies in the WAP specification are glaring, obvious, and readily apparent to any competent data communications professional. A recent (January 2000) e-mail discussion thread on the IETF mailing list [**?**] makes this point quite plainly – this thread demonstrates clear consensus among professionals that the WAP specifications are not a technically sound engineering solution.

Many of the technical problems stem from a strategic design decision, made early in the design process. As noted in the Introduction, a new set of protocols are clearly needed to address the efficiency needs of mobile wireless devices. One approach to this would be to treat these devices as just another kind of Internet host. Under this approach, the existing Internet protocol architecture would remain in place, and to this would be added a small number of supplementary Internet protocols, designed to provide the power and bandwidth efficiency required by wireless devices and networks.

The other approach is to treat these mobile devices as a unique special case, requiring their own, entirely new set of protocols. Under this approach, the existing Internet protocol stack is discarded, and a completely new protocol stack is re-designed from scratch.

The WAP Forum made the early strategic design decision to take the latter approach. They have developed an entire stack of network protocols analogous to, but largely incompatible with, the existing Internet architecture. Not only has this approach required an enormous engineering effort on the part of the protocol designers and implementers, it has also given rise to a number of fundamental design errors.

10.3.1 User Interface Assumptions

A basic principle of data communications protocol design is that communications considerations must be de-coupled from user interface considerations. In other words, user interface issues must not be part of the protocol.

The WAP specification, however, violates this principle completely. It is tailored primarily to the physical characteristics of the hand-held mobile telephone, with its unique user interface characteristics. Accommodation is made to these characteristics throughout the specification, with the result that user interface issues are repeatedly combined and confused with communications issues. To put this in the language of data communications professionals: *WAP presumes presentation*.

The WAP designers justify this on the grounds that the combined technologies of wireless communications, and the mobile telephone, create a unique special case which warrants this departure from standard design principles. In fact, this is a strategic design error.

10.3.2 Extreme Accommodation to Existing Networks

Because WAP is essentially a marketing construct, one of its goals has been to create mindshare within every segment of the wireless industry. In order to accomplish this goal, WAP has made extreme accommodation to existing wireless networks. The WAP specification claims to accommodate almost all existing networks, including several which are already obsolete in their usage and general design. This has significantly and unnecessarily increased the complexity of the specification.

The WAP specification claims to support all the following wireless networks: CDPD, CDMA, GSM, PDC, PHS, TDMA, FLEX, ReFLEX, iDEN, TETRA, DECT, DataTAC, Mobitex, SMS, USSD, CSD, IS-136.

Some of these networks, such as FLEX and ReFLEX, are not general-purpose wireless networks, and were neither intended nor designed to run Internet-centric application protocols of any kind. WAP's decision to accommodate such networks makes little sense in engineering terms. This decision can only have been based on marketing considerations – each additional network supported provides WAP with one more promotional bullet point. Engineering concessions to marketing are a fact of life in the world of consumer products, but they have no place in an industry-standard protocol.

Wireless networks are rapidly standardizing on IP, the Internet Protocol. Most modern wireless networks (e.g. CDPD, Packet CDMA) already support native IP, and we can expect that others will do so in the very near future. The convergence of wireless networks on IP at Layer 3 (the Network Layer) is already a technological reality, and it is inevitable that it will eventually become standard on all networks.

Therefore the correct approach to wireless application standardization is to accept IP as the standard service at Network Layer, and implement high-efficiency protocols above Layer 3.

Furthermore, the result of accommodating all these networks is that the specification is no longer Internet-centric. WAP insists that the specification *is* Internet-centric, but this claim is without substance. If one tries to accommodate all existing technologies one cannot claim that the result is Internet-centric - one simply cannot have it both ways.

WAP claims to achieve accommodation of this very wide range of networks by means of two lower layer protocols: Wireless Control Message Protocol (WCMP), and Wireless Data Protocol (WDP).

WCMP is an out-of-place imitation of ICMP. Since there are no multiprotocol wireless operators, use of WCMP is always addressed as a service provider specific mechanism. In other words, WCMP is largely irrelevant.

WDP is roughly equivalent to UDP. The only rationale for its reinvention is to accommodate airlink addresses and airlink restrictions on packet size. The equivalent of WDP could have been accomplished on a per-network basis outside the scope of wireless applications. In fact, the existence of WDP may become a hindrance towards evolution of existing wireless networks towards IP.

In general, backward compatibility is a worthy design goal. But in this particular case it is a wasted effort. The convergence of wireless networks towards IP is already a technological reality. The strength and benefits of "IP On Everything" by far outweigh the efforts in accommodation and backward compatibility.

WAP's choice has been to accommodate all existing old wireless networks – a fact which betrays its underlying market motivation.

10.3.3 Excessive Re-Invention in the Name of Wireless

Existing Internet protocols do not adequately meet the requirements of wireless data communications – on this point we are in complete agreement with the WAP Forum. However, we believe that the correct way to design the required protocols is to do so within the framework of the existing Internet protocol architecture – the new protocols should be compatible with and re-use existing protocols as much as possible.

The WAP designers, however, have taken precisely the opposite view. Instead of re-using existing protocols, they have created an entirely new set of protocols from scratch.

The main key piece of new technology in WAP is the Wireless Transaction Protocol (WTP). In many ways, WTP is the heart of WAP. WTP is responsible for delivering efficient reliability to applications. Even in that area, WAP could have re-used existing technology. Specifically, T/TCP [**?**] and ESRO [**?**], which had already been published as Internet RFCs, could have been used. While there may be some reasonable objections to the use of T/TCP due to its "heaviness" because of bundling with TCP, there is no reason why ESRO could not have been used instead of WTP. In fact, WTP is a poor attempt at accomplishing the services of ESRO.

Aside from WTP, in most cases WAP's new protocols are essentially the same as the previously existing protocols, but with minor modifications that render them incompatible with the original standard. The WAP specification discards and then re-designs almost every existing Web standard – for example, WAP replaces UDP with WDP, TLS with WTLS, HTTP with WTP, HTML with WML, and ECMAScript with WMLScript.

This large amount of re-invention is entirely unnecessary. There are many places throughout the design of the WAP protocols where the existing protocol structure could have been left untouched, but complemented by the addition of a limited number of new protocols, designed for optimal efficiency.

The scope of the WAP specification has also been expanded beyond what is needed or reasonable (e.g. WCMP, WDP). Again, the WAP designers justify all this re-invention and expansion of scope on the grounds that wireless mobile devices present a unique special case, requiring a completely new set of protocols. In fact there is nothing specific to wireless applications which justifies this exhaustive degree of re-invention.

10.3.4 Vulnerable Wireless Transport Layer Security (WTLS)

In his paper *Attacks against the WAP WTLS Protocol* [?], author Saarinen describes in detail a number of security problems with WTLS. Here we provide a brief summary of the problems and their cause.

Although the WTLS protocol is closely modeled on the well-studied TLS protocol, a number of security problems have been identified with WTLS. These problems include: vulnerability to datagram truncation attack, message forgery attack, and a key-search shortcut for some exportable keys.

Most of the text in the WTLS specification has been adopted, word for word, from the TLS specification. However, many of the changes that were made by WAP Forum have led to security problems.

This is another case of the WAP Forum's deliberate deviation from the mainstream of the Internet causing difficulties.

10.3.5 Bungled Protocol Number Assignment

As Khare discusses in greater detail [?], WAP's port numbering strategy is another example of botched reference-by-copy. Instead of obtaining legitimate WAP ports in the IANA-registered space, WAP uses the temporary private port space in the range of 49152 to 49159. In addition to the port numbers, TLS ciphersuite codes and HTTP methods were also not registered with IANA. Instead, the WAP Forum has created its own parallel IANA equivalent called WINA.

This is another case of WAP Forum's deliberate deviation from the mainstream of the Internet in the name of wireless, and for the purpose of maintaining control.

10.4 WAP - A Basic Misconception

Beyond its procedural and technical flaws, we believe that WAP represents a fundamental misconception of what can feasibly be accomplished using cell phones, and what actual users are going to want to do with their phones.

Mobile Messaging, and Mobile Web Browsing, represent two very different forms of mobile communications activity. Mobile Messaging refers to the ability to send and receive personally directed messages, while Mobile Web Browsing refers to general information retrieval from anywhere.

Both of these bring undoubted value to the user, both can be accomplished on a cell phone, and the mobile user of tomorrow will certainly indulge in both activities. However, the value that the two activities bring to the user, and the suitability of the two activities to the cell phone, are very different.

Mobile Messaging allows important and/or time-critical information, which may require the user's immediate attention, to be pushed to him/her quickly. This is something which is of inherent and compelling value to the off-line, mobile user. By contrast, the desire of the mobile user to go back on-line for web access rarely has the same importance or urgency.

A basic question is: which of these two mobile applications represents the best initial value? We believe that Mobile Messaging is the right answer for initial mobile applications development.

10.4.1 The Wrong Answer Initially: Mobile Web Browsing

The basic goal of the WAP specification is to allow Internet web browsing using mobile phones. The assumptions underlying this goal are, first, that web browsing capability can be adequately accomplished on a mobile phone, and second, that this is something that will be of significant value to mobile phone users.

However, we believe that neither of these assumptions is correct. First, the cell phone of today is a totally inappropriate device for the purpose of accessing the mature Internet. Not only is the cell phone user interface completely inadequate for general web page display, but the wireless network medium also imposes severe limitations on the speed, immediacy, and reliability of web page access. It is simply not practical to surf the web using today's four-line text displays, over a slow, congested network with unreliable coverage.

As Kevin Maney observes in his article *Cell phones let the Web 'go mobile'* [**?**]

> "Web phones can be slow and frustrating. Click on The Weather Channel, for example, and the phone takes 6 or 7 seconds to send the request through the Sprint wireless network, into the Internet and to The Weather Channel's WAP-enabled Web server, then get back the next menu. On that menu, click "cities," then wait a few more seconds before getting a request to enter a ZIP code. You do that using the phone keypad. A few seconds later, you get the report. Only about 10-15 words fit on the screen at one time. You scroll down to read it."

It can be argued that future improvements in display technology may act to alleviate these difficulties, and this may very well be the case. However, the need for convenient portability of "unconscious carry" communications devices such as cell phones and pagers exerts tremendous design force towards reducing the size of these devices. The effect of this force is plainly apparent in the current trend towards extreme miniaturization of cell phones.

The design force towards miniaturization is something which is in direct opposition to display capability. For this reason, we believe that unconscious carry devices will continue to have severely limited display capabilities.

Second, there is the question of what people are actually going to do with the brand new medium of wireless data communications. How, and in what forms, will this medium become part of society? In ten years time we may have the answer, but today, nobody knows.

Of all forms of risk, prediction is the most gratuitous – yet none of us seems able to resist it. WAP's answer is that mobile web browsing will be enthusiastically embraced by society, and that it will be done via the cell phone device.

Our answer is that we doubt it. People will do things on a mobile basis that make sense to do while mobile – they will access the information that is most useful to them while away from the home or office. This includes such things as urgent e-mail messages, and highly specific urgent and important information. But it does not include general-purpose web browsing.

Web browsing is an interactive activity, for which the user requires a near real-time response. Furthermore, browsing, as the word indicates, is not an activity that has any urgency, and is therefore not something that there is a compelling reason to do while mobile. For both of these reasons, we believe that people will continue to do their web browsing at the home or office, and that web browsing will remain a marginal component of mobile communications.

It is true that the prospect of mobile Internet access has created tremendous excitement in the marketplace. And there is something magical about putting a cell phone in someone's hand, and providing a demonstration of live mobile Internet access. But the enchantment that this creates in the user stems from its technological novelty, not from its enduring day-to-day usefulness.

10.4.2 The Right Answer Initially: Mobile Messaging

This is not to say that all Internet data access is without value to the user. On the contrary, consumers certainly will use their mobile phones for Internet data access. However, the nature and types of data they will access will be appropriate to the medium.

They will access data that they have a need for and can use while mobile, that does not require near-synchronous system interaction, and that can conveniently be accessed via a mobile device.

The single application that satisfies these criteria better than any other is mobile messaging, or e-mail. Interpersonal messaging has already become an indispensable aspect of modern life, and is now the dominant application for the fixed Internet. We believe that society will adopt mobile messaging with the same enthusiasm as conventional e-mail, and that it will achieve similar dominance on the wireless Internet.

10.4.3 Unsupported Claims

There is presently an enormous amount of hype surrounding WAP. The WAP proponents have hyped its abilities far beyond what the consumer will actually be able to do with his or her cell phone.

There is a general perception that WAP will essentially put the entire Internet into the hands of the cell phone user. It is understood that a good deal of the *form* of a website may be lost in translation to the cell phone, though its basic textual content will be preserved. Beyond this, however, there is a perception that *any website can be accessed in this way*; i.e. that WAP will make the entire Internet content accessible via cell phone, just as it currently is via a conventional ISP.

However, this is not the case. WAP provides access to websites by translating the HTML coding of web pages into something that a cell phone can understand and display. What this means is that in order to get information from a website, the site must have a WAP-enabled server, and the server must be programmed to extract the website content that can be displayed on the miniature cell phone screen. In other words, if your favorite website is not WAP-enabled, you probably will not be able to access it through your cell phone.

For this reason, the tremendous hype surrounding WAP has yet to be backed up by commercially available WAP products and services. As Antony Bruno comments in his article *Market gap producing WAP alternatives* [**?**]

> "A running joke among industry players has the WAP acronym meaning Where Are the Phones?"

WAP phones and services are not available in the marketplace, because the promises of WAP are simply not real.

Not Device-Independent

The WAP specification is being promoted by the WAP Forum as a general-purpose wireless protocol, suitable for a wide variety of end-user devices, including PDAs such as PalmPilot. As noted in Section 10.3.1, however, WAP is in fact highly phone-centric – it is strongly oriented towards the mobile telephone physical device, despite the WAP Forum's claims of device-independence.

We note in passing that during the time that it was promoting the general-purpose nature of WAP, Unwired Planet, a founding member of the WAP Forum, changed its name to Phone.com. There are, of course, many good reasons why a company might wish to change its name. It strikes us as ironic, however, that while promoting the device-independence of WAP, this founding WAP Forum member abandoned a device-independent name in favor of a device-dependent one.

Limited Web Browsing Capabilities

The WAP Forum claims to bring web browsing capability to the cell phone. However, the cell phone device is simply not equal to this task. Because of the user interface limitations of the cell phone – its very small screen and limited keypad – the resulting web browsing experience is clumsy and impractical.

Furthermore, use of the WAP specification to access web content requires the use of *WAP gateways*, which translate the WAP-enabled website content into a format which the end-user can access. These gateways are controlled by the Service Provider (typically the wireless phone service provider), not the information provider. This usage model is very much contrary to the existing Internet usage model, in which the Service Provider plays the role of an entirely passive intermediary between the website provider and the website reader. In other words the Service Provider functions purely as a pipe. In this model new website content becomes immediately available to all Internet users as soon as it is created by the website provider. This unrestricted connection between the creators and consumers of Internet content has been one of the keys to its extraordinary growth and vitality. Because of this open characteristic the Internet has been able to grow organically – i.e. spontaneously, autonomously, and without planning, control, or approval by any central authority.

In the WAP model, by contrast, the WAP gateway operated by the Service Provider plays the active role of translating and storing web content, and therefore controls access to the content by the end-user. New websites and new web content do not become available to the end-user without the active participation of the Service Provider. This places a layer of control and authority between the creator and consumer of website content, greatly diminishing the potential for unrestricted and organic growth of the Internet.

Existing Technology Adequate

The premise of providing access to important information through a cell phone is certainly valid. As noted above, however, the nature and quantity of information that in practical terms can be delivered via a cell phone is severely constrained by the device itself.

The nature and quantity of information that can be delivered is sufficiently heavily constrained, that it can be adequately delivered using existing technologies.

The equivalent of most, if not all, of what WAP promises to deliver on a cell phone can very reasonably be accomplished using existing and already widespread technologies such as SMS. Indeed, this is already being accomplished today. Various companies, including Xypoint, AmikaNow!, Roku, ThinAirApps and Microsoft, are already providing services equivalent to WAP's claimed functionality. This is being done using existing cell phones and existing cellular services. For more information, and a more complete list of companies providing such services, see [**?**].

Voice Interface Adequate

The screen user interface limitations of a cell phone are so severe, in fact, that its data access capabilities are almost always better served by its voice interface – with which, of course, all cell phones come equipped.

The genuine utility of WAP on a cell phone resides in those applications for which a visual data interface is superior to a voice interface – that is, in those applications for which screen and keypad are better suited than speaker and microphone. Given the limitations of the cell phone screen and keypad, however, this is an extremely narrow range of applications. In other words, if all you have is a tiny screen and a miniature keypad, then in most cases you are better off using the voice interface.

Use of the voice interface to access important or time-critical information is already fairly widespread. The implementation of increasingly reliable speech recognition and powerful text-to-speech systems can be expected to make data transfer via the voice interface even more convenient.

10.5 Conclusion: WAP is a Trap

We have no argument with the notion that a world-wide standard is needed to address the requirements of wireless data applications. However, we believe that WAP is utterly unfit for this purpose. As we have shown in this article, WAP is the result of a closed design process within a members-only club. It remains tightly controlled by the WAP Forum, is crippled by patents, and is riddled with technical design errors. In the long run WAP is doomed to failure. In the short run it can only do harm to the industry and the consumer.

All of this could be the result of a series of colossal blunders by a well-meaning but spectacularly incompetent industry association. However, we do not believe that this is the case. We do not believe that the WAP Forum is well-meaning; on the contrary, we believe that their fundamental motivations are crass financial self-interest, coming at the expense of engineering and business integrity.

The WAP Forum could easily have taken steps to eliminate every one of the criticisms we have leveled against it, yet they have not done so. We invite them to tell us why.

The WAP Forum claim that WAP is an extension of the Internet, and is part of the Internet mainstream. Yet in no way has the development of WAP been consistent with Internet conventions. The specification could have been designed by an open design process, by the straightforward expedient of establishing open working groups and public mailing lists. There is ample precedent for this in the history of Internet protocol development. Yet the WAP Forum has not done so. Why not?

The WAP Forum could have ensured free and permanent availability of the specification by publishing them as RFCs, the mainstream method of publishing Internet protocols, and supported by extensive precedent. Again, they have not done so. Why not?

The WAP Forum could have worked diligently towards the goal of a patent-free protocol, by means of processes with well-understood industry precedent. Once more, they have not done so. Why not?

We can come to only one conclusion: WAP was designed to create unfair market advantage for its developers from the outset. They have maintained strict and close control of the protocol from the beginning, in complete violation of Internet conventions. The WAP Forum members have knowingly and deliberately incorporated their own patents within the specifications, and now are demanding royalties.

We can think of no better way to characterize such a process than as a trap. WAP is far from being an enabling force in the wireless industry. On the contrary, it is a gigantic and poorly-designed red herring created by narrow business self-interests. It is not a genuine engineering construct; it is a bogus marketing one.

A recent quotation from Greg Williams, Chairman of the WAP Forum, provides an excellent illustration of the WAP Forum's preference for members-only processes. Commenting on the recent highly public Geoworks patent infringement claim, he says [**?**],

> "Companies in the WAP Forum generally settle licensing and cross-licensing deals between themselves."

10.6 Preventing the Harm of WAP

What can be done to prevent the spread of WAP? There are several possible steps that in principle could be taken:

- Work to reform the WAP Forum.
- Spread the word about the WAP fraud.
- Reject WAP at engineering level.
- Reject WAP at consumer level.
- Adopt a viable alternative to WAP.

10.6.1 Reform the WAP Forum

One possibility would be to work with the WAP Forum – to engage them in a dialogue, and try to persuade them to correct the procedural problems described in Section 10.2. Among other things, this would require that they establish an open working group for maintenance of the protocol, initiate RFC publication of the protocol, and make every effort to lift all patent restrictions from the protocol.

However, this would amount to a complete reversal of the WAP Forum's mission and values, and it is naive to imagine that this is possible. At this point, we regard the WAP Forum as being beyond redemption.

This also leaves aside the question of what to do about the technical deficiencies of WAP – even with the WAP Forum's complete cooperation, a major effort would be required to create a sound engineering solution. Working to reform the WAP Forum, therefore, is not a useful approach.

10.6.2 Spread the Word about the WAP Fraud

Given that we can expect no help from the WAP Forum, the most useful thing that can be done quickly and easily is to spread the word about WAP. It is for this purpose that this exposé has been written.

Please join us in letting it be widely known that WAP is a trap. You may freely make and distribute copies of this article, provided the copyright and permission notices are preserved. You are encouraged to bring this article to the attention of the appropriate persons within your organization.

10.6.3 Reject WAP at Engineering Level

Rejecting WAP at engineering level means working to prevent WAP from being adopted in the design of devices and systems. This is primarily the responsibility of the engineering community within the wireless industry.

It is the responsibility of design engineers to evaluate the controversy surrounding WAP, and decide for themselves whether or not it is a sound engineering solution. If as an engineer you decide that it is not, then we encourage you to inform your manager of this, justify your position on technical grounds, and recommend alternatives.

To provide support for this, the Free Protocols Foundation maintains a set of informational resources on its website at http://www.freeprotocols.org/harmOfWap/main.html. These resources include references to various other papers and articles which corroborate the fraudulence of WAP. Please feel free to use these materials in any way you wish. We also invite you to contribute to the information pool at the Free Protocols Foundation – any comments, articles or other information may be submitted to the FPF via our website.

10.6.4 Reject WAP at Consumer Level

Rejecting WAP at consumer level means encouraging the end-users of hand-held wireless devices to refuse to purchase WAP devices – in other words, to boycott these devices.

However, the WAP question is a very complex technical and business issue, and it is not easy to convince the consumer that this is something worth caring about. A successful boycott requires an immediate, gut-level understanding of the issues on the part of the consumer. The WAP issue is not something that can easily be condensed into a ten-second sound bite.

In any case, WAP is not sufficiently widespread for a boycott to be effective. For these reasons we do not consider a consumer boycott to be a useful approach at this point. For the moment, WAP must be dealt with by the industry, not the consumer.

10.6.5 Adopt an Alternative to WAP

Regardless of the shortcomings of WAP, the fact remains that at some point the wireless industry must agree upon a standard protocol for efficient data communications. Ultimately, therefore, WAP can be displaced only by the adoption of a suitable alternative.

One traditional source of Internet protocols is the Internet Engineering Task Force, or IETF. To our knowledge, however, the IETF does not currently have a working group assigned to this task, and so no protocol can be expected from them in the near future. Even if the IETF were to assign a working group to this immediately, it typically takes 18 months to achieve a workable first-draft protocol. This time frame is far too long to address the industry's immediately pressing need.

Other traditional sources of protocols are private industry, and the academic community. However, thus far a suitable protocol has been forthcoming from neither of these sources. Although there is general consensus within the industry that an alternative protocol to WAP is desperately needed, no such protocol has yet been publicly proposed.

In this article, we would like to be the first to present an alternative: LEAP, the Lightweight and Efficient Application Protocol. LEAP is immediately available, and has all the characteristics required to displace WAP and become the basis of an industry standard. A brief description of LEAP is provided in the following section.

To the best of our knowledge, LEAP is the only viable alternative to WAP. However, we invite readers of this article to seek out and bring to our attention any other alternatives that may exist. At the Free Protocols Foundation, we are ready to support and publicize any viable, patent-free alternative to WAP that is made known to us. Such alternative protocols will be referenced at the FPF website at http://www.FreeProtocols.org, and included in future revisions of this article.

In summary, the best ways to prevent the harm of WAP are to spread the word, reject WAP at engineering level, and adopt alternatives. We encourage readers of this article to join us in opposing WAP in all of these ways.

10.7 LEAP: One Alternative To WAP

Fortunately, an alternative to WAP exists: LEAP, the Lightweight and Efficient Application Protocol. LEAP consists of a set of high-performance, efficient protocols which are ideal for mobile and wireless applications. LEAP presently includes the following component protocols:

- **Efficient Short Remote Operations Protocol (ESRO).**
  ESRO can be thought of as a reliable connectionless transport mechanism, forming the foundation for the development of efficient protocols when TCP is too much and UDP is too little. ESRO was published as RFC-2188[**?**] in September of 1997. The ESRO protocol is publicly maintained and developed by ESRO.org at http://www.esro.org/.

- **Efficient Mail Submission and Delivery Protocol (EMSD).**
  EMSD is designed to address the mobile messaging (e-mail) application. EMSD was published as RFC-2524 [**?**] in March of 1999. The EMSD protocol is publicly maintained and developed by EMSD.org at http://www.emsd.org/.

- **Efficient Hyper Text Delivery Protocol (EHTD).**
  EHTD is currently undergoing development and will provide efficient delivery of web pages. It is currently undergoing public development at http://www.freeprotocols.org/EHTD.

Open source implementations of the ESRO and EMSD protocols are being made available as free software at: http://www.MailMeAnywhere.org/.

The LEAP protocols, in combination with existing Internet protocols, properly address the same set of requirements that WAP claims to address. They have all the preferred protocol characteristics described in Section 10.1.2. They are published as Internet RFCs, are publicly maintained, and suffer from none of the technical deficiencies ascribed to the WAP specifications. In addition, the LEAP protocols fully conform to the patent-free policies and procedures of the Free Protocols Foundation [**?**].

A comparison of LEAP to WAP, and justification of LEAP as the basis of an industry standard, is provided in *LEAP: One Alternative To WAP* [**?**], a companion article to this one. This article is available at the Free Protocols Foundation website at http://www.FreeProtocols.org.

A complete and detailed description of LEAP is provided in *The LEAP Manifesto* [**?**], available at the LEAP Forum website at http://www.LeapForum.org.

Anyone interested in participating in the development of the LEAP protocols is invited to join the mailing lists hosted at the above-mentioned websites. LEAP's development process is truly open and non-exclusive. There are no membership fees. Participation in the LEAP development process requires only that you commit to the patent-free intent of LEAP and the Free Protocols Foundation.

Chapter 11

LEAP: One Alternative to WAP

11.1 Introduction

Over the last few years, data communications has expanded dramatically and forcefully into the wireless environment. A major new Internet reality is that of wireless networks, providing service to legions of miniaturized, hand-held mobile devices. This reality has placed an entirely new set of requirements on the underlying communications protocols: they must now provide the power efficiency demanded by hand-held wireless devices, together with the bandwidth efficiency demanded by wide area wireless networks.

Existing Internet protocols do not adequately meet these requirements. Therefore a new generation of efficient protocols is needed, to satisfy the demands of wireless applications. At some point, the wireless data communications industry must agree on a single set of protocols that satisfies its requirements.

11.1.1 The WAP Trap

In April 1998, a business association called the WAP Forum published the **Wireless Application Protocol**, or **WAP**. WAP is a set of specifications for wireless data communications using hand-held devices such as mobile phones and palmtop computers. The WAP specification provides the users of these devices with mobile data communications capabilities such as web-browsing and e-mail.

The WAP specification purports to be an open, license-free protocol that will unify and promote the growth of the wireless industry. The WAP Forum claims that the WAP specification satisfies all the requirements necessary to become the industry standard, and is aggressively promoting it as such.

In a previous article entitled *The WAP Trap* [**?**], however, we have argued that WAP is utterly unfit for its claimed purpose. In that article we described the desirable characteristics of enduring, industry-building protocols, and we demonstrated that the WAP protocols lack all of them.

Among other things we showed that WAP is the result of a closed design process within a members-only club, that it remains tightly controlled by the WAP Forum, is crippled by patent restrictions, and is riddled with technical design errors.

Our conclusion was that the WAP specification is not a genuine engineering construct; it is a bogus marketing one. Its purpose is to create unfair market advantage and bring short-term financial gain to its developers, rather than to provide long-term benefit to the industry at large and the consumer. Far from being an enabling force in the wireless industry, WAP is a poorly-designed red herring created by narrow business self-interests.

In the long run WAP cannot survive as a viable solution. In the short run, however, it can do considerable harm to the industry and the consumer.

In *The WAP Trap* we went on to discuss the steps that can be taken to prevent this harm. A crucial step will be for the industry to adopt an alternative to WAP as soon as possible. We concluded the article by presenting one alternative: LEAP, the Lightweight & Efficient Application Protocols.

11.1.2 About this Document

In the present article, we will carry on where the previous article left off. The scope of *The WAP Trap* was limited to a critique of WAP, without actively promoting any particular alternative. The present article, on the other hand, is frankly partisan; our purpose here is to promote LEAP as an open alternative to WAP.

The authors of this article are participants in Free Protocols Foundation (FPF) activities, under whose auspices this article is being written. The mission of the FPF is to provide support for the development, maintenance, and promotion of patent-free protocols and software. It provides a forum in which developers can declare publicly that the protocols and/or software they have developed are intended to be patent-free, and that it is their intention to keep them permanently patent-free.

In addition, where the existence of patented components within protocols and/or software threatens their unrestricted usage and implementation, the FPF supports the promotion of patent-free alternative protocols and/or software. It is for this purpose that the current article is being written: to promote LEAP as a patent-free alternative to WAP.

In this article we will describe LEAP from both a technical and a procedural point of view. We will compare it to WAP, and will demonstrate that it has all the desired protocol characteristics that WAP lacks. Our conclusion will be that LEAP is destined to play a key role in the growth of the Mobile Messaging industry.

This article is one of several we have written that analyze the current status of the wireless data communications industry, criticize WAP, and present our view of what is truly needed to promote the growth of the industry. Related articles are:

- *The WAP Trap* [**?**]. Provides a critique of the WAP specification, and sets the stage for the current paper. Available at http://www.freeProtocols.org/wapTrap.
- *The LEAP Manifesto* [**?**]. Provides a complete analysis of the industry, and a detailed description of the LEAP protocols. *The LEAP Manifesto* is available at http://www.LeapForum.org/LEAP/Manifesto/roadMap/index. html.

11.2 The Need for Efficiency

Engineering is the art of making intelligent trade-offs between conflicting requirements. A perennial engineering trade-off is that which must be made between the need for *simplicity*, and the need for *performance*. In the case of wireless data communications, performance means such things as data transfer speed, power efficiency, and bandwidth efficiency.

The 1980s and 1990s were the decades of simple protocols - protocols such as the very aptly named Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP). A great deal of the success of these and other Internet protocols can be attributed to their simplicity.

The first generation of network engineers and network operators were only able to view network communications in relatively simple terms. It was appropriate to cater to that simplicity with simple protocols. A key reason for the success of these early protocols is the lack of technical sophistication on the part of first-generation network engineers and operators.

Simple protocols are easier to make widespread than "good" protocols (meaning those which have better capabilities and performance), for the basic reason that network engineers and operators are able to adopt and implement simple protocols much more easily than "good" protocols.

However, things have changed. Network communications has now expanded into the wireless and mobile data communications arena, and *wireless applications demand efficiency*. The move to wide-area wireless has significantly shifted the location of the ideal engineering balance between simplicity and performance - moving it away from simplicity, and towards performance.

We therefore need a new generation of high-performance, efficient protocols, to cater to the demands of wireless applications. The point is sometimes made that the need for efficiency in the wireless arena is a temporary one – that advances in wireless engineering technology in the form of third generation (3G) systems will eliminate existing bandwidth limitations, obviating the need for efficient protocols. As long as the capacity of wireless networks remains finite, however, the need for efficiency will persist. Efficient usage is an inherent requirement for any finite resource, therefore the requirement for efficient bandwidth usage and battery longevity will remain.

Thus far, professional protocol and standards producing associations, most notably represented by the IETF, have failed to produce an acceptable specification. The IETF continues to represent the tradition of simple protocols, a tradition which wireless communications has now made obsolete. Unfortunately, the IETF remains rooted in this tradition, and has not adapted to the new realities of wireless communications. Until it does so, the IETF will remain ineffective as a protocols and standards body. In the area of efficient protocols, the IETF is simply bankrupt.

11.3 LEAP: The Lightweight & Efficient Application Protocols

It is now time for a new generation of protocols to be implemented, designed to address the need for *performance*, rather than simplicity.

The **Lightweight & Efficient Application Protocols**, or **LEAP**, are designed precisely to address this need. LEAP is the general framework for a set of high-performance, efficient protocols which are ideal for mobile and wireless applications. LEAP is designed to address the technical requirements of the wireless data communications industry, and is oriented towards providing the greatest benefit to the industry and the consumer.

The LEAP protocols are patent-free, and open-source implementations of the protocols are being made available for a variety of devices and message-center platforms. The protocols are thus ready and available, and can be quickly distributed and implemented as a viable alternative to WAP.

11.3.1 A Brief History of LEAP

LEAP originated in 1994 as part of the research and development initiatives of McCaw Cellular's wireless data group (now AT&T Wireless). The development work that would eventually lead to LEAP was initially undertaken in the context of the CDPD network; its scope was later expanded to include the Narrowband PCS network also.

By 1996 McCaw Cellular was fully committed to paging, had recently purchased two nationwide narrowband wireless PCS licenses, and wished to develop an efficient wireless message transport and delivery system. Neda Communications, Inc., an independent consulting company working under contract to McCaw Cellular, played a significant role in the development of the required system. Neda Communications had also been involved from the outset in the development of the CDPD specification.

In 1997 however, soon after the purchase of McCaw Cellular by AT&T, the company abandoned narrowband PCS paging altogether. Prior to this event, Neda Communications had secured from AT&T the necessary rights to continue independent development of the protocols. Therefore, recognizing the eventual future need for these protocols, Neda then undertook to continue development of the protocols independently of AT&T. They were eventually completed by Neda, published as RFCs, and now form the cornerstone of the LEAP protocols.

11.3.2 Technical Overview of LEAP

In this section we will provide a brief technical overview of the LEAP protocols. For a detailed description of LEAP, refer to *The LEAP Manifesto* [**?**], available at
http://www.LeapForum.org/LEAP/Manifesto/roadMap/index. html.

LEAP is a set of wireless application protocols that are optimized for delivering small messages over wireless networks. Wireless networks are constrained by bandwidth limitations, and the hand-held devices they serve are constrained by limitations such as display size, battery capacity, and memory capacity. These constraints place a high premium on the efficiency of data transfer.

The LEAP protocols are up to five time more efficient than the ubiquitous SMTP e-mail messaging protocols. This increased efficiency translates into longer battery life for mobile phones, PDAs and other wireless Internet devices.

Layering of LEAP

The LEAP protocols are layered. The lower layer, called Efficient Short Remote Operations (ESRO), provides reliable connectionless transport services which can be used for a variety of applications. For example, in addition to mobile messaging services, ESRO can be used as a transport service for credit card verification applications and efficient micro browsers. On top of ESRO is the layer called EMSD. EMSD is a messaging protocol that is highly optimized for the submission and delivery of short Internet e-mail messages.

Various other LEAP protocol components are in the process of being designed and implemented. See *The Future of LEAP* article in *The LEAP Manifesto* for more details.

ESRO, Efficient Short Remote Operations

All efficient applications have the requirement for an efficient transport mechanism. For this reason, the initial focus of the protocol development effort has been on creating a general efficient transport mechanism. The resulting protocol is referred to as Efficient Short Remote Operations, or ESRO. ESRO is a reliable connectionless transport mechanism, forming the foundation for the development of efficient protocols when TCP is too much and UDP is too little.

ESRO was published in September 1997 as Internet RFC-2188 [**?**]. Additional information about ESRO is available at http://www.esro.org/

EMSD, Efficient Mail Submission and Delivery

The Efficient Mail Submission and Delivery (EMSD) protocol is built on top of ESRO, and is designed to address the Mobile Messaging application. EMSD provides for the submission and delivery of short (4 kilobytes or less) Internet e-mail messages. EMSD meets or exceeds the level of functionality, reliability and security provided by the existing SMTP protocols. EMSD is a great deal more efficient than existing Internet e-mail protocols.

EMSD was published in March 1999 as Internet RFC-2524 [**?**]. Additional information about EMSD is available at http://www.emsd.org/

Initial Focus: Mobile Messaging

The need for efficient protocols extends across all aspects of wireless data communications, including e-mail, web browsing, and other applications. The LEAP architecture accommodates all of these applications. The initial LEAP protocols, however, are designed to support the Mobile Messaging application, since this is the dominant application for wide-area wireless networks. Subsequent LEAP protocols are expected to address other applications as necessary.

11.3.3 Processes and Procedures

We believe that a public protocol must conform to each of the following basic, fundamental principles:

- Patent-freedom
- RFC publication
- Maintenance by open Working Groups

Each of these provides a vital assurance of protocol integrity. Patent-freedom ensures that a patent-holder cannot subvert free-market competition among products and services based on the protocol. RFC publication ensures that the protocol is freely available to anyone who wishes to use it. And maintenance by open Working Groups ensures that development of the protocol takes place by democratic, rather than oligarchic, processes.

This trilogy of principles represent the most basic guarantees of the integrity of a protocol.

The LEAP protocols are intended to be open in the fullest sense of the word; they are intended to be freely and permanently available, subject to public review and revision, and without usage restrictions of any kind. Therefore the processes and procedures used throughout the development and maintenance of the LEAP protocols have been such as to endow them with these characteristics, and to ensure their integrity as public protocols.

Complete details of the LEAP development process are provided in a separate article within *The LEAP Manifesto* entitled *The LEAP Protocol Development Model*. The major aspects of the development process are summarized in the following sections.

Freedom from Patents

As discussed in *The WAP Trap*, a highly desirable attribute of an industry standard protocol is that it be free from patents. The presence of patented components within a protocol undermines the ultimate purpose of the protocol: its unrestricted adoption and usage.

The development and maintenance of the LEAP protocols conforms fully to the policies and procedures of the Free Protocols Foundation. In particular, Neda has declared to the Free Protocols Foundation that the LEAP protocols are patent-free to the best of its knowledge, and that it intends to keep them patent-free permanently. For more information see `http://www.FreeProtocols.org`.

RFC Publication

Both protocols have been published as Internet RFCs; ESRO in September 1997 as RFC-2188 [**?**], and EMSD in March 1999 as RFC-2524 [**?**]. RFC publication is the mainstream Internet publishing procedure, ensuring that the protocols are freely, easily and permanently accessible to anyone who wishes to use them.

Open Maintenance Organizations

To provide an open forum for the continued development and maintenance of the LEAP protocols, Neda has established a public organization for each protocol.

The ESRO and EMSD protocols are maintained, respectively, by ESRO.org at http://www.esro.org/, and by EMSD.org at http://www.emsd.org/.

Each of these organizations allows public review of the respective protocol, and provides mechanisms for enhancement of the protocol as a result of collective experience.

WAP	LEAP
Subject to known patent restrictions	Patent-free
Self-published by the WAP Forum	Published as Internet RFCs
Revisions subject to change without notice	All revisions permanently fixed
Maintained by the WAP Forum	Maintained by open working groups
Re-invention of existing protocols	Efficiency-optimizing extensions to existing protocols
Tailored to mobile phone user interface characteristics	User interface independent
Inherent security vulnerability	Imposes no security assumptions
Inconsistent protocol number assignment	Consistent protocol number assignment
Initial focus: web browsing	Initial focus: messaging

Table 11.1: WAP versus LEAP

Any interested person may participate in the further development of the protocols. Participation in the development process is entirely open and non-exclusive; there are no membership fees. The only requirement is that participants must adhere to the principles and procedures of the Free Protocols Foundation, thus ensuring that the protocols remain permanently patent-free.

11.4 Comparison of LEAP to WAP

In *The WAP Trap*, we enumerated the characteristics of the WAP specifications that make them wholly unfit to play the role of an enabling industry protocol. These characteristics are summarized in Table 11.1, along with the corresponding characteristics of the LEAP protocols.

11.4.1 Patent Restrictions

As noted in *The WAP Trap*, the WAP specifications include patented components. Unlike WAP, the LEAP protocols are entirely patent-free.

11.4.2 Openness of Publication

As noted previously, the LEAP protocols are published as Internet RFCs, ensuring permanent, unrestricted availability of the protocols. The WAP specifications, on the other hand, are self-published by the WAP Forum, and therefore do not carry the same assurances of unrestricted availability. The availability and permanence of the WAP specifications is only as good as that of the WAP Forum itself.

Furthermore, in order to download any particular WAP specification, a user must agree to a license agreement. By contrast, the LEAP protocols may be downloaded and distributed without any restrictions.

In addition, the WAP Forum's publishing philosophy carries no guarantee of stability. As of February 2000, each WAP specification carries on its title page the disclaimer, "This document is subject to change without notice." By virtue of the RFC publication process, on the other hand, individual revisions of the LEAP protocols are permanently fixed.

11.4.3 Openness of Maintenance

LEAP's open maintenance processes are also in sharp contrast to WAP. Participation in the development of the WAP specifications requires payment of the $27,000 WAP Forum membership fee (as of February 2000), and takes place entirely behind closed doors. Unlike WAP, the LEAP protocols are maintained by public maintenance organizations in which anyone is free to participate.

11.4.4 Technical Deficiencies

The WAP protocols also include numerous technical deficiencies. For example, WAP is a broad-scope re-invention of existing protocols. The LEAP protocols, by contrast, consist of a small number of independent protocols that complement existing Internet protocols. Other WAP deficiencies are listed in Table 11.1; for a detailed discussion, see *The WAP Trap*.

11.4.5 Initial Focus

There are also significant conceptual differences between LEAP and WAP, of which we will mention two here. First, LEAP is primarily oriented towards the mobile messaging (i.e. e-mail) application, whereas WAP is primarily oriented towards mobile web browsing. We believe that this represents a serious misunderstanding of the mobile data communications industry on the part of the WAP Forum. Hand-held mobile devices are extremely well-suited to the e-mail application, whereas their severe user interface limitations render them highly ill-suited to web browsing.

Second, LEAP and WAP take very different approaches to the messaging application. The LEAP approach, embodied in the EMSD protocol, is a complete and efficient submission and delivery model. The WAP approach, on the other hand, is a mailbox access and selective message retrieval model.

A consequence of this is that the WAP protocol has several unresolved issues relating to message delivery. For example, the WAP protocol does not support the "push" model of message delivery, in which time-critical messages are actively delivered to the recipient. The LEAP protocol, by contrast, fully supports the "push" model.

11.4.6 Hype versus Reality

Our view of the evolution of the wireless Internet industry is illustrated in Figure 11.1. The early history of this industry is already known to us; in recent years the industry has undergone extremely rapid growth. And in the long term, there is general consensus among analysts that the industry is destined for continued strong and sustained growth.

So the early history is known, and the eventual history we can predict with confidence. But what about the more immediate future? Our view is that, largely thanks to the WAP Forum, the industry has been significantly over-hyped, with the result that expectations now greatly exceed realities. Our prediction is that this period of soaring expectation will be followed by a period of general disillusionment and frustration, as these expectations are inevitably disappointed.

Sooner or later the industry must adopt a more realistic understanding of its technological and business challenges. Part of this understanding will consist of the recognition that the wireless industry must adopt a single set of truly open protocols. Only then will the industry be able to undergo stable and sustained growth.

WAP represents the era of hype and disillusionment. LEAP represents the era of realism and maturity.

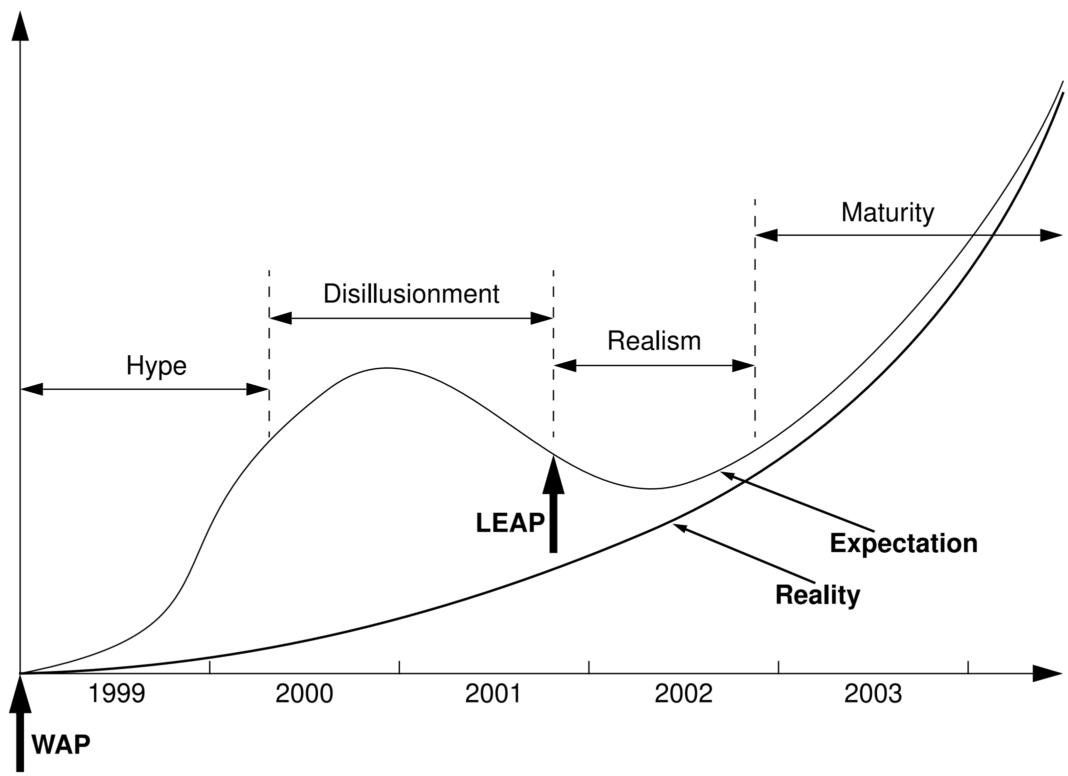


Figure 11.1: Wireless Internet Hype vs. Reality

11.5 Making LEAP Widespread

Thus far our discussion has been entirely theoretical; we have demonstrated on paper that WAP is not viable, and that LEAP has all the characteristics necessary to be considered a viable alternative. However this is all academic until the protocols are implemented as software and deployed in real world systems.

In order for the LEAP protocols to become widely used, they must be implemented in the form of software solutions that are readily available for deployment by end-users. To this end, we have created software implementations of the protocols for most common platforms. Protocol engines have been implemented in the form of portable code which has been ported to a variety of platforms. On the device side, software has been implemented for pagers and cell-phones; for hand-held PCs and Palm Pilot (Palm OS, Windows CE, Palm PC); for Windows 98, Windows 95, and Windows NT; and for Pine (UNIX, Windows, DOS). On the message center side, software has been implemented for Solaris, Linux and NT.

All of this software will be made publicly available in the form of free software in open-source format. At present, we have created the structures necessary to allow ready access and downloading of the software in binary form. Foundation libraries of LEAP protocol engines called the "Open C Platform (OCP)" are subject to the GNU Library General Public License and are available as open-source software.

The software is being made available at http://www.MailMeAnywhere.org/.

We expect to have the ESRO protocol engine software components subject to the GNU General Public License available at this location by September 2000. We expect the availability of the entire suite of open-source software implementations described above to be completed by December 2000.

As noted above, the initial emphasis of LEAP is on the mobile messaging application. Protocol engines are only a single component of a bigger picture; in order to provide complete solutions to the user it is necessary to integrate these protocols into other existing pieces of software. Fully-integrated solutions which combine LEAP with other open-source and free software packages such as qmail, sendmail, fetchmail will also be made available.

We invite those interested in using, enhancing, porting and integrating this software to join the relevant mailing lists at http://www.MailMeAnywhere.org/

We will also initially "prime the pump" by providing free subscriber services through ByNumber.net and ByName.net. This will provide initial support for the implementation of the protocols in end-user devices. Usage of the protocols among a sufficient number of user devices will then provide the motivation for usage among the message center systems.

11.6 Other Alternatives to WAP

In this article we have promoted LEAP as one alternative to WAP. An obvious question is: Are there any other alternatives?

A traditional source of Internet protocols is the Internet Engineering Task Force, or IETF. To our knowledge, however, the IETF does not currently have a working group assigned to this task, and so no protocol specification which addresses the requirements for efficient Mobile Messaging can be expected from them in the near future. Even if the IETF were to assign a working group to this immediately, it typically takes 18 months to achieve a workable first-draft protocol. This time frame is far too long to address the industry's immediately pressing need.

Other traditional sources of protocols are private industry, and the academic community. However, thus far a suitable protocol has been forthcoming from neither of these sources. There is general consensus within the industry that an alternative protocol to WAP is required. Apart from LEAP, however, no such protocol has yet been publicly proposed.

To the best of our knowledge, therefore, LEAP is the only viable open and patent-free alternative to WAP.

11.7 Summary

All of the basic components that are needed to launch LEAP are complete, in place, and ready to go. These components are:

The Protocols Themselves. The protocols are well-designed, meet all the technical requirements of the industry, and are published as RFC-2188 and RFC-2524. The complete text of the RFCs is available at `http://www.rfc-editor.org`.

Freedom from Patents. The protocols have been declared to the Free Protocols Foundation as patent-free. For more information see `http://www.FreeProtocols.org`.

Open Maintenance Organizations. The protocols are maintained by open and public organizations at `http://www.esro.org`, `http://www.emsd.org`, and `http://www.LeapForum.org`.

Open-Source Software Implementations. These are in the process of being made available for all major platforms and end-user devices. For details see `http://www.MailMeAnywhere.org`.

Free Subscriber Services. Provided to support initial deployment of the protocols in end-user devices. For details see `http://www.ByNumber.net` and `http://www.ByName.net`.

Together, these components represent a complete recipe for the success of LEAP. The protocols themselves are open and immediately available, and open-source implementations of the protocols are in the process of being made available as free software.

The combination of free protocols and open-source software is something which has enormous power. It is this combination of factors which has driven the overwhelming success of other industry standards such as Linux and the Web (HTTP/HTML). We believe that this same combination of factors will drive the acceptance of LEAP in the wireless data communications industry.

Finally, we do not claim that LEAP is technically ideal – like all engineering solutions it includes compromises. What we do claim is that LEAP is a good solution, and that its processes have integrity. Where the LEAP protocols fall short of the industry needs, the open maintenance processes will provide a mechanism by which they can evolve into a better solution.

11.7.1 The LEAP Manifesto

Every aspect of LEAP is described in *The LEAP Manifesto* [**?**], available at http://www.LeapForum.org/LEAP/Manifesto/roadMap/index. html. *The LEAP Manifesto* includes a technical description of the LEAP protocols themselves, and a description of all the components required to encourage their widespread usage. *The LEAP Manifesto* consists of the following articles:

Executive Summary. An overview summary of the entire LEAP Manifesto.

Overview of the LEAP Protocols. A general overview description of the LEAP protocols.

The LEAP Protocol Development Model. A description of the processes used to develop the LEAP protocols, and how and why these processes differ from the conventional development process. This article also includes a criticism of the IETF protocol development processes.

EMSD: The LEAP E-Mail Component. A technical description of EMSD, the e-mail component of LEAP.

ESRO: A Foundation for the Development of Efficient Protocols. A technical description of ESRO, the transport mechanism component of LEAP.

Efficiency of EMSD. A technical paper analyzing the efficiency characteristics of EMSD and comparing its efficiency to other e-mail protocols.

EMSD on Windows CE. A technical paper describing the architecture and implementation of EMSD on Windows CE devices.

EMSD on Palm OS. A technical paper describing the architecture and implementation of EMSD on Palm OS devices.

A Brief History of LEAP. A summary of the major events in the evolution of the LEAP protocols.

The Future of LEAP. A description of the planned future development of LEAP, including descriptions of several LEAP-based products and services which are currently under development.

The WAP Trap. A detailed criticism of a set of specifications called the Wireless Application Protocol, or WAP. This article demonstrates that WAP is entirely inappropriate to play the role of a Mobile Messaging industry standard.

LEAP: One Alternative to WAP. A point-by-point comparison of the LEAP protocols to the WAP specifications. This article compares and contrasts LEAP with WAP, and demonstrates that LEAP has all the desired characteristics of an industry-enabling protocol that WAP lacks.

Operation WhiteBerry. A description of how all the capabilities of the closed RIM BlackBerry mobile messaging solution can be duplicated using existing software implementations of LEAP, and existing off-the-shelf hardware components.

Strategy for Making LEAP Widespread. A description of our strategy for encouraging widespread usage of the LEAP protocols, including the distribution of open-source software implementations of the protocols, and the availability of free subscriber services.

Trying Out LEAP. A step-by-step, hands-on demonstration of how the LEAP protocols can be used to turn any Windows CE device into a fully functional Mobile Messaging device.

Lessons from History: Comparitive Case Studies. An analysis of the factors which lead to the success or failure of protocols, including discussions of several historical case studies.

The Mobile Messaging Industry. An overview of the Mobile Messaging industry, and a description of the essential factors that are required for its long term success and growth.

Chapter 12

WAP Scraps

What can be salvaged from what remains of WAP?

12.1 Introduction

We first published the article *The WAP Trap: An Exposé of the Wireless Application Protocol* [**?**] in April 2000. At that time it was the most comprehensive condemnation of WAP written to date. In it we demonstrated that WAP is crippled by patents, the result of a closed design process, inappropriately controlled by the WAP Forum, and riddled with technical design errors. We exposed WAP for what it is: a fraudulent marketing construct. Our conclusion was that WAP must be rejected and replaced with a set of truly open, patent-free, technically sound, mainstream Internet protocols.

12.1.1 Claiming the Day

In April 2000 we were one of a relatively small number of voices sounding the WAP alarm. At that time WAP was massively over-hyped, and a major portion of the wireless industry had succumbed to the hype. To use a phrase right out of the WAP hype machine: WAP was hot.

But in September 2001, 17 months after initial publication of *The WAP Trap*, our analysis and predictions have been convincingly validated.

The message in *The WAP Trap* resonated within the industry, and it has experienced widespread distribution and readership. Partly because of this, there is now a growing awareness of the fundamental fraudulence of WAP. The engineering community was never seriously taken in by WAP in the first place, and our raising of the alarm has had the desired effect among the business and media communities. Numerous articles have been published which support our position, including:

- *WapLash*. Meg McGinity, Inter@ctive Week, July 31, 2000. [**?**]
- *WAP 2.0: Mature Enough for Flight?* Keri Schreiner, IEEE Internet Computing, November/December 2000. [**?**]
- *WAP Usability: Deja Vu: 1994 All Over Again*. Marc Ramsay and Jakob Nielsen, Nielsen Norman Group, December 2000. [**?**]

The body of published articles includes several which quote us directly or otherwise build on our work. The above articles and others are available on the Free Protocols Foundation website at `http://www.FreeProtocols.org/harmOfWap/main.html`. We will continue to augment the Free Protocols Foundation library with additional relevant articles as they appear.

In addition, *The WAP Trap* has now been translated into French, under the title *Le WAP a la Trappe*. The translation of *The WAP Trap* into French represents another step forward in our campaign to expose WAP. Both English and French versions of the paper are available in HTML, PDF and PostScript formats on the Free Protocols Foundation website at `http://www.FreeProtocols.org/wapTrap/index.html`.

Though the tide of favor has turned against it, the WAP hype machine continues to operate. And there remain many within the industry who are still unaware of the problems with WAP. We will continue to counter the WAP hype by writing and distributing articles such as *The WAP Trap* and *WAP Scraps*.

However, the primary purpose of this article is not just to say NO to WAP. This article focuses on what needs to be done after WAP.

12.1.2 Mobile Web Browsing: An Open Industry Model

As we discussed in *The WAP Trap*, WAP has many shortcomings. But one of the major issues from a consumer-acceptance point of view is that it represents the wrong starting-point wireless Internet application. Though wireless web browsing certainly has its place, its end-user value proposition is entirely overshadowed by that of another wireless Internet application: Mobile Messaging.

This statement is fully supported by the user experiences and market acceptance of these two applications. The extremely poor end-user experience of WAP-based web browsing is very well documented in the Nielsen Norman Group field study report *WAP Usability: Deja Vu: 1994 All Over Again* [**?**]. By contrast, the end-user value of Mobile Messaging is well evidenced by the market acceptance of BlackBerry and other messaging systems, which are enjoying widespread popularity.

BlackBerry and other Mobile Messaging solutions are experiencing this popularity despite the fact that they are all closed, proprietary systems. In order for the Mobile Messaging industry to reach its full potential, these closed systems must be replaced by an *open* industry model, based on truly open and free protocols. All the components required to enable this, including the necessary open protocols, are now in place; and the development of the open Mobile Messaging industry is now assured. For a detailed discussion of the open Mobile Messaging industry, see the Manifesto article *Operation WhiteBerry* [**?**].

As in the case of Mobile Messaging, an open industry model is essential in order for the Mobile Web Browsing industry to reach its full potential. With the open Mobile Messaging industry well on its way, it is now time to focus attention on the development of the open Mobile Web Browsing industry.

12.1.3 About this Document

This paper is a follow-on paper to *The WAP Trap*. Both papers are endorsed by and published by the Free Protocols Foundation (FPF). The FPF is an independent public forum dedicated to the support of patent-free protocols and software. The FPF regards protocol and software patents as being highly detrimental to the industry and the consumer, and part of the FPF mandate is to oppose exceptionally harmful patents and patented protocols when they appear. One such patented protocol is WAP.

The FPF is a U.S. non-profit organization, and is tax-exempt under Section 501(c)(3) of the Internal Revenue Service regulations. All monetary contributions made to the FPF are tax deductible in accordance with these regulations. Any organization or individual wishing to support the goals of the FPF is encouraged to participate by joining the FPF mailing list, or by making an appropriate donation. For more information see the FPF website at `http://www.freeprotocols.org`.

One of the ways in which the FPF opposes patented protocols is by supporting and endorsing patent-free alternative protocols. This paper describes how WAP can be avoided by use of patent-free alternatives, and is therefore fully consistent with this mission. The purpose of this paper is to describe the development of the open Mobile Web Browsing industry. In particular, we will:

- Show how recent developments allow Mobile Web Browsing to be implemented based on an open industry model which avoids the WAP protocols entirely.
- Describe a new set of efficient protocols that can significantly increase the efficiency of Mobile Web Browsing.
- Show how the WAP protocols have now become entirely irrelevant, and discuss whether anything useful can be salvaged from them.
- Invite others to participate in the development of the open Mobile Web Browsing industry.

Acknowledgments

We gratefully acknowledge the assistance of the following persons in the preparation and review of this document: Pinneke Tjandana.

Conflict of Interest Disclosure

The authors of this article were also the initial developers of LEAP, and therefore have a vested interest in the success of LEAP over WAP.

However, we are also active participants in the Free Protocols Foundation (FPF), under whose auspices this article is being written. As participants in the FPF, we are fully committed to its patent-free principles. As noted above, part of the FPF mandate is to provide support for patent-free alternatives to patented protocols such as WAP. It is in the spirit of this mandate that this article is being written.

12.2 Mobile Web Browsing: Past, Present and Future

The basic problem that WAP purports to address is very real: that of providing website access from a cell phone. (More generally, the central web browsing problem is that of providing website access from miniaturized devices in general, including PDAs, pagers, etc. WAP is heavily oriented towards the cell phone in particular; however the required industry solution must address miniaturized devices in general.) A particular website may be very full-featured, including rich content which cannot be displayed on the limited cell phone display. In order for the phone to provide meaningful presentation of this website, an appropriate subset of the website content must be extracted and downloaded to the phone.

12.2.1 The Past: WAP

There is nothing bogus about this problem – only about the way WAP has gone about solving it. A key architectural component of the WAP solution is the **WAP Gateway**, which stands between the cell phone and the website, and which actively participates in the data extraction/download process. The catastrophic problem with this is that it totally violates the Internet End-to-End principle – the gateway is now interposed as an active authority between the client and the website.

Clearly, the WAP Gateway exists for business reasons, not engineering ones. Control of the gateway, together with control of the cell phone WAP software which can preferentially direct end-users to one gateway rather than another, creates enormous business opportunities for the gateway operator. This is the fundamental *raison d'etre* of the WAP Gateway; and everthing else in the WAP model, including the protocols themselves, falls secondary to this.

In other words, the entire WAP construct is an attempt by the wireless network operators and phone manufacturers to hijack the Mobile Web Browsing industry. If there was ever an example of the business dog wagging the engineering tail, this surely is it.

The basic motivation behind the WAP Gateway is aptly summed up in the following e-mail, posted to the IETF public mailing list by Phil Karn, an engineer at Qualcomm (itself a longtime WAP Forum member), in response to *The WAP Trap*. Material omitted from Mr. Karn's e-mail is indicated by ellipses; otherwise, he is quoted verbatim:

```
From: Phil Karn <karn@qualcomm.com>
To: public@MOHSEN.BANAN.1.BYNAME.NET
CC: ietf@ietf.org, karn@qualcomm.com
Subject: Re: WAP Is A Trap -- Reject WAP
Date: Tue, 20 Jun 2000 12:36:47 -0700 (PDT)
Message-Id: <200006201936.MAA26742@servo.qualcomm.com>

... I also scratched my head when WAP came out. It just didn't make any
technical sense. I see I'm not the only one; bravo for writing such a
good critique.

One thing missing from most block diagrams of WAP is the chute on the
bottom of the carrier's WAP gateway pouring out money. It's safe to
say that this chute is WAP's primary reason for existence.

... The Internet end-to-end model will once again prevail, putting the
cellular service providers back into their proper place as providers
of packet pipes, nothing more. And life will be good again. :-)
```

However, the wireless industry has now created the components required to solve the central web browsing problem the right way – and without WAP. In particular, two major developments have now rendered WAP completely irrelevant:

1. The XHTML[**?**] protocol from W3C
2. The LEAP family of protocols from the LEAP Forum

12.2.2 The Present: XHTML

Figure 12.1 shows the protocol stacks for web browsing under three different implementations: the WAP architecture, an architecture based on XHTML, and an architecture based on XHTML and LEAP.

XHTML is a markup language from the World Wide Web Consortium (W3C) that allows an appropriate subset of web page content to be provided to a requesting device, depending on the device's display capabilities. XHTML thus provides an immediate solution to the central problem of website access from a limited-capability device such as a cell phone. As shown in Figure 12.1(b), XHTML can be used in combination with HTTP [**?**] and TCP [**?**] to provide a complete web browsing solution which bypasses the WAP Gateway and avoids the WAP protocols entirely.

Furthermore, XHTML is an open, Internet-mainstream protocol, and conforms fully to the Internet End-to-End principle. The model of Figure 12.1(b) therefore provides Mobile Web Browsing the right way – i.e. based on a truly open industry model.

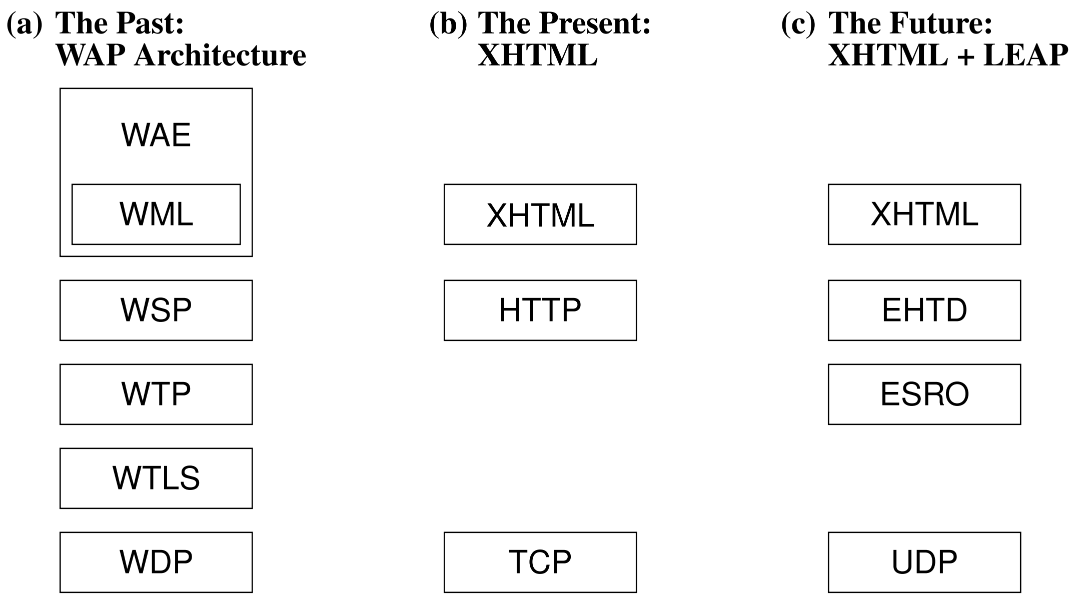


Figure 12.1: Mobile Web Browsing: Past, Present and Future

(Note: The model of Figure 12.1(b) is essentially the basis of the popular Japanese I-Mode system.)

For complete details about XHTML visit `http://www.w3.org/MarkUp/`. For information about W3C visit their website at `http://www.w3.org/`.

A disadvantage to the implementation of Figure 12.1(b) is that it requires the use of HTTP and TCP, which have serious inefficiency characteristics for the limited-size data transfer implied by miniaturized handheld devices. In the following sections we will see how the LEAP protocols can provide the required efficiency.

12.2.3 The Importance of Efficiency

The implementation of Internet applications such as web browsing in the wireless arena places a very high premium on the efficiency of data transfer. Wide area wireless networks demand bandwidth efficiency; miniaturized devices demand power efficiency; and the end user demands reliability and minimum latency. The underlying wireless protocols must provide all the required efficiencies.

The claim is sometimes made that the need for efficiency in the wireless arena is a temporary one – that advances in wireless engineering technology (such as third generation (3G) systems and 802.11b[**?**]) will eliminate existing bandwidth limitations, obviating the need for efficient protocols. And indeed some high-speed networks have been implemented which demonstrate the capabilities of next-generation technologies.

However, thus far such implementations have been limited in scope. They have been limited to a relatively small coverage area, or they have been limited in terms of their support for active mobility – i.e. they support only static wireless devices. But the efficiency demands of wireless networks with very large coverage areas (i.e. nationwide or worldwide), and which support active, in-motion mobility, are very much greater. And despite future advances in network speed, efficiency will remain a desirable goal as long as the capacity of wide-area wireless networks remains finite.

For these reasons, efficiency will remain a crucial aspect of wireless network usage for some time to come.

12.2.4 The Future: XHTML + LEAP

Because of the very limited display capabilities of miniature handheld devices, Mobile Web Browsing of necessity involves the transfer of relatively small amounts of data. In other words, Mobile Web Browsing is an inherently limited data-size activity.

As connection-oriented protocols, HTTP and TCP are most efficient for large data transfers; however they provide very poor efficiency for short data transfers. This means that HTTP and TCP are sub-optimal protocols for Mobile Web Browsing, and the scenario of Figure 12.1(b), though open and WAP-free, is a highly inefficient implementation.

Therefore an appropriate set of protocols are required to provide the necessary efficiency; and we are proposing the LEAP protocols as candidates for this role.

The Lightweight & Efficient Application Protocols (LEAP) is a family of high-performance, efficient protocols that are ideal for mobile and wireless applications. In sharp contrast to WAP, the LEAP protocols are truly open, RFC published, and patent-free; and declarations of the patent-free nature of the protocols have been made to the Free Protocols Foundation. For a comprehensive description of the LEAP protocols see *The LEAP Manifesto*, available on-line at `http://www.LeapForum.org/leap/`.

In addition, open-source implementations of the LEAP protocols are freely available through the **MailMeAnywhere** open-source software distribution center; for details visit the MailMeAnywhere website at `http://www.MailMeAnywhere.org`.

The initial focus of the LEAP protocols was on efficient Mobile Messaging, and the first two members of the LEAP family, EMSD (Efficient Mail Submission and Delivery; RFC-2524 [**?**]) and ESRO (Efficient Short Remote Operations; RFC-2188 [**?**]), were designed for this purpose. These protocols are now complete and in place, and a complete framework for the development of the open Mobile Messaging industry now exists. Given that, the focus of the LEAP Forum can move on to the next challenge: efficient web browsing.

Two members of the LEAP family of protocols are relevant to the web browsing application: ESRO and EHTD. ESRO provides reliable connectionless transport services which can be used for a variety of applications. For complete details on ESRO see the Manifesto article *ESRO: A Foundation for the Development of Efficient Protocols*, or visit the ESRO website at http://www.esro.org. EHTD (Efficient Hyper Text Delivery) is a hypertext transfer protocol which is optimized for the efficient transfer of short markup pages.

All the LEAP protocols are designed with a major emphasis on efficiency, and ESRO and EHTD together bring these efficiency benefits to the web browsing application. For short data transfers, EHTD is significantly more efficient than HTML, while ESRO is significantly more efficient than TCP – for example, TCP requires a minumum of 5 packets per transaction, whereas ESRO requires 2 or 3. For a detailed analysis of the efficiency of the LEAP protocols, see the Manifesto article *Efficiency of EMSD* [**?**]. That article analyses the efficiency of EMSD and ESRO specifically; however similar efficiency results can be expected in the case of EHTD and ESRO. In particular, ESRO and EHTD are highly efficient for the transfer of limited-size data, and are therefore ideal for the Mobile Web Browsing application.

Figure 12.1(c) shows how these protocols may be used in combination with XHTML and UDP [**?**] to provide a Mobile Web Browsing implementation that is completely open, WAP-free **and** efficient.

Note that the implementations of Figure 12.1(b) and Figure 12.1(c) are not mutually exclusive, but rather may be considered to be complementary. The connectionless protocol stack of Figure 12.1(c) is highly optimized for the short data transfers inherent to Mobile Web Browsing; whereas the connection-oriented stack of Figure 12.1(b) may be used for large data transfers whenever necessary.

12.2.5 Invitation to Participate

ESRO is a complete, RFC-published protocol, for which open-source software implementations are ready and available for immediate deployment. The EHTD protocol, however, is still in its early stages of development. Those who wish to participate in the development of EHTD are invited to do so, and may do so via the LEAP Forum website at `http://www.LeapForum.org`.

The experience gained in the development of the WAP protocols can be of great assistance in the development of EHTD. In particular, the WAP specifications include various technical design errors, from which important lessons can be learned. In this regard the engineers who took part in the design of WAP, or who otherwise have a technical understanding of WAP, represent a particularly valuable resource. Their participation is encouraged and welcomed.

12.3 WAP: A Salvage Operation

The WAP Forum has responded to the availability of XHTML by announcing WAP 2.0, which provides support for both XHTML and WML in the WAP protocol stack. This diminishes, but does not eliminate, the presence of the WAP Gateway in the WAP model. In addition, the in-place WAP 1.x architecture can claim to provide significant efficiency advantages over the connection-oriented stack of Figure 12.1(b).

However, all the other problems with WAP, detailed exhaustively in *The WAP Trap* and elsewhere, remain. Given the availability of truly open, Internet-mainstream alternatives, there is little remaining role for either the WAP specifications or the WAP Forum. At this point, WAP has become a salvage operation. There are three aspects to this salvage: engineering, business, and psychological.

12.3.1 Engineering Salvage: Scrapping WAP Layer by Layer

Before throwing WAP out completely, it behooves us to examine the specifications to determine whether there is anything worthwhile that can be salvaged for incorporation or usage in the open industry models.

The general WAP architecture is shown in Figure 12.2. This figure is taken directly from the WAP specifications, and all nomenclature, acronyms etc. throughout this section are those of the WAP model. Starting from the bottom of the figure:

- **WDP (Wireless Datagram Protocol).** The purpose of WDP is to accommodate non-IP networks. However, the convergence of wireless networks on IP at Layer 3 is now a technological reality. Most modern networks already support native IP, and IP will eventually become standard on all wireless networks. There is therefore no need for a protocol designed to accommodate non-IP networks.

  Since IP can be assumed to be present at Layer 3, UDP is entirely adequate at Layer 4. Therefore WDP is not needed at all, and can be scrapped completely.

- **WCMP (Wireless Control Message Protocol).** The purpose of WCMP is also to accommodate non-IP networks, which, as described above, is unnecessary. Assuming IP at Layer 3, the functionality of WCMP is adequately provided by ICMP. Therefore WCMP is not needed at all, and can be scrapped completely.

- **WTLS (Wireless Transport Layer Security).** The purpose of WTLS is to provide security functionality. However, a number of major security problems have been identified in WTLS, including vulnerability to datagram truncation attack, message forgery attack, and a key-search shortcut for some exportable keys. For a detailed description of the WTLS security problems, see Saarinen's paper *Attacks against the WAP WTLS Protocol* [**?**].

  Nevertheless, there may remain some worthwhile elements in WTLS. If the WAP Forum were to bring WTLS into conformity with the Internet mainstream by making patent-free declarations for it, publishing it as an RFC, and subjecting it to open review and maintenance procedures, then it may be worth examining for salvageable components.

- **WTP (Wireless Transaction Protocol).** WTP serves a genuine purpose; however, equivalent functionality to WTP is provided by ESRO. In addition ESRO predates WTP, is truly open and patent-free, is RFC published,

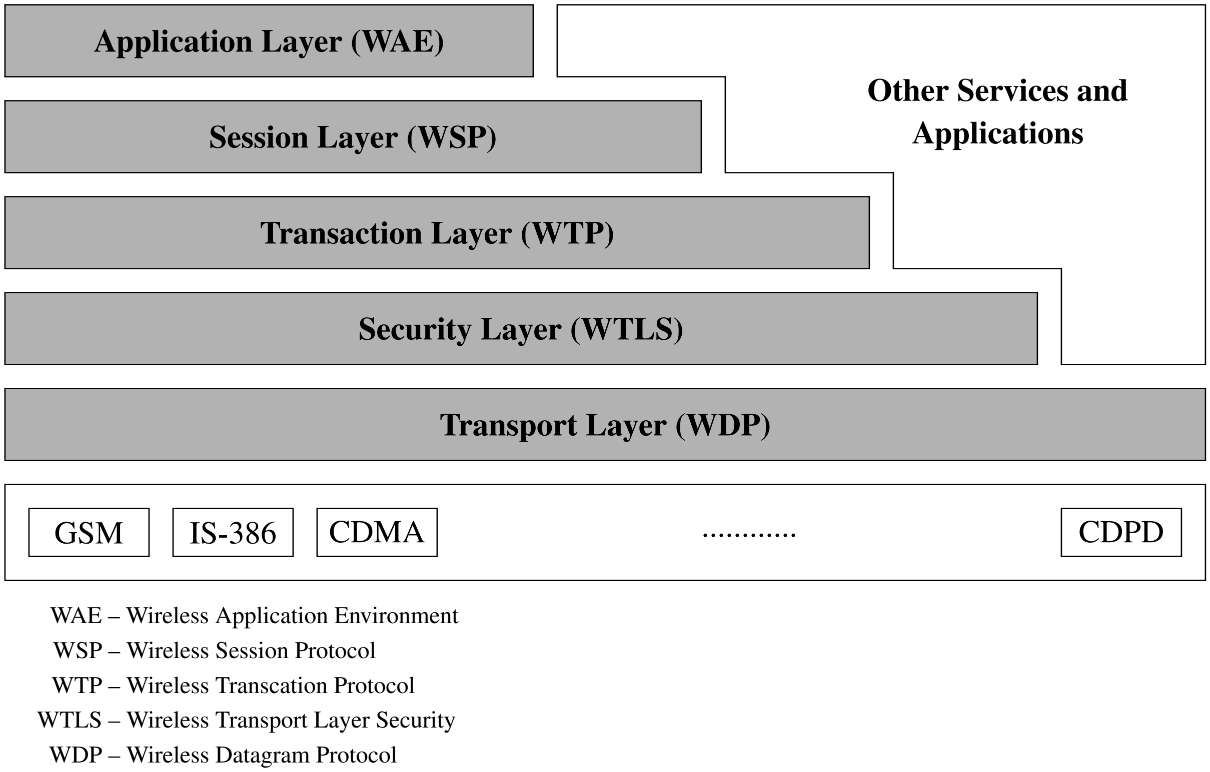


Figure 12.2: WAP Architecture

and otherwise conforms to the Internet mainstream. Therefore WTP is not needed at all, and can be scrapped completely.

- **WSP (Wireless Session Protocol).** WSP provides a binary form of HTTP. Therefore there may be components of WSP that can be used to facilitate the development of EHTD.

- **WML (Wireless Markup Language).** In the WAP model, WML is part of a broader specification called WAE (Wireless Application Environment). The functionality of WML is entirely provided by XHTML, which therefore renders WML irrelevant. WML is no longer required at all, and can be scrapped completely.

Thus every component of the WAP protocol stack is either functionally unnecessary, made irrelevant by an open alternative, or misdesigned; with the possible exceptions of WSP and WTLS, which may have some salvage purpose.

Our analysis of the WAP stack is supported by various other studies which come to conclusions consistent with the above. A good starting point is the article *W* Effect Considered Harmful* [**?**], in which author Rohit Khare presents a detailed analysis of WAP, and demonstrates its shortcomings and ultimate non-viability.

12.3.2 Business Salvage: Cutting Financial Losses

A huge amount of money has been sunk into the WAP fiasco – a large number of wireless network operators placed their bets on WAP, and invested heavily. And the WAP infrastructure is now complete; all the pieces are built and in place. The problem is that it falls disastrously short of its expectations; and as a result few people need it, few want it, and few are using it [**?**]. Apart from empty hype and broken promises, the WAP Forum has little to show for its massive investment.

Under circumstances like this, people may find it difficult to halt the investment Juggernaut. WAP has a huge amount of mass and momentum – it has the mass of its enormous investment costs, and the momentum of its own hype machine. The WAP Forum members may make the mistake of believing that this investment is still worth something. They may make the mistake of believing their own hype.

But WAP is doomed, and its investment costs are now sunk costs. The only thing for the investors to do now is pull the plug on WAP and cut their losses. Continued investment in WAP represents the throwing of good money after bad, and will only result in greater bottom-line losses at the end of the day. The sooner WAP is recognized as a costly failure, the better.

12.3.3 Psychological Salvage: Saving Face

Just about everybody joined the WAP Forum. The WAP Forum membership list is indeed impressive, including virtually every major player in the wireless and telecommunications industry. However, as we now know, this is not a meaningful endorsement of WAP. Rather, it is a testament to herd mentality and bet-hedging.

The arrival of XHTML and LEAP on the scene means that WAP is finished, and the WAP Forum has no significant role to play in the development of the wireless Internet industry. From this point on, the important work will no longer be taking place within closed forums such as WAP.

Given all of this, it is clear that the WAP Forum has little reason for continued existence, other than as a lame-duck organization with responsibilities that do not extend beyond face-saving activities for its members.

12.4 In Pursuit of Integrity

Much has happened in the 17 months since we first published *The WAP Trap*. The Internet bubble has burst catastrophically, causing the Nasdaq Composite Index to collapse from its peak of 5130 in March 2000 to around 1700 today – a staggering 67% loss of market capitalization.

And the WAP bubble has also burst. The fortunes of WAP are perhaps best represented by Openwave Systems, Inc. (formerly Phone.com, Inc.), one of the principal inventors and architects of WAP. The stock price of this company reached a peak of $208 in March 2000; at the time of writing in September 2001 it is trading at around $15 – a loss of 93%. Other WAP-related companies have experienced similar losses.

Meanwhile, the consumer has yet to see anything close to the promised ease and convenience of cell phone Internet access; and we are not aware of even a single company that has made significant profits from sales of WAP services.

12.4.1 The WAP Hype Machine Fraud

WAP has been a colossal failure in financial terms. Its usage has not and cannot recoup its investment costs. Nevertheless, WAP has created fortunes for a privileged few.

The WAP business model is based on the traditional supply chain model, in which the financial and other needs of all potential gatekeepers are addressed throughout the supply chain. The creation of this supply chain has required the construction of a major infrastructure. Though this supply chain model cannot and will not work as intended, its construction has presented enormous profit-making opportunities for those in the right position.

These profits have derived from two major sources. The smaller of these consists of the profits associated with building the WAP infrastructure itself; in particular the huge development contracts that have been awarded, together with sales of WAP gateways and other equipment.

But it is the larger source that represents the truly spectacular opportunity. This opportunity has been based, not on building the WAP infrastructure, but on the fairy-tale promises and expectations that have been created alongside it. The enormous amount of hype surrounding WAP led to huge increases in stock prices and company valuations across the entire WAP industry – nowhere better represented than in the valuation of Phone.com itself.

Various WAP promoters were also investors and stockholders in key WAP companies. These investors/promoters participated actively and collectively in the hyping of WAP, drove valuations up to levels far beyond what was realistic or supportable, then sold their WAP-related stock to the public at vastly inflated prices. One could be forgiven for wondering whether the activities of the WAP promoters were intentionally directed towards this happy outcome. As the disappointing reality of WAP inevitably became clear, virtually all these inflated stock prices collapsed to less than 10% of their WAP-bubble peak, making fortunes for the investors, while leaving the public holding the empty WAP bag.

This type of activity is commonly referred to as a "pump-and-dump" scheme – an ugly phrase to refer to an ugly operation: the deliberate over-hyping of a stock with the intention of artificially inflating its price, then dumping it on an unsuspecting public. From the perspective of the unfortunate losers, the collective activities of the WAP insiders must be hard to distinguish from a pump-and-dump operation on a grand, industry-wide scale.

The WAP bubble was part of the more general Internet bubble, which represented the aggregate effects of a multitude of contributary bubbles similar to WAP. The WAP bubble was thus both a consequence of, and a cause of, the Internet bubble. To the extent that WAP was a consequence, the WAP promoters may shirk their responsibility for the WAP bubble. But to the extent that WAP contributed, they must then accept responsibility for the broader Internet bubble.

We have no objection to those who make fortunes on the basis of something real. Authentic entrepreneurs make fortunes by building companies which provide something of value to the consumer, and which create enduring value for their stockholders. Such people fully deserve the wealth created by their ingenuity, commitment and hard work.

Nor do we object to profitable stock trading in which no misrepresentation takes place. The stock prices of many companies were swept up and down along with the general Internet bubble; but in most cases this took place without gratuitous hyping by insiders. Those who sold near the peak made money at the expense of those who bought; but those are the breaks in the high-tech industry, and these are the risks that investors must accept.

But neither of these considerations applies to WAP. In the case of WAP little of value has been provided to the disappointed consumer, the value of company equity has been fleeting, and a minority of people have been greatly enriched at the expense of a duped majority. The WAP fortunes have been made by selling WAP-related stock at inflated prices, *not* by delivering WAP services to satisfied customers.

Furthermore, the WAP hype campaign continued, and still continues today, *despite the fact that actual WAP usage remains dismal, and no one has ever made significant profits on the basis of WAP services*. Given these facts, we find it scarcely conceivable that the WAP insiders were unaware that WAP was being hyped far beyond its reality, that stock prices were being driven to levels far beyond their sustainable value, and that they would inevitably collapse.

We are making no suggestion here that actual, prosecutable criminal fraud took place. But there can still be breach of trust, even though no law may have been broken. When we consider that the WAP model includes a gateway whose primary purpose is to generate revenue for its operator; when we consider that WAP is patented; that WAP is a shoddy engineering construction; that WAP is the pseudo-open creation of a pseudo-open forum, then we have to wonder if everything is entirely above board.

In our judgement, the activities of the WAP investors/promoters amount to fraud in all but the letter of the law. Our readers may come to their own conclusions.

12.4.2 Protocol Integrity

Underhanded practices are a fact of life in the business world. But when such practices involve the creation of a large-scale engineering construct, and when they are based on the exploitation of vital industry protocols, this degrades the integrity of the engineering profession.

The engineering profession traditionally carries a responsibility to protect the safety and welfare of the public. An industry protocol is an engineering construct, held in public trust by the engineering community. It is the responsibility of the engineering community to defend this trust against exploitation by narrow business self-interests.

There are three fundamental principles for maintaining the integrity of public protocols [**?**]. These are:

- Patent-freedom
- Unrestricted access, permanence and stability of the published specifications (e.g. RFC publication)
- Maintenance by truly open organizations

Each of these provides a vital assurance of protocol integrity. Patent-freedom ensures that a patent-holder cannot subvert free-market competition among products and services based on the protocol. RFC publication ensures that the protocol is freely available to anyone who wishes to use it. And maintenance by open organizations ensures that development of the protocol takes place by technical engineering consensus, rather than business self-interest.

This trilogy of principles represents the most basic guarantees of the integrity of a protocol. If any one of these things is missing, then this means that some attempt is being made to control or limit access to the protocol. In the case of a public protocol, there is no valid reason for doing this.

The creation of the WAP specifications has violated every one of these principles. The use of patents and other access-control mechanisms has been a traditional way of life in the highly business-oriented, oligarchic telecommunications industry. But the Internet industry is not like that. Openness and freedom from authority lie at the heart of the Internet, and in no small measure account for its extraordinary vitality and success. Patents and other business-oriented control devices have no place in this industry. Though WAP may try to pass itself off as an “open Internet protocol,” its roots in the telecommunications industry are plainly evident.

In *The WAP Trap* we challenged the WAP Forum either to provide valid reasons for their violation of the above principles, or to bring the WAP specifications into line with them. Seventeen months later, neither of these things has happened, and our challenge remains unanswered.

We now repeat our challenge. We challenge the WAP Forum to abandon their closed, members-only model of operation, make patent-free declarations regarding the WAP protocols, publish them as Internet RFCs, and subject them to genuine public review and maintenance procedures. By taking these steps, the WAP Forum will allow the possibility of what remains of WAP being incorporated into the mainstream Internet development model.

12.4.3 Engineering Integrity

When first published in April 2000, *The WAP Trap* was well ahead of its time. At that time it represented a distinctly minority viewpoint, and seemed radical and extreme to many. Today it seems much less so.

The same may be true of this article. To the casual observer, the WAP Forum may appear to be a healthy organism, engaged in creating something important and worthwhile. WAP has not yet been fully discredited, and it may not for some time. Meanwhile, the naive or the inexperienced may find themselves impressed by the sheer scale of financial investment and engineering effort that has gone into WAP. Such observers may find themselves puzzled by, and skeptical of, our rhetoric. It may be hard to accept that something so big can be so fallacious; nevertheless, that is the fact.

In this paper we are lobbying for a Mobile Web Browsing based on a truly open industry model. By definition, WAP in its present form can play no role in such a model. Not all the things we are lobbying for will take place, and the things that do may not take place as soon as we would like. The LEAP protocols we are proposing may become part of this model, or they may not. The WAP salvage operation we are suggesting may contribute to this model, or it may not.

But the eventual outcome is clear. WAP is non-viable, and sooner or later the rest of the wireless industry will come to this realization. And at some point it will be replaced by a truly open solution.

In the meantime, we urge those engineers who have an interest in the ethics of their profession to distance themselves from WAP, because it is specious. Given a choice between WAP and something else, we encourage the engineering men and women of the wireless industry to invest their precious talents in something that has both business and engineering integrity.

Chapter 13

Operation Whiteberry

13.1 Introduction

The wireless data communications medium of today is struggling to define itself. It is clear that this medium is enormously important; it is also enormously complex, with major technical, societal and business consequences that no one fully understands.

Of the many questions that this new medium raises, two of the most basic are: (1) What is the key starting-point application, and (2) What is the best way to implement this application? In other words, what wireless application is of the most immediate and compelling value, and what is the most appropriate model for delivering this application?

We believe that the answers to these questions have now become clear. The right entry-point application is Mobile Messaging, and the right implementation model is an entirely open paradigm, based on open and free protocols.

Mobile Messaging is in complete harmony with the wireless medium, provides users with an incontrovertible value proposition, and has already achieved widespread market acceptance.

And a completely open paradigm is the right model for all large-scale communications systems. An open industry model provides the greatest benefit to the end user and the industry at large, by allowing free market entry and competition at any point within the Mobile Messaging solution domain. This in turn results in greatly increased business opportunities, more and better solutions for the end user, and unrestricted industry growth.

Unfortunately, these two key ideas are not well understood within the wireless industry. Numerous wireless initiatives are under way, but in all cases these are misdirected either because they are pursuing the wrong application, or because they are pursuing the right application but based on the wrong model.

As a result of this, the wireless industry of today is in a state of chaos. A large segment of the industry, most notably the WAP Forum, has mischaracterized the major entry-point application, and is pursuing wireless web browsing and other second-string applications.

Meanwhile, another industry segment is investing in the right application, but based on the wrong model. Mobile Messaging is the right application, but this industry segment is characterized by the existence of various closed, proprietary solutions such as BlackBerry and ReFLEX. The components of these competing systems do not interoperate, and they cannot build on each others assets. The result of this is the fragmentation of the Mobile Messaging market into a number of isolated islands of consumers, each limited to a particular solution and unable to take advantage of the assets of any other island.

The remedy to all this exists, in the form of a set of protocols called the **Lightweight & Efficient Application Protocols**, or **LEAP**. LEAP is a set of high-performance, efficient protocols that are ideal for mobile and wireless applications. In this article we describe how equivalent Mobile Messaging functionality to the existing closed systems can be provided, in the form of an entirely open solution based on the LEAP protocols. Our goal is to use the inherent power of truly open and free protocols to displace these closed solutions. We refer to the execution of this goal as **Operation WhiteBerry**.

WhiteBerry provides true Mobile Messaging, based on a set of open protocols—and therefore represents the right application *and* the right model. All the pieces required to implement WhiteBerry exist and are in place, and the openness of the WhiteBerry solution ensures that it will displace all existing closed solutions.

13.1.1 The Problem

The wireless industry of today is characterized by systems which are closed, restricted, and which violate the Internet End-to-End principle. Two of these are of particular relevance:

1. The Wireless Application Protocol, or WAP
2. The BlackBerry Mobile Messaging system

WAP is discussed in detail in the article entitled *The WAP Trap* [**?**], while BlackBerry is discussed in the present article.

Our analyses of WAP and BlackBerry are very different. As we argue in *The WAP Trap*, WAP is a bogus marketing construct that can bring nothing but harm to the industry and the consumer. WAP claims to be an open specification, whereas in fact it is anything but—on the contrary, it is booby-trapped with patents. Furthermore, the basic premise of WAP (namely, that of providing Web browsing capability on a cell phone) is inherently limited and impractical, and is the wrong starting point application.

Mobile Messaging systems such as BlackBerry, on the other hand, bring unquestionable value to the user, and these systems have experienced widespread usage as a result. Without exception, however, all existing Mobile Messaging system are closed, and the emerging competitive scenario is that of a struggle among these closed systems.

Thus, while part of the wireless industry now appears to understand the enormous importance of the Mobile Messaging application, it does not appear to understand the importance of the open model.

13.1.2 The Solution

The solution to this closed-system donnybrook is not another set of products and services. Rather, the solution is the industry-wide adoption of the right set of open protocols as the basis for interoperability and cooperation.

Throughout this article, we will take the BlackBerry system from RIM as a particular example of a closed Mobile Messaging system. In common with other messaging systems, BlackBerry brings unquestionable value to the user. But despite its end user value, BlackBerry remains a closed, single-vendor system, based on a set of proprietary protocols. The proprietary nature of the underlying protocols prevents competition on the basis of any of the constituent components of the BlackBerry system.

And this is something which is true in general: *closed systems diminish competitive opportunity*. It is, of course, entirely possible for a closed system to encounter competition in the form of another closed system. However, the market entry barriers and competitive challenges are far greater for a competing *system* than for a competing *system component*—such as a wireless modem, for example.

Therefore closed systems act to inhibit competition both at system level and component level. This inhibition of competition is all to the good of the company that owns the closed system—which is the fundamental *raison d'etre* for all closed systems, of course. However, this same inhibition of competition acts greatly to the detriment of the industry at large and the consumer.

In contrast to this focus on narrow business self-interest, we are committed to the following principles:

- We are advocates of patent-free protocols.
- We are advocates of truly open systems, based on free protocols and open-source software.
- We are committed to the **Internet End-to-End model**. In this model the Internet serves merely as a passive communication pipe, allowing direct and unrestricted communication between client and server.

These principles greatly benefit the industry and the consumer, by promoting innovation, competition, and freedom of choice.

In this article we describe how the equivalent functionality to BlackBerry can be implemented in the form of a completely open system, based on existing protocols and technologies. We refer to this open equivalent to BlackBerry as the **WhiteBerry** solution. In contrast to BlackBerry, WhiteBerry is a multi-vendor solution, allowing market entry and competition at any point within the Mobile Messaging technology chain. The result is greatly increased business opportunities throughout the Mobile Messaging industry, increased competition, and better solutions for the end user.

Though we are presenting WhiteBerry as an open alternative to BlackBerry, there is nothing unique about BlackBerry in this regard. WhiteBerry represents an open alternative to *any* closed Mobile Messaging system, including others besides BlackBerry such as *Mail on the Run!* from River Run Software Group, and ReFLEX from Motorola. We have chosen BlackBerry as the subject of this article because it is in widespread use, and at this point appears to be the market leader. However, the same principles of openness versus closedness apply equally to any closed Mobile Messaging system.

13.1.3 Complete and Available

The entire WhiteBerry solution that we describe in this article is complete and available today. The key component of WhiteBerry is a set of mobile messaging protocols. These are complete, fully satisfy all necessary technical requirements, are truly open and patent-free, and have been published as RFCs. In addition, a comprehensive framework for the development of WhiteBerry implementations has been established. This framework consists of free, open-source software implementations of the protocols, an open public forum for the development and distribution of integration tools, an initial set of free Subscriber Services, and an initial end-to-end WhiteBerry implementation. Anyone can use these tools and facilities to create a fully functional WhiteBerry Mobile Messaging implementation immediately.

Furthermore, the WhiteBerry development framework is highly generalized, and goes beyond simply duplicating the limited capabilities of the BlackBerry system—it also allows the development of a more general set of wireless applications, including Instant Messaging and others. A discussion of some of the future possibilities of the WhiteBerry model is provided in Section 13.9.

We begin this article by discussing the general functional requirements for Mobile Messaging. We then describe how these requirements are provided by a typical closed system, using BlackBerry as an example. Next we provide a detailed description of the WhiteBerry model, and show how it provides equivalent functionality to BlackBerry, based on an open model. We then describe the WhiteBerry development framework, and discuss the security issues relating to Mobile Messaging.

The solution we propose is not just an academic, engineering exercise; it also has a well-founded business dimension. As will become clear later, there is a great deal of money to be made in Operation WhiteBerry.

13.1.4 Free Protocols Foundation Endorsement of Operation WhiteBerry

The Free Protocols Foundation (FPF) is a non-profit organization dedicated to the support of patent-free protocols and software. The FPF views software patents as being extremely harmful to the industry and the consumer, and part of the FPF mandate is to oppose such patents when they appear. For more information see the FPF website at `http://www.freeprotocols.org/`.

In April 2001 Research In Motion (RIM), the manufacturer of the BlackBerry system, was granted U.S. Patent # 6,219,694 on the basis of its BlackBerry technology, then promptly sued a competing company (Glenayre Electronics, Inc.) for infringement against this patent. While BlackBerry already limits competitive opportunity by virtue of being a closed system, the assertion of this patent is a giant step beyond that in terms of anti-competitive practices: it represents an attempt by RIM to eliminate all competition entirely.

The FPF regards this patent as fundamentally invalid, and RIM's infringement claim as an egregious example of patent law abuse. For a detailed statement on this patent see the FPF paper *Position Statement regarding the RIM Mobile E-Mail Patent Assertion* [**?**], available on-line at `http://www.freeprotocols.org/position-rim-6219694.`

The FPF is strenuously opposing this patent by means of a variety of actions; see the above position statement for details. One of the general strategies by which the FPF opposes patented software is by supporting the creation and development of patent-free alternatives. Since *Operation WhiteBerry* describes an open, patent-free alternative to the closed and patented BlackBerry system, it is directly aligned with this strategy.

Therefore, as part of its broad compaign to oppose the RIM patent, the Free Protocols Foundation endorses *Operation WhiteBerry* fully, and is pleased to re-publish this revision of the paper on its own website. *Operation WhiteBerry* was first published and remains available on the LEAP Forum website at
`http://www.LeapForum.org/operationWhiteberry/index.html.`

13.2 Mobile Messaging Requirements

The mobile user of today can choose from a variety of messaging technologies. At one extreme, the user may choose to carry a miniature paging device. At another extreme, he[1] may choose to carry a laptop device which he can use to dial in and retrieve e-mail messages. Each of these technologies provides a particular set of functions and capabilities to the mobile user.

Modern e-mail systems allow a comprehensive richness of message content, including To, From, Cc and Subject fields, and the ability to include attachments. Traditional paging devices, by contrast, typically allow very limited message content—in some cases no more than a 10-digit telephone number.

On the other hand, paging devices provide the user with a very high degree of portability, which cannot be matched by a laptop or even most palmtop devices. Furthermore, paging systems deliver messages directly to the recipient, who is then unilaterally alerted to their arrival by the paging device. By contrast, e-mail systems usually do not provide this direct delivery capability, but instead require an explicit retrieval action on the part of the recipient.

The mobile user of tomorrow will expect and demand the best that each of these technologies can offer. While on the move, mobile users need to have important or time-critical information delivered to them. This information may be more complex than a 10-digit telephone number. And it may be necessary to present this information to the user immediately, without requiring him to dial in for messages. Users must be able to accept messages at any time, at any place, and on a device that they can carry anywhere.

In order to satisfy these requirements, a Mobile Messaging system must have the following characteristics:

- Allow the **full information content** of modern e-mail. The limited message content of traditional paging devices is not adequate.

- Provide mobile messaging capability on an **unconscious carry** device—meaning that the device can readily be carried anywhere by the user, without moment-to-moment awareness of its presence. A palmtop is the maximum size form factor that can be considered to satisfy this requirement.

[1] To avoid the clumsy he/she construction, throughout this article we will assume a male protagonist. In other articles in this series we may assume the same or opposite gender.

- Support the **push** mode of message delivery—meaning that messages are delivered directly to the recipient, without requiring an explicit poll or retrieval action.

The required Mobile Messaging system must therefore combine the message richness of typical e-mail, with the unconscious carry and push delivery characteristics of typical paging devices. We will refer to a messaging system that satisfies these criteria as a **true Mobile Messaging** system.

Unconscious carry, and the push mode of delivery, are essential requirements of the mobile user. Showing up to a romantic dinner carrying a laptop is idiotic. Showing up with a pager in your pocket is not (though answering it may be). And if much later your now-wife is soon to give birth, you certainly do not want to have to dial into your Message Center to discover that she has gone into labor. Information like this must be presented to you immediately, and wherever you happen to be.

13.3 The BlackBerry Solution

Note:*This section contains technical descriptions of the BlackBerry system, as well as statistical and other data about the Mobile Messaging industry in general. At the time that this article was first published in February 2001, this information was correct to the best of our knowledge. However, at the time of current writing (November 2002), much of this information is now out of date. We intend to update this section in a subsequent revision; in the meantime the reader is asked to bear in mind that technical and statistical data throughout this section may no longer be correct.*

To illustrate the shortcomings of closed messaging solutions, we will take the case of BlackBerry as an example. While the description and analysis throughout this section applies specifically to the BlackBerry system, similar principles apply to all closed messaging systems.

BlackBerry is a Mobile Messaging solution from Research In Motion (RIM). It satisfies all the Mobile Messaging requirements specified above: it provides users with general e-mail capability; the end user devices satisfy the unconscious carry criterion; and it supports the push mode of operation—incoming e-mails are delivered directly to the handheld device, which then immediately advises the user of their arrival.

BlackBerry can be used as a stand-alone Mobile Messaging system; alternatively, it can be integrated with a user's existing home or office e-mail account. In this case it appears to the user as a wireless extension of his existing desktop, corporate, or ISP-based e-mail service. The user can freely access and manage the wireless mailbox, while the system maintains synchronization with the land-based mailbox.

In the following sections, we will describe the major characteristics of the BlackBerry system. For complete details, see the BlackBerry website at `http://www.blackberry.net/`.

It is worth noting that in technological terms, the BlackBerry messaging implementation represents nothing fundamentally novel. All the basic concepts, methods and processes represented in the implementation have been previously known and put into practice by others. What RIM has done that is genuinely new is to package those concepts and bring them to market as a consumer product.

13.3.1 How BlackBerry Works

The overall operation of the BlackBerry system is shown in Figure 13.1. All BlackBerry-specific components are shown shaded in this figure; all generic or third-party components are unshaded. (Note: Figure 13.1 and the following description represent *our understanding of the BlackBerry system at the time this article was first published in February 2001*. Furthermore, because the system is closed and technical information regarding its operation is unavailable, we cannot guarantee the accuracy of this description; nor is there any assurance that RIM will not change the operation of the system in the future. The following description is based on RIM's promotional information, supplemented with our own educated assumptions where necessary. For the latest and most reliable information, refer directly to RIM's own materials.

Note also that our purpose in providing this description is not to reverse engineer the BlackBerry system; rather, it is to illustrate the characteristics of closed Mobile Messaging systems in general. Minor inaccuracies in our description of BlackBerry are therefore not of any great importance.)

The BlackBerry user is provided with a wireless handheld device, shown on the left side of the figure. The user has a choice of either of two alternative device styles: a pager-style form factor, or a palmtop-style form factor. The device has an integral wireless modem, allowing communications over the BellSouth Intelligent Wireless Network.[2] The device is equipped with RIM's software implementation of their proprietary wireless-oriented protocol, allowing the device to communicate with the similarly equipped BlackBerry Message Center.

The mobile device is supported by the proprietary, RIM-operated BlackBerry Message Center, shown in the center of Figure 13.1. Any e-mails addressed directly to the mobile device are fielded by this Message Center using standard Internet protocols, then delivered to the mobile device using the proprietary RIM wireless protocols. The device modem remains on at all times to accept incoming messages, and the device notifies the user immediately whenever a new message arrives.

To send a message, the user composes the message then submits it to the BlackBerry Message Center via the RIM protocols. The Message Center then sends the message to its destination using standard Internet e-mail protocols.

More often, the user wishes his mobile messaging capability to function as a wireless extension of an existing e-mail account. Under one scenario, the user may choose to have the BlackBerry system function as an extension of his Microsoft Outlook desktop mail application. In this case, the user installs the BlackBerry Desktop Redirector software on his PC. This situation is shown at the top of Figure 13.1.

The Desktop Redirector software integrates with the Microsoft Outlook mail application, and allows selected e-mails to be forwarded to the wireless device on the basis of user-defined e-mail filters. Properly qualified e-mails are forwarded to the BlackBerry Message Center using standard e-mail protocols, which then delivers them to the mobile device using the RIM protocols.

When the user sends a message, the BlackBerry Message Center sends the message to its destination as usual, but in addition sends a notification to the BlackBerry Desktop Redirector so as to maintain mailbox synchronization between the handheld device and the desktop mail application. Similar synchronization is maintained for any other action the user might take, such as message forwarding, deletion, etc. The mailbox synchronization protocols required to accomplish this are also proprietary to RIM.

For integration with corporate e-mail systems, RIM provides the BlackBerry Enterprise Server, which integrates with the Microsoft Exchange Message Center, as shown on the right of Figure 13.1. The BlackBerry Enterprise Server functions in much the same way as the Desktop Redirector, except that mail redirection and synchronization take place at the corporate Message Center, rather than the user's desktop mail application. As before, qualifying e-mails are forwarded to the BlackBerry Message Center, then delivered to the mobile device.

The BlackBerry system can also be integrated with certain third-party Service Providers; at the time that this article was first written in February 2001 there were four of these: OneMain, RCN, Rogers AT&T, and PageNet Canada. The nature of the business relationship between RIM and these companies is unknown to us; our guess is that in each case RIM licenses the BlackBerry Enterprise Server to the Service Provider, which then integrates it with its in-house Message Center system. Regardless of the actual business arrangement the end result is the same: the Service Provider is then able to offer the BlackBerry system to its customers as a wireless extension of their regular e-mail account. This situation is shown in the lower right of Figure 13.1. The functional logistics in this scenario are essentially no different from integration with corporate Message Centers.

[2] Formerly known as the RAM network

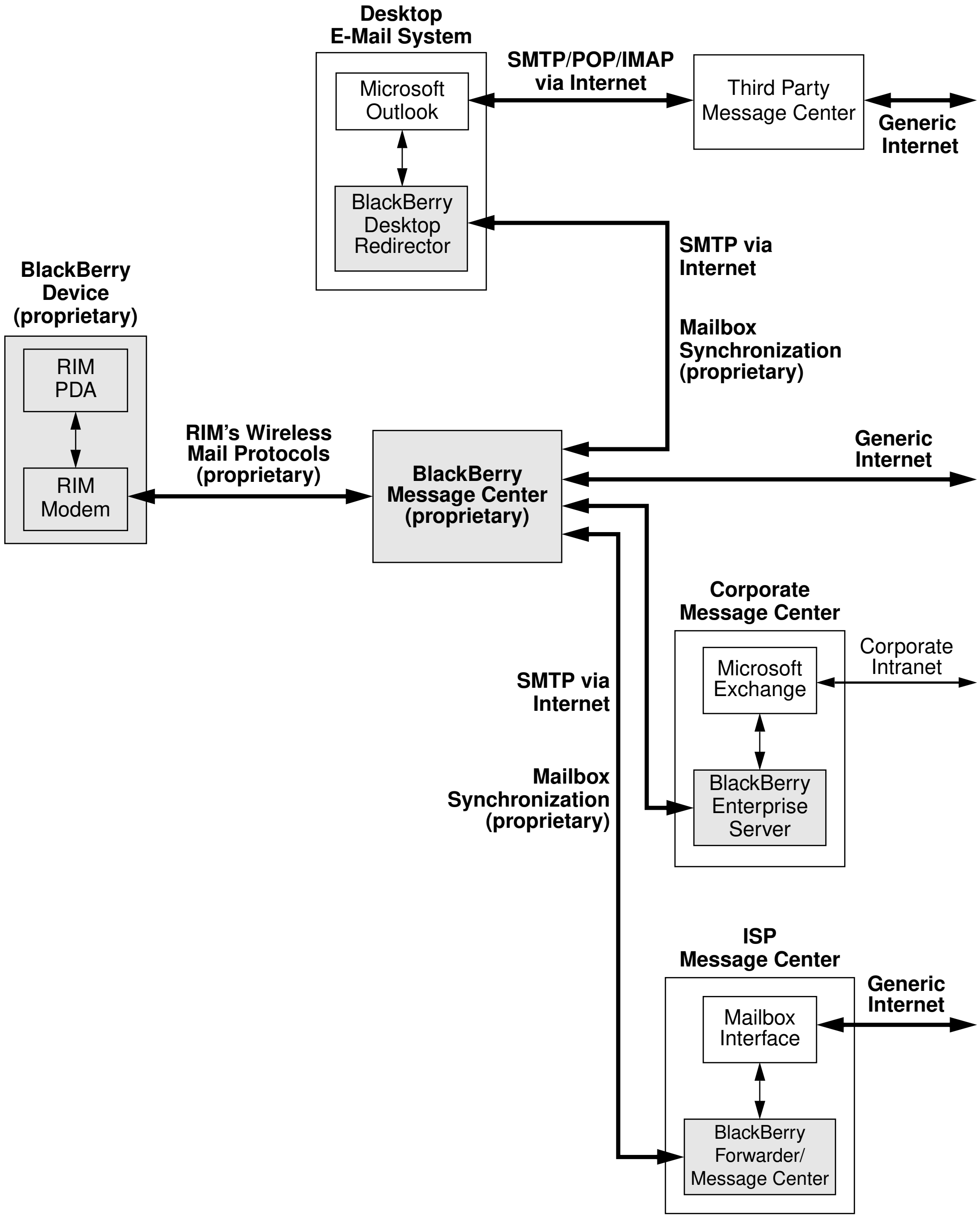


Figure 13.1: The BlackBerry Solution

13.3.2 BlackBerry: Mobile Messaging Confirmation

Mobile Messaging solutions such as BlackBerry have been well received by consumers. Users report a high level of customer satisfaction, and such systems have experienced steadily increasing adoption and usage. The basic reason for this is that these systems provide a clear value proposition to the user:

> Mobile Messaging systems provide *instant delivery of important or time-critical e-mail messages* to *mobile users*, using an *unconscious carry device*. The mobile device is always on, and is guaranteed to get you the information you need immediately, wherever you happen to be.

The market acceptance of systems like BlackBerry unequivocally confirms that true Mobile Messaging, as defined in Section 13.2, is the killer application in the wireless data communications arena. Unconscious carry, push mode, and full e-mail functionality: these are the key characteristics that have established Mobile Messaging as a viable commercial concept. And these are the same characteristics that will define the Mobile Messaging industry of the future.

13.3.3 BlackBerry: A Closed Solution

Despite its proven value proposition, BlackBerry remains a closed messaging solution. It is a proprietary package of integrated components and services, supplied by a single vendor.

The key component of the BlackBerry system consists of the protocols used for final-leg communication between the BlackBerry Message Centers and the mobile devices. However these protocols belong exclusively to RIM. They are heavily protected by patents, are not published or otherwise made available to potentially competing vendors, and are otherwise treated as trade secrets.

For this reason, BlackBerry consists of *precisely the components and services defined by RIM*. Because of the closed nature of the protocols, it is not possible for any other vendor to provide any alternative component of the BlackBerry solution. This is all very much by choice on the part of RIM, of course, who are executing a business model based on the classical "sustainable advantage."

An integrated package of components is required to bring Mobile Messaging capability to the end user. The various required components, and the way in which BlackBerry supplies these components, are summarized below:

- **Mobile device**. Only the two RIM-manufactured BlackBerry devices are available. No other device, such as a Palm or Windows CE device, is currently supported. (Of course, RIM could readily provide support for these and other devices by licensing its software for use on third-party devices, and we would not be surprised if at some point they do exactly that. Even if they did, however, these would merely be additional device incarnations of the same BlackBerry system. The system would still remain closed, because the underlying protocols would remain closed and proprietary to RIM.)
- **Wireless modem**. Only the RIM modem which is integral with the device is supplied. No other modem is supported.
- **Wireless network**. As of the time of initial writing in February 2001, the BlackBerry system is based exclusively on the BellSouth Intelligent Wireless Network (and its Canadian extension, the Rogers AT&T Wireless Network). No other wireless network, such as GSM/GPRS or CDPD [**?**], [**?**], is supported.
- **Message Center service**. Message Center service for final-leg communications with the mobile device is provided exclusively by the RIM-operated BlackBerry Message Center. No other Message Center service may be used for this purpose.
- **Protocols to tie these components together**. BlackBerry is based on RIM's proprietary protocols.

- **Where necessary, integration with the user's Desktop, Corporate, or ISP-based e-mail system**. In the case of desktop integration, BlackBerry provides support for the Microsoft Outlook mailbox application only; no other mailbox applications, such as Netscape Messenger, are supported. In the case of integration with corporate Message Center systems, BlackBerry provides support for Microsoft Exchange only; no other Message Center systems, such as sendmail [**?**] or QMail [**?**], are supported. In the case of integration with third-party ISPs, RIM currently has licensing agreements with only four: OneMain, RCN, Rogers AT&T, and PageNet Canada; no others are presently supported.

- **Integration of all these components into a packaged solution for the end user**. The BlackBerry system is integrated and packaged exclusively by RIM.

Thus from beginning to end, BlackBerry is defined and controlled by RIM.

A further consequence of the closed nature of the RIM protocols is that they have received no external engineering scrutiny or validation. They are known to function adequately in the environment and usage patterns defined by the BlackBerry solution, but no information is available regarding their technical merits—for example, no data is available by which to judge the performance, reliability or scalability of these protocols.

In addition, no detailed information is available regarding the BlackBerry security implementation. Security issues are discussed in greater detail in Section 13.6.

13.3.4 BlackBerry: Not All Things to All People

Most of the closed aspects of the BlackBerry system do not directly affect the experience of the end user. For example, the typical user neither knows nor cares that the system is limited to the BellSouth Intelligent Wireless Network.

However, the closed nature of system is immediately evident to the end user in the form of the BlackBerry devices. While most users are extremely satisfied with the Mobile Messaging functionality of their device, they may be less than completely satisfied with its other features.

A device manufacturer may be able to do one thing very well, but it is extremely difficult to do everything well. BlackBerry provides Mobile Messaging functionality very well. The BlackBerry devices also provide the other standard PDA applications: calendar, address book, memopad, task list etc.; however, the BlackBerry implementation of these applications falls well short of others in the marketplace. And even apart from the merits of one device over another, there is the simple matter of taste and familiarity; people grow used to a particular device or application, and have a reluctance to change to something different.

As a general-purpose PDA, BlackBerry is inferior in many ways to other devices, most notably the popular Palm family. For this reason users find themselves carrying around two devices: their preferred PDA, plus BlackBerry as a specialized paging/messaging device. The solution to this inconvenience is abundantly obvious to the puzzled end user, who asks, "Why must I carry both of these? Why does my preferred PDA not do what BlackBerry does?"

Why not indeed. Clearly, what is required is for the Mobile Messaging capability of BlackBerry to be available on *any* mobile device—and WhiteBerry makes this possible.

13.3.5 Strategic Myopia: More Closed Solutions

Because the value proposition has now become so clear, various companies are eager to jump on the Mobile Messaging bandwagon. For those companies that already have existing relevant assets, this is not difficult to do. A good example is Palm, which has a tremendously powerful asset in the form of its installed base of approximately 11 million PDAs[3]. These can be wireless enabled with a variety of existing modems, and can therefore readily form the basis for a Mobile Messaging solution. In addition to its very positive effect on their PDA sales, this would also provide Palm

[3] As of February 2001

with immediate entry into the Subscriber Services business arena, a highly desirable and natural extension of their existing business model.

Palm thus has every reason to launch a Mobile Messaging solution: they have a vital asset, it is easy to do, and the business advantages are enormous. Evidently none of this is lost on Palm, who recently announced their intention to market wireless-enabled versions of their PDAs in the second half of 2001 [**?**]. Also, the opportunity to capitalize on the shortcomings of the BlackBerry devices is not lost on Palm CEO Carl Yankowski, who says:

> "We can do everything that (Research In Motion) does not"

For reasons discussed in Section 13.7.4, there is a great temptation for businesses to implement closed and proprietary solutions. Thus far, every player to enter the Mobile Messaging arena has succumbed to this temptation. Palm has not yet provided information on whether it will be implementing a closed or an open messaging solution, or what the underlying protocols will be. However, we do not expect Palm to be any exception.[4]

Other companies have been faced with a similarly clear-cut imperative to enter the Mobile Messaging market, and as a result there is now a multiplicity of messaging solutions available. In addition to BlackBerry, messaging solutions are currently being marketed by Glenayre, Skytel, Metrocall, Motorola and River Run. AOL has also recently joined the marketing fray with its Mobile Communicator product (though this is in fact the BlackBerry system re-marketed under AOL's name).

However, all of these are closed solutions, and are therefore subject to precisely the same limitations, and ultimate non-viability, as BlackBerry.

The companies who are developing and marketing these solutions are suffering from strategic myopia. The existence of a plethora of closed, non-interoperating Mobile Messaging solutions results in technological and market fragmentation, which clearly cannot persist in the long run.

Sooner or later, the Mobile Messaging industry must and will adopt an open solution model. Any closed solution is doomed to eventual (and we believe speedy) extinction.

13.4 The WhiteBerry Solution

The basic concept of WhiteBerry is to bring the same true Mobile Messaging functionality to the user as does BlackBerry—but based on a set of open protocols, as opposed to RIM's proprietary protocols.

BlackBerry is a single-vendor solution, in which every system component is defined and supplied by RIM. By contrast, WhiteBerry is a multi-vendor solution, in which the same functionality is provided by a series of products and services, and the necessary cross-vendor interoperability is guaranteed by the openness and integrity of the underlying protocols.

WhiteBerry is thus not merely a competing system to BlackBerry—it is radically different in nature. WhiteBerry is not owned by anyone, any more than the economy is owned by anyone. Since the underlying protocols are completely open, WhiteBerry will expose every part of the generic Mobile Messaging value chain to market entry, cooperation, and competition. Mobile Messaging will be transformed from its incarnation of today, consisting of a melange of closed, non-interoperating systems, into a true industry.

13.4.1 Technological Components of WhiteBerry

All the technological components required to implement WhiteBerry already exist and are in place. These components are:

[4] As expected, Palm did deploy a closed solution subsequent to the initial publication of this paper in February 2001. At the time of current writing in November 2002, Palm still has no open solution.

- **Mobile Devices & Platforms**. A wide variety of mobile devices and platforms are available which satisfy the unconscious carry criterion, such as the Palm product line, Windows CE devices, and the already ubiquitious cellular telephone. Note that the BlackBerry end user devices are also part of the general pool of mobile devices, and if RIM wishes to participate in WhiteBerry on the basis of its mobile device technology, it can certainly do so.

- **Wireless Networks**. A variety of networks are in place that can provide the necessary coverage footprint, in-building penetration, and bandwidth. These include: GSM/GPRS, Packet CDMA, and CDPD.

  (Note: Throughout this paper we will frequently cite CDPD as an example of a Wireless IP network. This is not because we necessarily believe that CDPD has any greater relevance than other networks, but because as the largest native IP public wireless network in the world today, it provides a ready and familiar example. The WhiteBerry solution is in no way limited or specific to CDPD—WhiteBerry is compatible with *any* Wireless IP network.)

- **Wireless Modems**. A variety of wireless modems are available for the above networks, and in miniaturized form factors appropriate for unconscious carry usage. These include the AirCard product line from Sierra Wireless, and the Minstrel product line from Novatel.

- **Message Center Systems**. There are several commercially available Message Center systems that can provide generic back-end e-mail support. The major systems are: sendmail, QMail, and Microsoft Exchange.

- **Subscriber Services**. There exist a large number of ISPs and other Subscriber Services companies that already host Message Centers and provide e-mail service to their subscribers. Any of these can readily provide support for WhiteBerry Mobile Messaging.

Given that all these technological components are mature and available, the final component required to bring WhiteBerry to life is the right set of open protocols to tie everything together. As we will see in the following sections, these are now available too.

13.4.2 The Unifying Component: A Set of Open Protocols

The key to the open Mobile Messaging industry is the right set of protocols.

Existing Internet e-mail protocols are not suitable for the Mobile Messaging domain, because they are lacking in two major respects. First, they lack the necessary efficiency characteristics. Wireless networks are severely constrained by bandwidth limitations, and mobile devices are constrained by limitations such as display size, battery capacity, and memory size. These constraints place an extremely high premium on the efficiency of data transfer. Existing Internet protocols such as SMTP [**?**], IMAP [**?**] and POP [**?**] do not provide the required efficiency.

Second, existing Internet e-mail protocols do not properly support the push mode of delivery, which is an essential element of true Mobile Messaging. For more detailed discussions of the shortcomings of existing protocols, see the Manifesto articles *EMSD: The LEAP E-Mail Component* [**?**], and *Efficiency of EMSD* [**?**].

The inadequacy of existing protocols has been well demonstrated by the business failure of Mobile Messaging services based on them. Several service providers (Omnisky among others) have offered mobile e-mail services based on existing land-based Internet protocols such as POP, IMAP and SMTP. But because of the inefficiency of these land-based protocols, and because they do not support the push mode of delivery, these services are just no match for true Mobile Messaging systems like BlackBerry.

Therefore a new generation of high-performance, efficient protocols is required to address the demands of the mobile and wireless environment. The necessary protocols must satisfy the following key functional requirements:

- Support the submission and delivery of short (e.g. 4 kilobytes or less) Internet e-mail messages, while meeting or exceeding the functionality of the existing SMTP protocols.

- Meet or exceed the reliability and security of the existing SMTP protocols.
- Support push mode for immediate message delivery.
- Provide the required efficiency characteristics. These include: minimization of the number of transmissions, minimization of the number of bytes transmitted, and minimization of the latency of message submission and delivery.
- Make reasonable trade-offs between specification complexity, implementation complexity, extensibility, scalability and efficiency.

In addition to these technical requirements, the required protocols must be truly open. They must satisfy the conventions and criteria of openness that have come to be expected by the Internet mainstream community, ensuring that there are no restrictions whatsoever on their availability and usage. Specifically, the protocols must satisfy the following criteria:

- They must be functionally free of patents, licensing requirements, or any other form of usage restriction.
- They must be published as stable specifications which are freely, easily, and permanently available to anyone who wishes to use them. The ideal availability mechanism is Internet RFC publication.
- They must be open to public review and commentary; and participation in the maintenance, revision and enhancement of the protocols must be open and free to any interested party. The maintenance process must also preserve the patent-freedom of the protocols.

13.4.3 The Key to WhiteBerry: The LEAP Protocols

All the above requirements are fully satisfied by a set of mobile messaging protocols called the **Lightweight & Efficient Application Protocols**, or **LEAP**. LEAP is a set of high-performance, efficient protocols that are ideal for mobile and wireless applications.

LEAP is a layered family of protocols, of which two are of particular relevance to Mobile Messaging. The first of these is the Efficient Short Remote Operations protocol, or ESRO, published as RFC 2188 [**?**]. ESRO provides reliable connectionless transport services which can be used for a variety of applications, including but not limited to Mobile Messaging. For complete details on ESRO see the Manifesto article *ESRO: A Foundation for the Development of Efficient Protocols*, or visit the ESRO website at `http://www.esro.org/`.

Built on top of ESRO is the Efficient Mail Submission & Delivery protocol, or EMSD, published as RFC 2524 [**?**]. EMSD is a specialized native Internet messaging protocol which meets or exceeds the level of functionality, reliability and security provided by the existing SMTP protocols. EMSD has been designed with a major emphasis on efficiency, and fully supports the push mode of message delivery. For complete details on EMSD see the Manifesto article *EMSD: The LEAP E-Mail Component*, or visit the EMSD website at `http://www.emsd.org/`.

In addition to satisfying all the necessary functional and technical requirements, the LEAP protocols also fully satisfy the required openness characteristics. They are open in the broadest sense of the word: they are freely and permanently available, subject to public review and revision, and without usage restrictions of any kind. For complete details, see the Manifesto article *The LEAP Protocol Development Model* [**?**].

In principle, the WhiteBerry solution could be based on any set of protocols that satisfies the necessary requirements. Other than LEAP, however, we are aware of no set of protocol specifications that satisfies these requirements. Though we have actively searched for alternatives, to the best of our knowledge no such alternative exists. As the basis for WhiteBerry, therefore, the LEAP protocols are both ideal and unique.

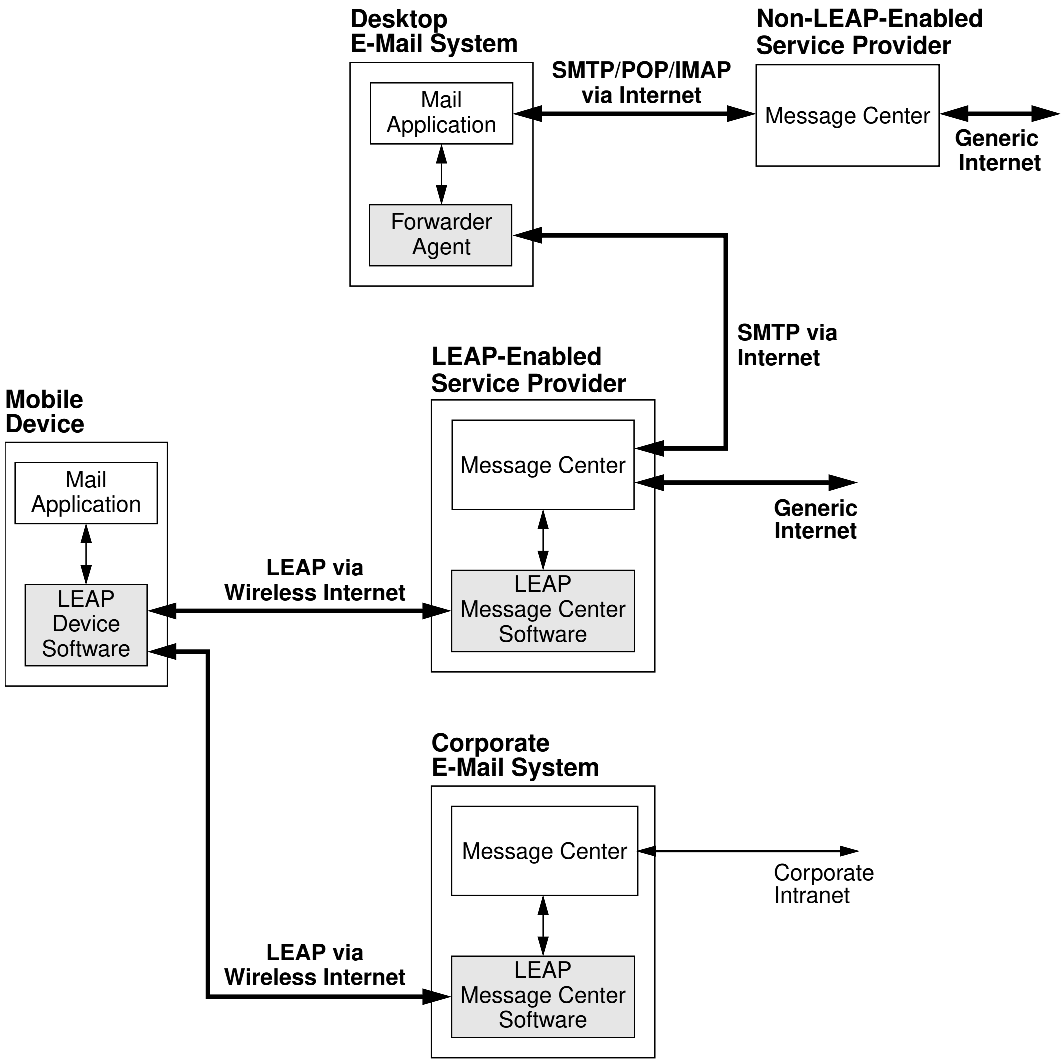


Figure 13.2: The WhiteBerry Solution

13.4.4 How WhiteBerry Works

The way in which the WhiteBerry solution brings Mobile Messaging capability to the user is illustrated in Figure 13.2. All LEAP-specific components are shown shaded in this figure. This figure is somewhat similar to Figure 13.1, and the description of WhiteBerry in this section is similar to the description of BlackBerry in Section 13.3.1. However, there are also very significant differences between the two systems.

First, the user must be provided with some form of mobile handheld device, as shown on the left side of the figure. In the case of BlackBerry, this could only be one of the two form factors provided by RIM. In the case of WhiteBerry, this can be *any* unconscious carry device, such as a PDA, a cell phone, or a dedicated two-way pager, from any device manufacturer who has chosen to adopt the WhiteBerry model by LEAP-enabling the device.

The mobile device must be equipped with a wireless modem. In the case of BlackBerry, this is the integral RIM modem for the BellSouth Intelligent Wireless Network. In the case of WhiteBerry, this can be *any* suitably miniaturized modem, which supports *any* appropriate wireless network, such as GSM/GPRS, Packet CDMA, or CDPD. Though the device may be turned off, the modem will normally remain on at all times to accept incoming messages.

The device must also be equipped with LEAP device software, allowing it to communicate via the LEAP protocols with LEAP-enabled Message Centers.

Next, the user must be provided with a LEAP-enabled Message Center to support his Mobile Messaging capability. In the case of BlackBerry, this role could only be played by the RIM-operated/licensed BlackBerry Message Center. In the case of WhiteBerry, however, there is considerably more versatility in how this service can be provided.

Under one scenario, Message Center service can be provided by an independent Subscriber Services provider, as shown in the center of Figure 13.2. Under WhiteBerry, this role can be played by *any* Message Center operator—for example, by any one of the large number of existing ISP companies. All that is required for an ISP or other Message Center operator to become a provider of LEAP-based Mobile Messaging services, is for them to install the necessary LEAP Message Center software.

Anyone with access to the Internet can now exchange e-mails with the mobile user. E-mails addressed to the mobile account are fielded by the Subscriber Service provider from the generic Internet using standard Internet protocols, then delivered to the mobile device using the LEAP protocols.

The device modem remains on at all times to accept incoming messages, and the device notifies the user immediately (in any of the ways commonly used for pager notification, e.g. beep or vibrate) whenever a new message arrives. The user can then activate the device and read the message.

To send a message, the user composes the message then submits it to the LEAP service provider via the LEAP protocols. The service provider then sends the message to its destination using standard Internet e-mail protocols.

Users typically wish their mobile messaging capability to function as a wireless extension of an existing land-based e-mail account. For example, the user may wish the mobile device to act as an extension of a home or office desktop mail application, as shown at the top of Figure 13.2. This functionality is provided by installing the appropriate mail forwarding software on the desktop system. This software integrates with the desktop mail application, and allows messages to be selectively forwarded to the mobile device on the basis of user-defined e-mail filters. Properly qualified e-mails are forwarded to the LEAP service provider using standard e-mail protocols, then delivered to the mobile device using the LEAP protocols.

When the user submits a message from the mobile device, the LEAP service provider sends the message to its destination as usual, and in addition it can send a notification to the desktop mail application, to maintain mailbox synchronization between the handheld device and the desktop system.

Under a different scenario, there may be a need to bring Mobile Messaging capability to a corporate e-mail system, as shown at the bottom of Figure 13.2. This functionality is provided by installing the appropriate LEAP software in the corporate Message Center. This endows the corporate Message Center with the same LEAP-enabled functionality as the independent LEAP Subscriber Service provider, which is therefore not needed under this scenario.

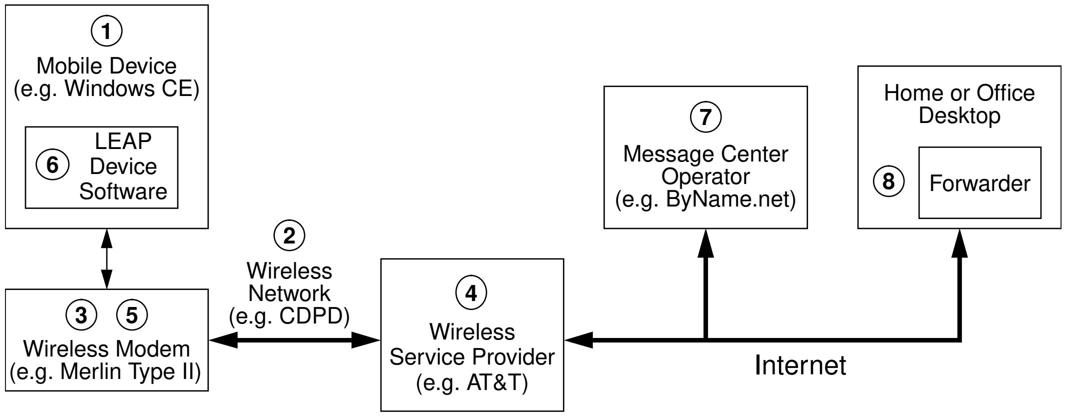


Figure 13.3: WhiteBerry Components

Like the desktop software, the LEAP Message Center software allows e-mails to be selectively forwarded to the mobile device. But since the corporate Message Center is fully LEAP-enabled, these can be delivered directly to the mobile device using the LEAP protocols.

This aspect of WhiteBerry operation constrasts starkly with the BlackBerry system, in which all messages *must* pass through the BlackBerry Message Center—even those forwarded from a peer Subscriber Service provider.

13.4.5 Putting Everything Together for the End User

As described above, the WhiteBerry solution requires coordinated integration among several components. Figure 13.3 shows how these components fit together to provide true Mobile Messaging capability to the end user. If a sophisticated end user were to assemble these components himself, he would need to follow these steps:

1. Acquire a suitable PDA; e.g. a Palm or Windows CE device.
2. Decide on an appropriate wireless network; e.g. GSM/GPRS or CDPD.
3. Acquire a wireless modem for the selected network, in a form factor compatible with the selected PDA; then attach the modem to the PDA. E.g. in the case of the Palm family of devices, and the CDPD network, suitable modems are available from Sierra Wireless and Novatel.
4. Purchase a subscription to the selected network through an appropriate network Service Provider; e.g. AT&T or GTE.
5. Activate the modem for use with the selected subscription, by configuring the modem as directed by the Service Provider.
6. Download the appropriate WhiteBerry client software for the selected PDA; e.g. from MailMeAnywhere.org.
7. Choose a WhiteBerry Message Center operator, e.g. ByName.net, and activate an e-mail account with that operator.
8. (If desired) link the mobile e-mail account to an existing home or corporate e-mail account by installing and configuring a rule-based forwarder on the existing home or corporate desktop system; e.g. by use of the Microsoft Outlook autoforwarding facilities, or by use of FetchMail and ProcMail.

And that's it. The user now has fully equivalent functionality to BlackBerry—but using his preferred choices of PDA, wireless network, wireless modem, network Service Provider, and Message Center operator. The user can assemble his WhiteBerry system using the best and/or most cost-effective option available for each component.

And if the user should happen to own a PDA/modem combination already, e.g. for wireless web browsing, then the user can provide himself with full Mobile Messaging capability *at zero incremental hardware cost*.

In principle, there is nothing to prevent an individual end user from doing all this for himself. In reality, of course, it is impractical for the typical user to do this on his own—which is why we have systems integrators. A systems integration company such as Omnisky can readily bundle together all the above components and present the end user with a complete, turnkey WhiteBerry solution.

These two scenarios (total assembly from scratch by the end user, vs. total third-party integration) represent two integration extremes. There are many shapes and sizes of partial integration lying between these extremes. For example, steps 3, 5 and 6 can readily be combined into a single product/service offering (as Omnisky does already, in fact, by packaging together a CDPD modem, AT&T account, and account activation as a wireless-enabling add-on for the Palm family of devices).

The open WhiteBerry paradigm allows a supplier to step into any conceivable market niche.

13.4.6 Technical Challenges & Responses

There are several challenges that can be made to the WhiteBerry model on technical and/or cost grounds. For example, the arguments can be made that wireless modems are too expensive to allow deployment of a cost-effective solution; or that battery longevity is too short to allow widespread acceptance by users.

However, these objections and others like them are ultimately without basis. In the case of the modem cost, and all other cost-related objections, the response is the same: modem and other hardware costs have been falling, and will continue to fall. There is nothing about any of the WhiteBerry hardware components to prevent them from eventually becoming available as inexpensive commodity items.

And there is no technical challenge that does not have a clear and straighforward resolution. To take the case of battery longevity, the responses are: (a) increased network cell density will reduce active device power consumption; (b) inactive device power consumption will be drastically reduced by implementation of network protocols for efficient device sleep mode, and also by close integration between the modem and the protocol software for sleep mode operation; and (c) inherent battery longevity will continue to undergo gradual improvement.

Clearly, there are no show-stoppers here. For a conclusive demonstration of this one need look no further than BlackBerry itself—if RIM can successfully bring a viable product to market, then so can others. And as the technical and cost considerations become even less relevant, things can only get easier.

A Word About Networks

In general, the desired characteristics of a network depend upon its intended usage. The ideal technical network requirements for paging, and for generalized wireless data transmission, are sufficiently different that these two applications have traditionally been considered to require different networks. Historically, paging networks (both one-way and two-way) have been considered unsuitable for generalized wireless data usage; while generalized wireless data networks have been considered inadequate for specialized paging and Mobile Messaging applications.

Because of this historical viewpoint, there is a continuing perception that only a certain class of networks are technically suitable for Mobile Messaging. However, this perception is incorrect.

The key functional criteria for suitability of a network for Mobile Messaging are:

- **Coverage**. In general, the greater the coverage footprint the better, but nationwide or global coverage is by no means essential. Depending on the requirements of the user, local or regional coverage may be more than sufficient.

- **In-building penetration**. In general, the higher the degree of in-building penetration the better. But again, depending on the usage pattern of the user, a lesser degree of in-building penetration may be adequate.

- **Device miniaturization and battery lifetime**. Today's mobile professional expects an unconscious carry form factor, and at least two full days between battery recharges. Among other things, the density of the network transceivers must be sufficient to meet both these requirements. A variety of networks satisfy this requirement.

- **Data throughput**. In general, a very high data throughput capability is not an essential requirement for Mobile Messaging. Because messaging is an inherently asynchronous activity, based on a store-and-forward paradigm, data latency is not a major issue. Messaging is entirely feasible on networks with data rates as low as 2400 bits per second.

Based on the above, it can be seen that a broad range of networks are entirely suitable for Mobile Messaging.

The BlackBerry system is currently limited exclusively to the BellSouth Intelligent Wireless Network. This network scores high marks on all four criteria listed above, and is therefore well-suited to Mobile Messaging.

But it is by no means the only network that is suitable for this. And the WhiteBerry model allows *any* suitable wireless data network to be used, including GSM/GPRS, Packet CDMA, and CDPD—which collectively account for most of the wireless network market of today. With proper industry cooperation and integration, a messaging solution based on any of these networks can compete directly with closed solutions such as BlackBerry, or those based on the closed ReFLEX protocol.

13.4.7 WhiteBerry versus BlackBerry

The major differences between BlackBerry and WhiteBerry are summarized in Table 13.1.

In the world of closed systems, competition with BlackBerry takes the form of other proprietary turnkey systems, which provide similar functionality.

The WhiteBerry model is fundamentally different from this: it is an open industry model. Since the mobile messaging solution is based on a set of open protocols, each component of the solution is exposed to free competition. Vendors and service providers can now compete in every aspect of the mobile messaging solution: handheld devices, wireless modems, networks, message centers, and the provision of Subscriber Services to the user.

The result of this is not just a set of competing, closed, non-interoperating solutions. Rather it is a thriving industry, in which active and independent competition takes place within every component of the user's ultimate value proposition.

WhiteBerry Patent Resistance

As noted previously, RIM has been granted a U.S. patent on the basis of its BlackBerry technology, which potentially threatens other closed Mobile Messaging systems.

The WhiteBerry solution, however, enjoys a major advantage over closed systems in terms of its vulnerability to patents. Unlike BlackBerry and other closed systems, WhiteBerry is not a single static messaging solution; rather, it is a highly mutable meta-solution. That is, any particular WhiteBerry implementation is created by integrating together an appropriate set of components, so as to achieve the particular functionality desired by the systems integrator or the end user.

	BlackBerry	**WhiteBerry**
Mobile Device	Only the two RIM-manufactured devices	Any suitable mobile device
Wireless Modem	Only the integral RIM modem	Any suitable wireless modem
Wireless Network	Only the BellSouth Intelligent Wireless Network	Any suitable wireless network
Message Center Service	Only the RIM-operated or RIM-licensed service	Any independent service provider; any corporate e-mail system
Protocols	Proprietary RIM protocols	Open LEAP protocols
Desktop Integration	Only Microsoft Outlook	Any desktop mail application
Message Center Integration	Only Microsoft Exchange	Any Message Center system
System Integration	Exclusively by RIM	Any systems integrator
Security	Not true end-to-end Implementation details unknown Precludes other implementations	Open paradigm permits external security implementation

Table 13.1: BlackBerry vs. WhiteBerry

The components that go into any given WhiteBerry implementation may be drawn from a large family of components which includes the EMSD protocol engines, FetchMail, ProcMail, mail forwarders, and various others. Each of these components is independent, freely available, useful in its own right, and entirely unrelated to the RIM patent.

Because of this inherently component-based nature, the WhiteBerry solution is exceedingly resistant to the RIM patent claim, and indeed to patent infringement claims in general. By making all the necessary components freely and publicly available, WhiteBerry provides systems integrators and end users with a variety of methods and strategies to circumvent or nullify invalid patent assertions.

13.5 Framework for Development

All the essential components required to implement WhiteBerry now exist. As discussed in Section 13.4.1, all the technological components (devices, modems, networks) are well established and readily available.

The necessary open protocols are also ready and available: the LEAP protocols are complete and have been published as Internet RFCs, and open support organizations for their continued revision and enhancement are up and running at LeapForum.org, ESRO.org and EMSD.org.

In addition, a framework for the development of WhiteBerry solutions has been established. This framework consists of:

- Open-source software implementations of the LEAP protocols for various devices and Message Center platforms
- An open public forum for the development and distribution of LEAP-related integration tools and facilities
- A set of free Subscriber Services to support immediate deployment of the WhiteBerry solution in end user devices

- A resource center providing starter kits for various types of implementor, and an initial end-to-end WhiteBerry implementation

Each component of this development framework is described in the following sections.

13.5.1 Open-Source Software Implementations

Open-source software implementations of the LEAP protocols are available for all major device and Message Center platforms.

Reference Protocol Engines have been implemented in the form of portable code, which has been ported to a wide variety of platforms and end user devices. On the device side, software has been implemented for pagers and cell phones; for palmtop devices (Windows CE, Pocket PC, Palm OS, EPOC) [**?**] [**?**]; for Windows 2000, Windows 98/95, and Windows NT; and for Pine (UNIX, Windows, DOS). On the Message Center side, software is available for Windows NT, Solaris, and Linux.

All of this software is freely and publicly available at the **MailMeAnywhere** open-source software distribution center. The software is available as free software, subject to the General Public License (GPL) [**?**]. For complete details visit MailMeAnywhere at `http://www.MailMeAnywhere.org`.

13.5.2 The MailMeAnywhere Development Forum

The protocol engines provide the basic message submission and delivery capability for the mobile environment. But this capability must be supplemented by a variety of other tools and capabilities in order to bring a complete, well-integrated solution to the end user.

Among other things, the user must be provided with flexible tools for selecting which e-mail messages are forwarded to the mobile device, and which can remain in the fixed mailbox for later review. The end-user solution must also include facilities to ensure proper synchronization and continuity between the mobile account and the fixed account.

In addition, there is a need to provide Mobile Messaging solutions in a variety of contexts. A mobile user may have a fixed e-mail account in any of various shapes and sizes, such as a corporate account or an ISP-based account, and he may wish to control and administer his mobile capability in various ways, such as via his desktop mail application, or via his corporate or ISP-based mail server.

In order to deliver this broad heterogeneity of Mobile Messaging implementations, the integration and development community requires a variety of tools for accessing and processing the user's fixed mailboxes. In particular, comprehensive and close integration of the LEAP protocol engines will be required with mail routers, mail retrievers, mailbox synchronizers, inbox filters and rule- and directory-based fowarders.

For this purpose, MailMeAnywhere includes an open public forum for the development and distribution of the necessary tools and capabilities. The MailMeAnywhere development forum is a public framework for the collective development and enhancement of LEAP integration tools on a continuous, on-going basis.

This open development model, based on open protocols and open-source software, ensures the availability of a rich and expanding suite of tools and accessories for integration of the WhiteBerry solution in various configurations and across multiple platforms.

This open, heterogeneous development model is in sharp contrast to BlackBerry's narrow focus on Microsoft Outlook and Exchange.

13.5.3 ByName Subscriber Services

The open WhiteBerry paradigm must address a classical bootstrapping problem: a useful WhiteBerry solution requires implementation of the LEAP protocols in both devices and Message Centers, each of which cannot function without the other.

A set of Subscriber Services has been created to address this problem. The **ByName Services** are a set of comprehensive, well-integrated, and highly personalizable computing and communications services. They are fully WhiteBerry-enabled, and can provide complete and immediate support for WhiteBerry end-user devices.

In addition to playing this WhiteBerry support role, the ByName Services are also the world's first **Libre Services**. Libre Services are an extension of the principles of the Free Software movement to the domain of Internet Application Services. As Libre Services, the ByName Services are based entirely on patent-free protocols, and implemented exclusively in free and open-source software. As a result, the entire ByName construct can be copied and redistributed by anyone. Everything required to recreate the ByName Services *in their entirety* is freely available at the MailMeAnywhere software distribution site.

The ByName Services are implemented as a pair of sister sites:

- My.ByName.net
  `http://My.ByName.net`
- My.ByName.com
  `http://My.ByName.com`

My.ByName.net is a showcase site, demonstrating the concept and capabilities of Liber Services. My.ByName.com is a hands-on look-and-see environment, provided for try-out by the public. Anyone can sign up for a free account at My.ByName.com.

The two sites provide identical services, including complete WhiteBerry support. In particular, anyone can sign up for completely free WhiteBerry service at My.ByName.com.

13.5.4 The WhiteBerry Resource Center

The WhiteBerry Resource Center is a central location where developers can pick up everything they need to begin implementing WhiteBerry solutions. A set of starter kits are available, providing the resources required by various types of implementor: individual, corporate, ISP/ASP, or systems integrator. For complete details visit the resource center at `http://www.mailmeanywhere.org/wbStartKit/`

The starter kits provide complete step-by-step descriptions of how to build a WhiteBerry solution. As a case-study example, we will describe the experiences of an individual implementor—let's call her Lisa Simpson. The steps Lisa followed and the choices she made in assembling her WhiteBerry system are summarized in Table 13.2. The numbered steps in this table correspond to the generic steps described in Section 13.4.5 and Figure 13.3.

At the end of this process Lisa had her very own, fully functional, true Mobile Messaging solution—and so can you.

Initial WhiteBerry Implementation

The WhiteBerry Resource Center also provides an initial WhiteBerry implementation, which can be freely used as a development tool by anyone—for examination, enhancement, or as the starting point for further implementations.

An initial base implementation, of course, is not the same thing as a comprehensive, market-ready messaging solution. But the existence of this implementation clearly demonstrates the viability of WhiteBerry as a complete,

WhiteBerry Step/Component	Lisa's Choice
1. Select a PDA	Lisa chose to use a HP 660LX palmtop device, running Windows CE 2.0
2. Select a wireless network	Lisa selected the CDPD network
3. Select a wireless modem	Lisa selected the Sierra Wireless Aircard 300, a CDPD modem which is compatible with her HP660LX
4. Select a network Service Provider	Lisa chose AT&T since she resides in Seattle, and AT&T is the CDPD Service Provider for the Seattle area
5. Activate the modem	Lisa provided AT&T with the modem's EID (Equipment ID) number, received an IP address from AT&T, then configured the modem to use that address
6. Download the LEAP device software	Lisa went to the MailMeAnywhere.org website, and downloaded the appropriate Windows CE LEAP software – in this case SH3 Gold Version 1.2
7. Select a Message Center operator	Lisa set up a free e-mail account for herself at ByName.net
8. Select and install an e-mail forwarder	Lisa used FetchMail and Emacs Lisp code to define her directory and rule-based forwarding preferences

Table 13.2: A WhiteBerry Case Study Implementation: Lisa Simpson

end-to-end messaging solution. It means that WhiteBerry is not just an idea—it is an actual working system, which is up and running today.

13.6 Mobile Messaging Security

We claim in this article that WhiteBerry meets or exceeds the entire functionality of BlackBerry. There is one possible objection that could be made to this statement: that, as described, WhiteBerry does not match BlackBerry's claimed security functionality.

13.6.1 BlackBerry Security

In their promotional literature, BlackBerry claims to provide "end-to-end" security using Triple DES encryption technology. A casual reading of this would lead one to believe that BlackBerry provides full message encryption all the way from the sending mail application to the receiving mail application—i.e. all the way from the point of origin to the point of delivery. In the Internet world, this is conventionally what "end-to-end" means.

In fact, the BlackBerry system only provides encryption between the BlackBerry handheld and the corporate e-mail account. This provides security only for messages originating in or being delivered to the in-house corporate e-mail system. In the case of generic e-mails, which may be transmitted to/from any point on the Internet, BlackBerry provides no security whatsoever beyond the point of relay at the corporate system.

In other words, one of the ends being referred to in BlackBerry's claim of "end-to-end" security is not the ultimate user mail application, but the corporate account. BlackBerry thus only provides security over part of the transmission route from sender to recipient. This is rather like going out to do battle wearing a suit of armor that only offers protection from the waist down—it is only slightly better than no armor at all.

Therefore BlackBerry does **not** provide true End-to-End security, and RIM's use of this term is disingenuous and misleading. The claimed security is limited and superficial, providing only a degree of psychological reassurance to the user—it is more of a security blanket than genuine security.

Furthermore, since the BlackBerry system is closed, details of the actual security implementation are unknown, and have not been subjected to external review. RIM's description of its security method may be adequate for marketing purposes, but this does not provide sufficient information for a formal engineering validation.

In general, e-mail users can achieve true End-to-End security by means of the existing PGP [**?**] or Secure MIME (S/MIME) [**?**], [**?**] technologies. These are the two methods most commonly used for e-mail security today. However, if these technologies are to provide true End-to-End security, they must be implemented at both ends—i.e. at both the sending and receiving e-mail applications. But since the BlackBerry devices are closed, it is not possible for a BlackBerry user to make use of these or any other independent security mechanisms.

Therefore not only does the BlackBerry system not provide genuine End-to-End security, its closed nature also precludes implementation of genuine End-to-End security by any other means.

13.6.2 WhiteBerry Security

There is no doubt that security is of major importance in the e-mail domain. However, the basic Internet e-mail protocols (SMTP, POP and IMAP) were originally designed without adequate consideration for the security issues. Because of this, and because of certain flaws in the implementation of these protocols, the Internet e-mail structure of today includes serious security compromises.

Nevertheless, the existence of the above security compromises has not prevented the widespread adoption and usage of Internet e-mail. Today, much of the daily e-mail traffic still takes place without any formal security protection. In the case of those messages which do require secure transmission, additional security mechanisms are typically applied.

There is a perception that there is a greater need for security in the wireless world than in the wired world, because of the far greater ease of physical access to the wireless communication channel. However, most modern IP-based networks already provide a confidential data link at Layer 2, usually obviating the need for additional over-the-air, point-to-point confidentiality mechanisms.

Things have changed a great deal since the early days of Internet e-mail, and today's security expectations and requirements exceed the capabilities of existing protocols.

Mobile Messaging must be a seamless and consistent extension of the existing Internet e-mail structure. Likewise, in order to provide true end-to-end security over both the wired and wireless Internet, the security mechanisms for Mobile Messaging must be an integral part of the overall Internet e-mail structure. Since WhiteBerry is designed as a natural extension of Internet e-mail, it provides the same or better level of security as the existing SMTP protocols.

Also, the WhiteBerry paradigm fully supports the implementation of existing Internet e-mail security mechanisms such as PGP or S/MIME. For those e-mail applications that require it, true End-to-End security can readily be implemented in the context of WhiteBerry by means of these technologies.

However, the currently available implementations of PGP and S/MIME are not the ideal solution for the Mobile Messaging domain. The major factor that is genuinely different in the wireless world is the set of constraints imposed by device miniaturization. For this reason, and in contrast to the wired world, there is a crucial need for efficiency. The currently available implementations of PGP and S/MIME impose undesirably heavy computational demands on the CPU-limited mobile devices, and also impose undesirable message transmission overhead on the bandwidth-limited wireless networks.

Therefore the long-term solution to the need for security in the wireless domain is to provide mainstream security mechanisms such as PGP and S/MIME, but based on more efficient public key algorithms and mechanisms such as elliptic curve cryptography. This is the WhiteBerry approach to security, and one of the planned future tasks will be to extend the LEAP protocols to provide efficient security mechanisms. It is anticipated that this work will take place at EMSD.org.

In addition, generalized security services (in particular authentication and confidentiality) can be provided by establishing a Secure Short Remote Operations (SSRO) layer on top of ESRO. It is anticipated that work related to SSRO will take place at ESRO.org.

Meanwhile, the existing WhiteBerry security mechanisms are adequate for most wireless e-mail traffic, and where necessary they can be augmented by conventional End-to-End security mechanisms.

13.7 The Business Case for WhiteBerry

The Mobile Messaging industry of today is limited and fragmented by the existence of a number of competing, non-interoperating, closed solutions. For this reason the industry remains only a fraction of its eventual size and scope.

By shifting competitive engagement from system-level to component-level, the open WhiteBerry solution will change everything. WhiteBerry will expose the entire Mobile Messaging value chain to market entry, resulting in a flood of innovation, cooperation and competition, and catalyzing enormous industry growth.

13.7.1 Immediate Opportunity: Installed Hardware Base

There is a substantial segment of the messaging market that can be targetted without any product development effort whatsoever, and with minimal sales effort. This segment consists of those users who can be provided with the WhiteBerry solution immediately and at virtually zero cost, *because they already own the necessary hardware*.

Any user who owns an appropriate device/modem combination is a latent WhiteBerry user. To take the example of a specific network (say CDPD) and a specific device (say Palm), note that every person who owns a Palm device and a compatible CDPD modem can become a WhiteBerry user *at zero hardware cost*. All that such users need is the necessary LEAP software. The same applies to other network/modem/device combinations.

This pre-enabled market segment is quite substantial. In terms of modems, as of February 2001 there is an estimated existing installed base of between 250,000 and 300,000 PDA-compatible CDPD modems; and the installed base of other wireless IP modems such as Packet CDMA is also rapidly growing. In terms of devices, there is an estimated installed base of around 20 million PDAs such as Palm, Windows CE, etc. [**?**]. It is also worth noting that a user who owns one of these components (i.e. device or modem) but not the other can be provided with WhiteBerry functionality at correspondingly discounted incremental hardware cost.

Furthermore, since the owners of these components are already known to the component vendor, they can be targetted directly. A ready-made marketing/distribution channel to these users is immediately available to the appropriate hardware manufacturer or integrator. Companies who can take immediate advantage of this include modem manufacturers such as Novatel and Sierra Wireless, and hardware integrators such as Omnisky.

In summary, the installed base of device/modem owners represents a golden business opportunity because:

- This is a large market segment
- These users can be provided with equivalent functionality to BlackBerry at zero cost, as opposed to the high cost of BlackBerrry
- These users are precisely known and reachable

This segment therefore provides the opportunity for very rapid initial deployment and propagation of the WhiteBerry solution. Beyond this segment, deployment of WhiteBerry solutions in general requires extremely low development cost, while offering truly gigantic upside potential.

13.7.2 Precedents for Success

The skeptical reader may have doubts that the LEAP-based WhiteBerry solution can succeed, on the grounds that it is a market newcomer, going head-to-head with an already established competitor—a sure-fire recipe for business failure.

This belief, however, represents a basic misunderstanding of the fundamental nature of WhiteBerry and BlackBerry. These are not peer competitors; rather, they are two entirely different business models, one of which has inherent long-term viability, while the other does not.

History shows that, other things being equal, the underdog loses. But in the case of BlackBerry and WhiteBerry, other things are far from equal. BlackBerry is a closed commercial construct; while WhiteBerry is an open and egalitarian industry enabler. Under these circumstances, history has very different lessons to teach.

In this case, a repeated historical pattern is for a new concept or technology to be initially deployed in the form of one or more closed, commercial implementations. Once these commercial implementations have proved the basic value proposition, they are followed by free and open implementations of the same concept, which then overtake and supplant the commercial implementations. There is no shortage of historical precedent for this. Consider the following examples:

- The proprietary networking protocols SNA from IBM, and DECnet from DEC, both of which were followed and rendered obsolete by TCP/IP
- Various closed operating systems, followed and rendered obsolete by GNU/Linux
- More recently, ASP (Microsoft's server-based web development tools) followed and supplanted by PHP

In every one of these cases an entrenched solution, representing commercial vested interests and with aggressive industry backing, was supplanted by an open equivalent that at the outset had no industry backing at all. Anyone interested in the future of closed messaging systems would do well to study these case histories.

The BlackBerry/WhiteBerry scenario follows precisely the same pattern, and this is why WhiteBerry can and will prevail.

13.7.3 Shifting Opportunities: Winners & Losers

From the point of view of the commercial interests they supplant, these open usurpers have the effect of ruining a profitable business, typically without any apparent profit motivation on the part of the open system proponents. To the commercial interests, this appears to be an act of economic vandalism—the destroying of a viable business opportunity, for no good reason. For this reason, these open systems are often characterized as market "spoilers."

But this is a narrow view. At the very least, open systems merely shift economic benefit from one place to another. And in virtually all cases, the industry-wide adoption of open systems results in greatly increased overall economic benefits.

WhiteBerry will certainly act as a spoiler for the closed Mobile Messaging industry. Simply by distributing LEAP in the form of open-source software solutions, we can ruin the market for any closed messaging solution.

In so doing, we will shift economic benefit from the closed system purveyors, to the many users who would otherwise have purchased a closed solution. But our intention is to accomplish far more than this simple dollar shift—our intention is to create greatly increased opportunities and benefits.

The key to accomplishing this resides in the participation of others. With this participation, WhiteBerry will move beyond being a mere industry "spoiler," and will create enormous new business opportunities. We actively welcome and encourage such participation.

The losers in this will be the closed-system marketeers such as RIM. But the winners will be the many more companies who are able to participate in the Mobile Messaging market—and the end user who will benefit from the resulting competition and innovation.

13.7.4 Business Conservativism

In the data communications world, the era of the closed system is dying. This is because it has now become clear that:

> *In the domain of large-scale interconnected systems, open systems always displace closed systems.*

The historical precedents for this are clear and irrefutable; yet despite this observation, businesses doggedly persist in deploying closed systems such as BlackBerry, or pseudo-open ones such as WAP. One may reasonably ask: Why is this? What is the great allure for businesses of closed systems?

The answer is that historically, closed systems have provided precisely the sort of classical sustainable advantage that MBAs are taught to seek. And there are certainly many business arenas in which closed systems do, in fact, provide a long-term competitive advantage.

However, the domain of large-scale interconnected systems is not one of them. In this domain things work differently, and here closed systems inevitably lose to open systems. But this domain is new, and its new rules are not yet well understood within the mainstream of business theory. Business thinking remains rooted in the traditions of conventional products and services, and continues to apply traditional principles to an entirely new domain, *to which these principles do not apply*.

But the opportunities for closed systems are diminishing. The time lag between the deployment of a closed system, and its supplantation by an open equivalent—i.e. its period of profitable tenure—is growing shorter. There is no better example of this than WhiteBerry itself, which is broadsiding BlackBerry even before it has fully left the ground.

At watershed moments like this, in which there is some sort of fundamental technological migration, the business community is always the last to know.

A perfect example is the case of wired e-mail. As recently as 1990, all members of the Electronic Mail Association (EMA) were in business terms solidly committed to X.400 (i.e. e-mail based on the telecommunications infrastructure, as opposed to the Internet infrastructure) as the long-term e-mail solution. But in the engineering world it was always perfectly clear that Internet e-mail (i.e. SMTP, POP etc.) would win out over telecommunications e-mail—as, of course, it did.

The evolution of WhiteBerry will likely be no different. At this point, WhiteBerry may seem radical and implausible to the business world, but this merely represents a general lack of understanding and imagination on the part of business. In the engineering world, the case for WhiteBerry is already conclusive. Initially therefore, it will be the responsibility of the engineering community to lead the way forward.

13.8 Framework for Participation

We refer to the widespread industry adoption of the WhiteBerry model, appropriately enough, as **Operation WhiteBerry**. The intended scope of Operation WhiteBerry is extremely broad. We are not proposing the creation of single integrated solution to compete with BlackBerry. Rather, we are proposing that WhiteBerry appear in multiple implementations within every component of the Mobile Messaging value chain:

- On all appropriate device platforms: Palm, Pocket PC, HandHeld PC, EPOC, cell phones
- Over all appropriate wireless networks: GSM/GPRS, Packet CDMA, CDPD, Wi-Fi/802.11
- Integrated with all appropriate mailbox applications: Microsoft Outlook, Netscape Messenger
- Integrated with all appropriate Message Center systems: sendmail, QMail, Microsoft Exchange
- In all appropriate scales and configurations: desktop, corporate, ISP

Clearly, this is not something that can be accomplished by any single company acting alone. In order to achieve the full potential scope of Operation WhiteBerry, participation and action throughout the entire Mobile Messaging industry will be required.

Operation WhiteBerry is first and foremost an engineering construct as opposed to a business one—we are not promoting a business model, or actively soliciting business partnerships. Operation WhiteBerry is an analysis of an industry problem, a proposed solution, and a roadmap for implementing this solution.

We intend to move forward with the implementation of this solution. We will continue to expand and enrich the development framework described in Section 20.4. We will continue to create additional device and Message Center software implementations of the LEAP protocols, and distribute these in the form of free, open-source software. Based on the starting point WhiteBerry implementation, we will create new and enhanced implementations. And we will continue to promulgate the WhiteBerry model throughout the engineering community.

By means of these activities, we expect to create a grass-roots awareness and adoption of the WhiteBerry model throughout the industry.

However, the scope of this awareness and adoption will be greatly augmented, and its pace will be greatly accelerated, by the proactive participation of others; and for this reason we have established a comprehensive framework to accommodate and support such participation. Details of this framework are provided in Section 13.8.5.

We therefore invite and encourage active participation throughout the industry community. The major challenges and opportunities are described in the following sections.

13.8.1 Device Integration

A variety of LEAP protocol engines are already complete and available. A base device LEAP protocol engine is available in the form of portable software that can be integrated with various devices, operating systems and platforms. A number of such device-side portations and integrations have already been completed, and portations to additional devices and platforms are in progress.

A Java/J2ME implementation of ESRO (a lower-layer LEAP protocol) has also been completed, allowing immediate deployment of ESRO in Java-enabled cell phones. Development of a Java implementation of EMSD is currently in progress; when complete this will allow deployment of the full WhiteBerry solution on Java-enabled phones.

All existing portations and implementations are freely available at `MailMeAnywhere.org`; in-progress implementations will be made available as they are completed.

Palm and Windows CE are the two most important device platforms for initial implementation of WhiteBerry, and open-source implementations of the LEAP protocols already exist for both these platforms. However, as described below, there are opportunities for enhancement, and we invite others to participate in this.

In addition, we invite others to port the LEAP software to additional general-purpose device platforms, based on the existing open-source implementations. In particular, we invite cell phone manufacturers to include the LEAP protocols as an integral part of their next generation devices.

Windows CE Enhancement

Integration of the LEAP protocol engine with Windows CE has been facilitated by the existence of well defined APIs (Application Programming Interfaces) in the Microsoft Windows CE environment.

A standard mail application called Inbox is bundled with all Windows CE devices. Microsoft has defined a generic mail transport service provider interface underneath Inbox, which allows alternative mail submission and delivery protocols to be integrated with Inbox. Microsoft has also defined a winsock interface which provides access to UDP and TCP by third party applications.

These well defined APIs allow a new mail submission and delivery protocol to be integrated with Inbox in the Windows CE environment. For more information see the Manifesto article *EMSD on Windows CE* [**?**]. The existing open-source implementation is available at http://www.mailmeanywhere.org. See Section 13.5.4 for more details.

In addition, recent versions of Windows CE now support the push mode of delivery, which, as we have noted, is an essential element of true Mobile Messaging. For these reasons, the existing portation/integration of the LEAP protocols with the Windows CE environment is mature and fully functional, including immediate delivery and notification to the user.

However, there are at least two desirable enhancements which can be made to the existing Windows CE integration:

1. The Inbox mail application is fundamentally based on the classical e-mail retrieval (poll) paradigm, rather than the pager-style delivery (push) paradigm. Though Windows CE now supports this paradigm, it is an add-on feature which is not well integrated as a native capability of Inbox.

   We invite the designers of the Inbox mail user agent to correct this shortcoming. We invite them to think of the Inbox application not just as a traditional mail user agent, but also as a paging application, and to include the appropriate facilities as a native part of the mail service provider interface. Meanwhile, implementations of LEAP in the Windows CE environment must take responsibility for this.

2. In order to minimize battery power usage, it is very desirable for the integrated solution to make use of the wireless modem's sleep mode of operation. The existing implementation of LEAP in the Windows CE environment includes the necessary virtualization layers for this purpose; however, the participation of the modem manufacturer is required to complete the necessary integration.

By now the importance of playing a major role in the Mobile Messaging industry cannot be lost on Microsoft. And Microsoft has a powerful set of directly relevant assets (MSN, Exchange, Windows CE etc.). Microsoft could easily build a closed Mobile Messaging solution based on these assets, and the temptation to do so must be very strong. However, we urge them to resist this temptation. Rather, we urge them to consider the long-term consequences of an open solution. These consequences include enormous industry growth, and corresponding benefits to the owners of assets precisely like those of Microsoft. Therefore a strategic decision by Microsoft to eschew closed in favor of open is not just an act of good corporate citizenship (not that this has ever been a major consideration for Microsoft)—it is also good business.

The inclusion of WhiteBerry capability has the effect of transforming a Windows CE device into a true Mobile Messaging device, and therefore represents an extremely powerful value-added component. In fact, we believe that the WhiteBerry value proposition is sufficiently strong to justify full integration of this capability into the operating system.

We therefore invite Microsoft to embrace the open WhiteBerry paradigm by including a complete implementation of EMSD as an integral part of the Windows CE distribution, rather than relying on third parties to do the necessary OS integration.

Palm OS Enhancement

A mail application is included as a standard component in the Palm OS environment. In contrast to Windows CE Inbox, however, the Palm OS mail application has only a very basic set of features and capabilities. Also in contrast to Windows CE, in Palm OS there do not exist well defined APIs for integration of alternative mail protocols.

For this reason, integration of the LEAP protocol engine in the Palm OS environment has not been as convenient as in Windows CE.

To complicate matters further, various third party mail user agents have now become widely available for Palm OS (largely because of the inadequacy of the standard Palm OS mail application), with a variety of different user interfaces. Because of this multiplicity of incompatible Palm OS mail applications, each integration of LEAP with Palm OS is specific to a particular mail user agent.

The existing implementation of LEAP for Palm OS [**?**] is specific to the standard mail application, based on file access. This implementation is available at MailMeAnywhere.org. Further integrations and portations of LEAP for the Palm OS environment are currently in progress.

Like Microsoft, Palm has a powerful messaging-related asset in the form of its family of devices. Unlike Microsoft however, Palm has succumbed to the temptation to build a closed solution based on this asset; it has recently announced its intention to bring a Mobile Messaging solution to market by the middle of 2001 [**?**]. As in the case of Microsoft, and for precisely the same reasons, we urge Palm to reconsider this decision.

Just as for Windows CE, WhiteBerry capability represents a powerful value-added component for Palm OS, and again justifies its full integration into the operating system. As in the case of Microsoft, therefore, we also invite Palm and Handspring to adopt the open WhiteBerry paradigm by including a complete implementation of EMSD as an integral part of Palm OS.

13.8.2 Modem Integration

More than any other constituency, the participation of the wireless modem manufacturers is essential for Operation WhiteBerry. We invite modem manufacturers such as Sierra Wireless and Novatel to include the LEAP protocols as an integral part of their next generation devices.

In addition, there is a requirement for tight integration between the modem and the protocol software for proper sleep mode operation. The modem must always remain ready to receive messages; then as soon as a message arrives the modem must wake up, acknowledge it at the correct level, and deliver it to the user. The current level of integration is inadequate for this. The involvement of the modem manufacturers is essential to achieve the required integration.

Challenge to Modem Manufacturers

Historically, modems have been thought of as a device that resides at or below Layer 3, and modem manufacturers continue to think of them this way.

However, the wireless modem is the key, non-generic hardware component of any Mobile Messaging device. The combination of a wireless modem and a PDA is fully capable of being a Mobile Messaging device—all that is required is that the LEAP protocol engine be included with the modem.

Our challenge to modem manufacturers is therefore this: Why not think of your device, not just as a pure Layer 3 device, but as a Mobile Messaging device? What reason is there to limit yourself to building and marketing modems, when the proven market application is Mobile Messaging devices? The costs of moving into this market are negligible, while doing so provides the following major business advantages:

- Allows immediate entry into the applications market above Layer 3.

- Makes your product of immediate value to the end user, allowing you to target this market directly.
- Allows you to establish a direct relationship with the end user, including brand-name awareness on the part of the end user.
- As modems increasingly become a commodity item, allows your product to remain distinct from generic competitors.

The traditional technological role of modems, that their manufacturers continue to maintain as a self-imposed limitation, is no longer valid. By stepping beyond this limitation, modem manufacturers can address a gigantic new market.

To direct our challenge towards a specific set of companies, we will take the case of the CDPD network as an example. The major manufacturers of CDPD modems are: Sierra Wireless, Novatel, Tellus and Nextcell (now Enfora). So our challenge to these four companies is this: What argument do you have with any of this?

Cell Phone Integration

The discussion we have presented above is not limited to the traditional add-on modem, but applies more generally to *any* appropriate airlink device. In particular, the discussion applies to cell phones that have data communications capabilities. While airlink functionality is usually provided to a PDA device by means of an add-on modem, the airlink requirement can in principle also be provided by a data-capable cell phone—provided the cell phone has a suitable means of communicating with the PDA.

Under such circumstances the add-on modem can be dispensed with, and the user can be provided with Mobile Messaging capability by means of a data-capable cell phone, closely integrated with a PDA. Since both the cell phone and the PDA can be best-in-class, user's-choice devices, this represents a very powerful implementation of the WhiteBerry model. This scenario is described in greater detail in the Manifesto article *WhiteBerry and BlueTooth* [**?**].

An alternative to the physically separate cell phone and PDA is the integrated cell phone/PDA, which provides the user with voice communications, Mobile Messaging, and traditional PDA data management capabilities, all in a single device.

Regardless of the relationship between the cell phone and the PDA, we invite the cell phone manufacturers to adopt the Internet End-to-End model, and include the LEAP protocols as an integral part of their devices. Note that in the case of Java-capable cell phones, the availability of the Java/J2ME implementation of LEAP allows the phone to be immediately WhiteBerry-enabled.

13.8.3 Network Services Integration

Wireless ISP Integration

Existing network infrastructures are already adequate for Mobile Messaging applications; however, there are several technical enhancements that can improve the value proposition to the end user.

The key network-based user expectation is ensured reachability. In other words the user wants to be able to go anywhere, and still be confident that he will get his messages. The way in which the network can raise the satisfaction of this expectation is by means of increased coverage. Therefore a steadily increasing coverage footprint will result in steadily increasing end user satisfaction.

A further network-related user requirement is battery lifetime. Two specific ways in which the network can increase battery longevity are by means of increased cell density, and by the implementation of network protocols for highly efficient device sleep mode.

Network Operator Business Practices

Perhaps more important than these technical enhancements, network operators must also modify their business thinking, or run the risk of being bypassed.

Existing telecommunications-based and phone-based Mobile Messaging solutions are severely limited. Although Short Message Service (SMS) has experienced widespread usage it is not IP-based, and is therefore inconsistent with the Internet model. For this reason it cannot be considered a long-term strategic solution, and will eventually be replaced by mainstream Internet solutions.

The continued focus of the telecommunications industry on non-end-to-end services such as SMS is a hindrance to the development of the Mobile Messaging industry. The telecommunications industry must recognize that their continued investment in dead-end technologies like SMS and WAP is counterproductive, and must shift their focus towards alignment with the Internet model.

The crucial issue with regard to network operation—both in technical terms and in business terms—is that of conformity to the Internet End-to-End model. While a wireless network such as CDPD may conform to the Internet End-to-End model in technical terms, its operators may greatly inhibit End-to-End usage by means of their business practices. They may limit End-to-End usage by means of high pricing of such usage, while simultaneously subsidizing network access by controlled gateways such as WAP. The result is that in practical terms the network is only available for privileged, non-End-to-End usage—it becomes what is sometimes referred to as a Walled Garden.

This type of access-controlling business practice has historically been very successful for network operators, and it remains a well entrenched part of their business thinking. However, times have changed, and it has now become greatly to their advantage to commit fully to the Internet End-to-End model. In this model, the network operator is first and foremost a Layer 3 access provider. The network operator may also be a systems integrator, or an application service provider. But it must do so fairly, and it must fully respect the equal access principle.

The fundamental goal of a network operator is to generate high volumes of premium-revenue traffic through its network. By adopting the open WhiteBerry model and the Internet End-to-End model, network operators can greatly increase both the volume and the revenue-generating value of the traffic they carry. This is because:

- Mobile Messaging consists of inherently high-premium traffic, since its primary value is in delivering urgent messages or time-critical information to the user
- The open WhiteBerry model generates greatly increased volumes of this high-premium traffic
- The WhiteBerry model requires that the underlying network conform fully to the Internet End-to-End model

Inevitably, some network operators will realize this sooner than others, and we can also expect a general reluctance on the part of the operators to change their business models. However, the inherent growth characteristics of the End-to-End model, and competitive forces among the network operators, will ensure its eventual adoption.

Wireless ASP Integration

In the longer term, a variety of other mobility-oriented applications and services will need to be integrated with the Mobile Messaging application. We believe that eventually, all such services will be based on the same sort of open model as WhiteBerry.

We invite Application Service Providers (ASPs) such as AOL, Yahoo and AT&T to participate in the general concept of open-model services—i.e. services which are based on free protocols and open-source software, and which are fully consistent with the Internet End-to-End model. As a starting point, we invite these providers to integrate the appropriate members of the LEAP family of protocols into their suite of Subscriber Services.

An example of open-model services, available for examination and inspiration, is provided by the free ByName Services at `http://www.ByName.net`. The ByName Services are the world's first **Libre Services**, based entirely on patent-free protocols and free software.

Challenge to ASPs

By integrating the LEAP Message Center software into their existing messaging services, Application Service Providers can offer WhiteBerry service immediately. Our challenge to ASPs is this: What possible reason is there to do otherwise? Consider the following:

- Wireless applications are undergoing explosive growth
- Wireless networks are converging on IP
- LEAP-based Message Center software is available as free, open-source software

It is clear that the costs are negligible, while the benefits are huge.

13.8.4 Systems and Solutions Integration

As described in Section 13.5.4, it is already possible to put together a fully functioning WhiteBerry Mobile Messaging solution based on existing products, software and services. However this requires specialized knowledge, and is time-consuming and inconvenient. The typical individual or corporate consumer cannot be expected to do this sort of custom integration.

The typical WhiteBerry consumer will require a complete, fully-integrated, out-of-the-box working solution—like BlackBerry, in fact. Clearly, there is a major systems integration opportunity here.

We therefore invite systems integrators such as Omnisky to assemble all the required components into a multi-platform, turnkey solution for the end user.

As we have previously noted, Omnisky and other service providers have already marketed mobile e-mail solutions based on inadequate land-based Internet protocols, which, not surprisingly, have done extremely poorly in the face of the superior functionality of BlackBerry. By adopting the LEAP-based WhiteBerry model, these companies can deliver true Mobile Messaging functionality which equals or exceeds that of BlackBerry.

Message Center Integration

The Message Center software arena is currently dominated by three major systems: the Unix sendmail system, QMail, and Microsoft Exchange. These are therefore the most important Message Center systems for initial implementation of WhiteBerry, and the LEAP protocols must be integrated as a specific mail submission and delivery component of these systems.

We therefore invite the developers of Message Center software to create and/or enhance integration of LEAP with these systems. We also invite the developers and integrators of complete, customer-premise Message Center solutions to incorporate LEAP into their existing products.

In addition, a major opportunity exists for the integration of the core WhiteBerry capability (i.e. wireless submission and delivery) with a variety of other messaging features and tools such as: rule-based message processing and forwarding, custom mailbox synchronizers, firewall bypasses, and Message Center management tools. By means of these and other integration implementations, Message Center integrators can build a rich and diverse capability on top of WhiteBerry.

13.8.5 How to Participate

Organizations and individuals who wish to participate in Operation WhiteBerry can do so in several ways, and there are a number of resources available to assist such participation. These resources include:

- The LEAP Forum at
  `http://www.LeapForum.org/`
- The EMSD Organization at
  `http://www.emsd.org/`
- The ESRO Organization at
  `http://www.esro.org/`
- The MailMeAnywhere open-source software distribution center and development forum at
  `http://www.MailMeAnywhere.org`.

EMSD.org, ESRO.org and MailMeAnywhere.org are oriented towards engineering development and usage of the LEAP protocols. ESRO.org and EMSD.org are development forums for their respective protocols, while MailMeAnywhere hosts a public forum for the collective development and enhancement of LEAP protocol engines and integration tools.

LeapForum.org is oriented towards coordinating usage of the LEAP protocols from a business development perspective; it provides a means for companies to announce their participation in Operation WhiteBerry, seek out business partners, and coordinate cooperative effort.

All of these forums host a number of mailing lists to enable different forms of participation.

Organizations and individuals who wish to participate in the development and enhancement of the LEAP protocols themselves can do so via the appropriate mailing lists at EMSD.org and ESRO.org.

Those who wish to participate in WhiteBerry from a systems integration point of view will need to determine what components are available, and evaluate their merits. By joining the relevant mailing lists at MailMeAnywhere.org, integrators can advise component suppliers of their requirements and interests; while suppliers can let integrators know about available products and components.

We also welcome and encourage participation by individual or *ad hoc* users such as early self-integrating adopters, individual programmers, or those who wish to contribute written materials. Programmers who wish to enhance the LEAP software or port it to new environments can build on the existing GPL code, then return their enhancements to the base software through MailMeAnywhere.org. Those who are interested in promoting the WhiteBerry model may do so by contributing additional material to *The LEAP Manifesto*, revising existing material, translating the Manifesto into foreign languages, or contributing written materials to other forums.

Getting the LEAP Software

Participants who wish to implement the WhiteBerry model will need to get the necessary LEAP software, and there are three basic options for doing this. First, free open-source software implementations for both devices and message centers is available at MailMeAnywhere.org. This open-source software is distributed under the GNU General Public License (GPL), permitting a broad range of uses of the software. Those wishing to use this free software are invited to join the relevant mailing lists at MailMeAnywhere.org. MailMeAnywhere also provides various other resources to support usage and growth of this free software base.

There are certain situations in which use of the GPL is not appropriate, such as the incorporation of software into cellular phones. Under these circumstances the source code can be licensed from a commercial supplier such as Neda Communications, Inc. (`http://www.neda.com`).

As a third option, WhiteBerry adopters may develop LEAP software on their own, based on the patent-free protocol RFCs. Those wishing to do so are encouraged to join the relevant mailing lists at EMSD.org and ESRO.org.

13.9 Beyond Operation WhiteBerry

In this article we have focussed on LEAP and WhiteBerry in terms of a specific application, namely, interpersonal Mobile Messaging. But this is just the beginning. WhiteBerry is also the starting point for a more general set of wireless messaging applications and usages; and the LEAP protocols are the starting point for a more general set of Internet services.

13.9.1 Building on WhiteBerry

The WhiteBerry model can be augmented and expanded in various ways to create new wireless messaging applications; the following are some examples.

- **Broader forms of usage**. Note that person-to-person messaging is a special case of a more general set of communications scenarios, which also includes person-to-machine, machine-to-person, and machine-to-machine messaging. Existing e-mail applications already encompass these other scenarios; for example, person-to-machine messaging applications include the use of e-mail robots which can be invoked by the user to accomplish various tasks; while machine-to-person messaging applications include various sorts of automatic e-mail alert and notification services.

  All this existing capability can be immediately brought to the wireless world by means of WhiteBerry. In other words, in addition to providing interpersonal Mobile Messaging services, WhiteBerry can also function as a generalized efficient message transfer infrastructure, which generic applications can use to deliver message-related services to the user.

- **Greater richness of content**. The capabilities of WhiteBerry can also be expanded in terms of richness of information content. Though the content of interpersonal messages typically consists of plain text only, this basic information payload can be augmented with more complex forms of content—for example, a variety of MIME attachments have been defined which greatly enhance the utility of conventional e-mail. Similar content-expanding features can readily be brought to the wireless domain by incorporation into the WhiteBerry model.

  Note that the implied message shortness of the WhiteBerry model is not necessarily in conflict with richness of content. This simply means that such content must be appropriate to the wireless and mobile domain—it must bring added value to the user, without violating message size restrictions. An example of such content is the use of embedded responses, in which the message originator provides the recipient with a set of pre-defined, multiple-choice replies from which to select [**?**].

- **Expanded usage patterns**. E-mail has traditionally been considered an asynchronous activity, and unsuitable for real-time or near-real-time usage. However, since a mobile user is available for communication to a far greater extent than a desktop user, and since the efficiency of the LEAP protocols permits very rapid message exchange, various interactive applications can be built on top of WhiteBerry.

  A good example is Instant Messaging. Note that in this application the exchanged messages are by nature very short, and therefore the efficiency of EMSD brings a significant performance premium. The LEAP protocols can readily be used to extend existing Instant Messaging capabilities to mobile devices. This extension of the WhiteBerry model requires only simple integration with existing Instant Messaging solutions, and with existing facilities such as Presence Management and Location Management.

- **Location-based applications**. The availability of mobile device location information enables numerous location-based applications such as taxi dispatch, driving directions, local weather conditions, availability of local products & services, emergency assistance, and many more. Location information can also be integrated with Mobile Messaging services to provide the user with a powerful set of new applications. For a comprehensive and imaginative survey of the location-based services domain, see WaveMarket's white paper entitled *Location, Presence and Event-Driven Mobile Applications*, available on the WaveMarket website at `http://www.wavemarket.com/`. Many of the applications described in that document can be built on top of WhiteBerry.

13.9.2 Building on LEAP

It is also worth noting that the LEAP family of protocols has broader scope than just WhiteBerry messaging. WhiteBerry is part of a larger family of LEAP-based applications, which also includes other applications such as LEAP-based web applications. These applications can be tightly integrated, so as to provide a consistent package of capabilities to the user.

Note in particular that use of ESRO is not limited to WhiteBerry. ESRO is a generic enabling foundation, upon which a variety of efficient applications can be built. For more information on generalized uses of ESRO, refer to the Manifesto article *ESRO: A Foundation for the Development of Efficient Protocols*.

In Operation WhiteBerry we have established a framework that can continue to be built upon to bring tremendous benefits to the wireless industry and the consumer.

13.10 Summary

BlackBerry has conclusively established the validity of the Mobile Messaging value proposition. The popularity of BlackBerry clearly demonstrates the large user demand for this type of service, and the correspondingly large market opportunity. There can no longer be any doubt about the future growth and importance of the Mobile Messaging industry.

The Mobile Messaging industry of today is populated by a number of systems which are closed, which do not interoperate, and which violate the Internet End-to-End principle. In the long run this situation is untenable. Sooner or later, the industry must adopt a single set of open protocols that guarantee industry-wide interoperability. This will enable market entry and competition across the entire Mobile Messaging value chain. The result will be better solutions for the consumer, and enormous market rewards for the providers of those solutions.

The opportunity to do this now exists. The three key technological components are all in place:

1. A variety of suitable unconscious carry devices are available, and at the right price points
2. Suitable IP-based wireless networks with adequate coverage are in place, and there is a continuing convergence of all networks towards IP
3. Wireless modems are available for a variety of networks, and with adequate battery life

The final enabling component is a set of truly open and free protocols; and this last piece is now also available in the form of LEAP.

Given that everything is in place, there remains only one obstacle to the open Mobile Messaging industry: the will of the industry itself. Operation WhiteBerry is no longer a matter of technological development. It is now a matter of consensus, collaboration and integration.

There is nothing in this article that is remotely theoretical or speculative. All the technological components for WhiteBerry are real and operational *now*. Any data communications professional can acquire the necessary components, and begin enjoying his fully functional WhiteBerry mobile messaging system *today*. A systems integrator or industry alliance can create a fully integrated open equivalent to BlackBerry and bring it to market *within 90 days*.

RIM has demonstrated the enormous market potential of Mobile Messaging, and is reaping the profits. Meanwhile, the entire community of potentially competing device manufacturers, modem manufacturers, network service providers, Message Center operators, and systems integrators stands idly by, gazing blankly off into space.

At the time of writing in November 2002, the state of the Mobile Messaging industry resembles nothing so much as a surreal game of musical chairs, in which nobody seems to have realized that the music has stopped. The chairs are all sitting there, empty. There ought to be a mad scramble to sit in them. Yet inexplicably, there is not.

But sooner or later, there certainly will be. And those industry players that realize this soonest, and stake their claims quickest, will be the ones that enjoy the most commanding first-mover advantage.

Acknowledgments

We gratefully acknowledge the assistance of the following persons in the preparation and review of this document: Payman Arabshahi, German Burtscher, Matt Champagne, John Coffey, Murat Divringi, Tom Huseby, Bahman Khamneian, Pean Lim, Derrell Lipman, Mike Marks, Daryoush Mehrtash, Mark Taylor, and Pinneke Tjandana.

Part III

Making LEAP Widespread

Chapter 14

Strategy for Making LEAP Widespread

14.1 Introduction

Thus far in *The LEAP Manifesto* our discussion has been largely theoretical; in previous articles we have demonstrated on paper that WAP is not viable [?], and that LEAP has all the characteristics necessary to be considered a viable alternative [?]. However this is all academic until the protocols are implemented as software and deployed in real world systems. In this article we describe our strategy for accomplishing this goal.

The key elements of our strategy are that (1) we have created software implementations of the protocols for most common platforms, and (2) this software will be made publicly available in the form of free software in open-source format.

In order for the LEAP protocols to become widely used, they must be implemented in the form of software solutions that are readily available for deployment by end-users. To this end, protocol engines have been implemented in the form of portable code which has been ported to a variety of platforms. On the device side, software has been implemented for pagers and cell-phones; for hand-held PCs and Palm Pilot (Palm OS, Windows CE, Palm PC); for Windows 98, Windows 95, and Windows NT; and for Pine (UNIX, Windows, DOS). On the message center side, software has been implemented for Solaris, Linux and NT.

14.2 The Power of Free Software

The enormous power of free software and Open-Source Software (OSS) has been well demonstrated by the tremendous success of such OSS systems as Linux, Imap, Sendmail, and various others. At this point it seems clear that creating a rich and powerful software system, then providing it as patent-free, free software, in open-source form, and at no cost to the user, is an extremely effective mechanism for encouraging its adoption and usage.

Our strategy for making LEAP widespread is based on the power of the Open-Source Software model. By making the base implementation available as free software, we are enabling anyone to expand or enhance it, or make it available on any open platform. This model reduces the end cost to the ultimate user, and frees the user from the restrictions of traditional commercial software.

For further discussion and information about the Open-Source Software model, see article [?]. Also, a particularly revealing analysis of OSS is provided by Microsoft's infamous Halloween documents, [?], [?]. These leaked documents provide embarrassingly revealing insights into Microsoft's viewpoint and attitude towards the threat posed by OSS.

On the face of it, the OSS model would appear to have little commercial application – giving away free software may be an unbeatable way of promoting its usage, but it would not appear to be a useful business strategy. However, four basic business models have been proposed based on OSS:

1. **Provide Secondary Services**. The vendor/developer of the OSS system makes money by means of service contracts, customer integration, etc.

2. **Loss Leader for Market Entry**. The vendor/developer of the OSS system uses the advantages of the OSS process (especially that of credibility) as a competitive advantage against established commercial vendors.

3. **Commoditizing Downstream Suppliers**. The vendor/developer of the OSS system is also the producer of a product or service further in the value supply chain, and closer to the consumer.

4. **Standards Preemption**. Because the OSS process can be argued to be a winner-take-all process, it may be to the advantage of the vendor/developer to seed the OSS market with its own codebase, so as to preempt a competitive codebase from taking hold.

These four models represent the current status of business thinking, regarding the possible business uses of the OSS model. All four of these business models are valid, and each represents a practical, workable business strategy.

14.2.1 Irrelevance of the Supply Chain Model

According to conventional supply chain analysis, a successful business model must satisfy the financial and other needs of all potential gatekeepers throughout the supply chain. The OSS model, however, renders the conventional supply chain model irrelevant.

If the protocol-based Mobile Messaging industry were to be looked at from the traditional supply chain point of view, the picture would look something like Figure 14.1. This figure shows the supply chain relationships among all the major players in the Mobile Messaging industry. According to the conventional wisdom, in order for a business model to work, it is necessary (though not sufficient) for all the companies in the chain to make profits and satisfy their non-financial needs. In the protocols-based model, the LEAP protocols, shown at the bottom of the figure, are the basic enabling technology for the entire industry. The Consumers, shown at the top of the figure, are the ultimate end-users of the industry products and services. In general, products and services flow from the bottom of the figure towards the top, while revenues flow from the top down towards the bottom.

Some of the key Mobile Messaging industry constituencies are the Modem Manufacturers (e.g. Novatel), the PDA Manufacturers (e.g. Palm), and the Network Service Providers (e.g. AT&T). Integrators such as OmniSky, who are bundling fully integrated solutions and focusing on direct Layer 7 relationships with the consumer, are best positioned to exploit the increasing returns opportunities of the Virtual Communities phenomenon [**?**].

Much of the success of WAP can be attributed to the WAP Forum's masterful manipulation of the traditional supply chain model, and to their winning the support of all the relevant gate-keepers. The creation of the WAP Forum, and their accomodation of all the significant network operators, is the classical business execution of what is needed to make the supply chain work in one's favor.

The OSS model, however, is completely different. The OSS model bypasses the traditional gatekeeper roles altogether, thereby rendering them irrelevant. Those gatekeepers who do not recognize that the End-to-End Internet model will make their gatekeeper role obsolete will sooner or later find themselves out in the cold. All four of the OSS models described in the previous section will fullfil their respective purposes. In addition to the OSS model, the Internet End-to-End model, which eliminates the WAP-like gateway model, is also in favor of LEAP.

In the telecommunications industry, the accomodation of gate-keepers and supply chain models have been traditional requirements of a successful strategy. However, we do not consider that tradition relevant in the context of LEAP. Instead, we consider that our main challenge will be to fully understand and direct the power of the Internet End-to-End and OSS models.

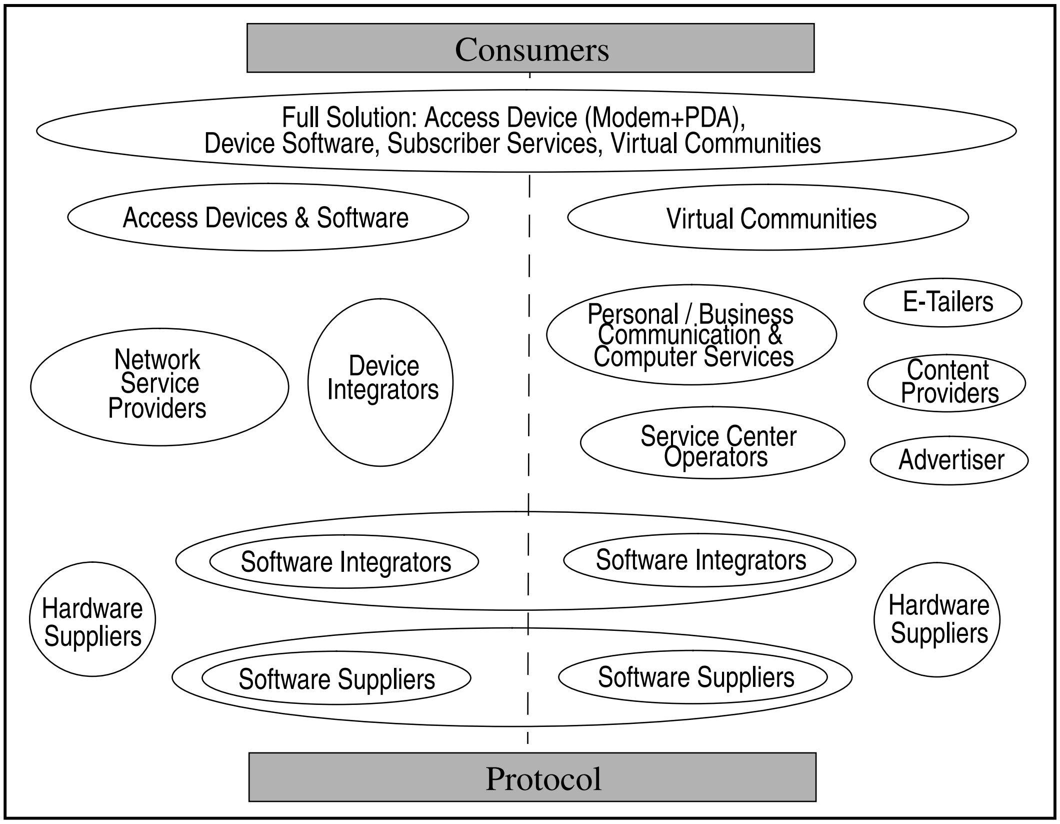


Figure 14.1: Traditional Supply Chain Model

14.2.2 Bypassing the Telecommunications Gatekeepers

The strength and effectiveness of the Mobile Messaging industry requires cooperation and understanding among several players, including mobile device manufacturers, wireless network operators, ISPs, and software providers.

Two of the key players are the telephone companies and the wireless network operators. We will refer to these two players collectively as the Telecommunications Industry. For largely historical reasons, the Telecommunications Industry tends to have a different culture and set of attitudes than the broader Data Communications Industry.

Most of today's closed wireless data networks are monopolies, or have their roots as monopolies. Also, the operation of these wireless networks is similar to the operation of phone companies. The type of fitness resulting from competition which the Data Communications and ISP market enjoys has been absent from today's wireless data networks. Many key technical decisions in the wireless data networks are likely to continue being made by unfit wireless data operators.

This represents a potential obstacle to the efficient development of Mobile Messaging, and to some extent can be expected to retard its development.

However, the power of the Internet end-to-end model and OSS will ultimately overwhelm all resistance. The trend towards open networks is clear, inevitable, and unstoppable; and any gatekeepers who resist this trend will eventually be eliminated.

14.3 How LEAP Will Become Widespread

It is in the nature of LEAP to grow organically and become widespread. Briefly, these protocols will succeed and become widespread because:

- They meet the technical requirements of the industry.
- A set of protocols which do what LEAP does are desperately needed.
- Paradoxical though this may seem, there are no credible competing protocols at present.
- They are truly open and patent-free. Any company, organization, or individual may implement and use the protocols without incurring licensing or any other fees. Nobody, including Neda, may earn revenue by virtue of ownership of these protocols.
- Open-source implementations of the protocols are freely available. They are available because Neda has written them, and is giving them away in the form of open-source software.

All the technological and software assets required to promote the widespread usage of LEAP are complete and in place. These assets consist of:

- The LEAP protocols themselves
- The patent-freedom of the protocols
- Open-source software implementations of the protocols for all major platforms and end-user devices
- Free subscriber services to support initial deployment of the protocols in end-user devices

We will initiate the propagation of LEAP by making industry-wide exposure of all the assets we have created for this purpose. We will publish and distribute the relevant documents (*The WAP Trap*, *The LEAP Manifesto* etc.), and we will make all our software implementations available as open-source. This will be sufficient to initiate the organic growth of LEAP described above.

In addition to this organic growth, Neda will also participate in active promotion of the LEAP protocols, by actively marketing message center software and systems licenses, and LEAP technology licenses. We expect that as usage of the LEAP protocols increases, other companies will quickly begin to compete with Neda on this basis. From the point of view of making LEAP widespread, this competition is greatly to be desired, since this will serve to accelerate the adoption of LEAP within the industry.

14.4 NEDA's Free Software Base

To build large multi-platform, scalable and industrial strength software, a solid development foundation is required. The following large pieces of software provide the foundation (consistency, portability, managability, scalability, ease-of configuration etc.) for Neda's products and services.

Figure 14.2 provides a high-level overview of Neda's software components and architecture. A brief description of the key software components is provided below.

Open C Platform (OCP): OCP is an efficient virtualization layer that provides the basic support for all of Neda's developed products. OCP presently supports 8 distinct target platforms and a large number of development environments. OCP is freely and openly available, and is very well documented in [**?**].

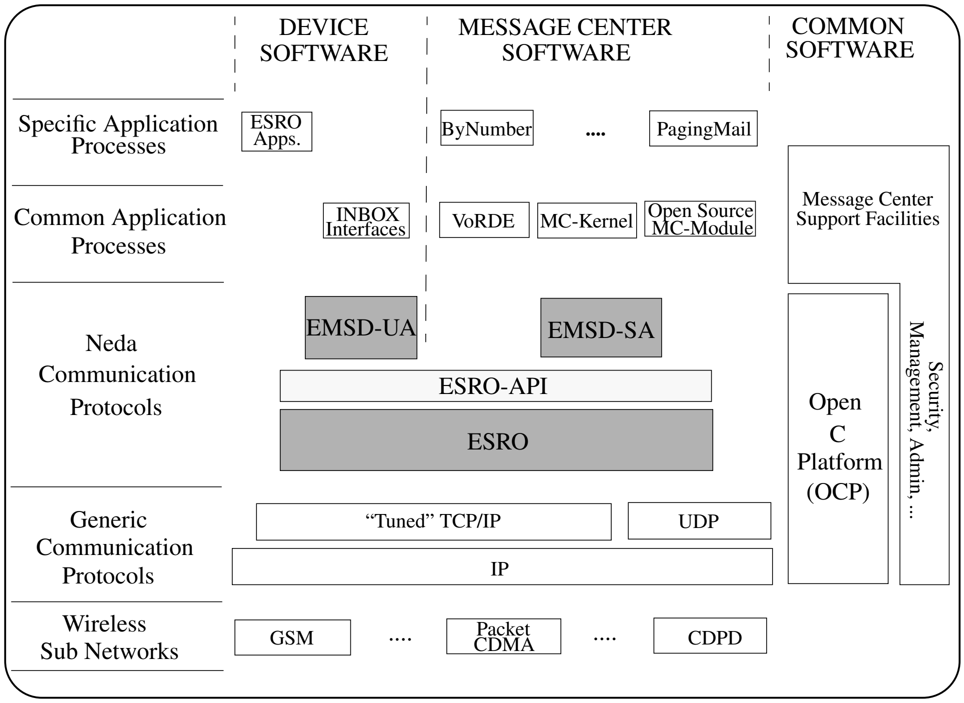


Figure 14.2: Neda Software Architecture

All of Neda's products use OCP and conform to its style and conventions.

To understand Neda's software development style, quality and philosophy, we recommend a review of OCP [**?**]. At Neda we take great pride in our work. OCP is free, open and publicly available. We invite interested readers to check it out.

ESRO Protocol Engines: The ESRO protocol engines are highly portable and are shared across a large number of platforms.

The ESRO protocol engines are based on OCP, are fully documented, and have been very extensively tested.

For additional information about the ESRO Protocol Engines, please refer to [**?**], [**?**], [**?**].

EMSD Protocol Engines: All of our EMSD products use the EMSD protocol engines. The EMSD protocol engines are highly portable and are shared across a large number of platforms.

All EMSD protocol engines are based on OCP, are fully documented, and are very extensively tested. A particular packaging of the EMSD Protocol Engines is one of our technology products.

For additional information about EMSD Protocol Engines, please refer to [**?**], [**?**], [**?**].

Open-Source Message Center Architecture: Over the years we have invested a great deal of time in the use of open-source software modules. In particular, we have made significant investments in the following large pieces of software: Qmail, Sendmail, Bind, IMAP C-Client, and Hylafax.

We have also invested greatly in Neda-developed utilities and tools that bring these individual software components under a consistent umbrella for administration, management, accounting and monitoring.

Voice Response Development Environment (VoRDE): VoRDE is the basis for all of Neda's Interactive Voice Response (IVR) systems and services.

For additional information about VoRDE, please refer to [**?**], [**?**].

Our basic goal is to make the above software heavily used and widespread.

For generic platforms, we will make the software available as free software through the http://www.MailMeAnywhere.org/ software distribution center. The challenge here is to create a large virtual community of developers who use this software and further enhance it. On the device side, device software will be made available on most generic platforms including PalmOS, EPOC, WindowsCE, Windows9X, and Windows-NT. On the Server side, software will be available on Solaris, Linux and NT.

In addition to generic platforms, we will also make the software available in various embedded environments. See [**?**] for a list of Neda licensees.

14.5 Neda's Free Software Licensing Strategy

Neda's free software is available based on a set of free software licenses. Neda's open-source software is usually subject to either the GNU General Public Licenese (GPL) or GNU Library General Public License (LGPL) [**?**].

Neda chose GPL and LGPL as its free software licensing method after considering many other licensing models including: Mozzila Public License, Qt Free Edition, and BSD License. We chose GPL because of its track record, and because it is well understood and accepted within the software industry.

In addition to the GPL and LGPL licenses, Neda software is also available under Neda Professional Software Licenses, for those environments and usages where the GPL is too restrictive.

Almost all of Neda's software is available as open-source through http://www.mailmeanywhere.org/. However that does not mean that Neda software can be used by all and in all environments. The GNU General Public License prohibits commercial use of the software and its integration with commercial software and prohibits its use in closed, embedded environments.

Chapter 15

EMSD on Windows CE

Preface

This article was originally published in March 1997. It is being included in the Manifesto, essentially unchanged from its original form.

15.1 Summary

Windows CE (WinCE) is a capable, general purpose mobile computing platform which, among other applications, comes bundled with an extensible e-mail application named "Inbox."

CDPD, the premier wireless, mobile network with full Internet connectivity has reached critical mass with its large scale deployment and CDPD wireless modems becoming available in ever-shrinking form factors (PCMCIA) with improving power consumption and other characteristics.

Interpersonal messaging is the most important and proven value proposition of the wireless medium. Existing Internet e-mail protocols have built-in assumptions, which makes them sub-optimal for use over wireless networks. The EMSD protocol fills this gap, allowing fully open, extensible (MIME) Internet e-mail connectivity over wireless networks.

Given that the fundamental components outlined above (WinCE, CDPD, EMSD) are real products and technologies, integrating them to create an end-user product is within reach. If the integration is performed properly, the availability of wireless E-mail capability can become straightforward for the end users of this system. Any Windows CE user can purchase a "mobile e-mail kit," which includes a CDPD modem, CDPD account activation and the necessary software add-ons to add the EMSD capability to the palmtop. In fact, there are several channel models which can be used to deliver the wireless messaging capability to the end user, so forming a "mobile e-mail kit" is just one of them. This paper focuses on the technical issues of the integration effort at hand as opposed to marketing and channels issues. A WinCE based EMSD/CDPD wireless messaging solution, when coupled with the back-end EMSD Message Transfer Agent (MTA) offers a strong value proposition with minimal infrastructure. The EMSD MTA can be running at a central server at the CDPD Network Operator site (example mail address: john.doe@emsd.cdpdProvider.net) or as a Customer Premise Equipment (CPE) at the Internet ISP or IS Department of an Organization (example mail address: john.doe@emsd.boeing.com).

15.2 About This Document

This paper is available in many formats. You may wish to obtain a copy of this document in a more appropriate format before proceeding further. HTML, PDF, Postscript and plain text versions of this paper can be found at http://www.emsd.org/documents/whitePapers.html.

This paper is one of a series of white papers that introduces the ”world” of Efficient Mail Submission & Delivery (EMSD).

If you are not familiar with EMSD General, you may wish to read the ”Introduction to EMSD - White Paper” [**?**].

15.3 Background

15.3.1 Components involved

Wireless Network

In the context of this paper, any wireless, IP based network. Several networks such as the pACT network under development by AT&T Wireless Services, as well as the widely deployed CDPD network fit this description. Also applicable are a number of other proprietary wireless networks which expose an IP interface, such as Ricochet wireless IP network from Metricom, Inc.

CDPD is a wireless, mobile two-way data network offering coverage footprint equivalent to cellular voice in many markets. CDPD exposes an IP interface and a fixed, ”real” IP address to each subscriber node (End System) as assigned by the CDPD Network Service Provider at the time of end user provisioning. CDPD mobile nodes enjoy full mobile connectivity using the same IP address throughout the entire CDPD national coverage area due to inherent support of mobility built into CDPD.

EMSD

EMSD is an open, extensible and efficient message submission and delivery protocol designed specifically for the wireless network. It minimizes the network traffic required to send and receive messages, thus producing a messaging protocol that meets the needs of the mobile communicator. Fewer and smaller packets means extended battery life, efficient use of carrier bandwidth and support for marginal coverage areas. EMSD is an open specification that is an extension of the existing messaging world.

EMSD is up to 5 times more efficient than SMTP both in terms of the number of packets transmitted and in terms of number of bytes transmitted (see Efficiency Study of EMSD vs. SMTP/POP3/IMAP (PDF ¡61 KB)¿ (Postscript ¡165 KB¿)

EMSD protocols are openly available through online RFC Library (ftp://ftp.isi.edu/in-notes/rfc2524.txt)

As with other open standards like SMTP, POP, etc., multiple implementations of EMSD are available and various development tools and products are emerging. Being the premier developer of the EMSD Protocols and Implementations, Neda Communications, Inc. offers several products for the mobile device manufacturers, network operators and corporate IS organizations.

Windows CE

Microsoft Windows CE is an open, scalable Windows operating system platform for a broad range of communications, entertainment and mobile-computing devices. Unlike previous PDA offerings from various vendors, Windows CE

Functions Protocols	SMTP	IMAP	POP	EMSD
Submission	XXX			XXX
Delivery	XXX			XXX
Relay	XXX			
Retrieval		XXX	XXX	XX
Mailbox Access		XXX	X	
Mailbox Sync.		XXX		

Table 15.1: Messaging Protocols vs. Supported Functions

enjoys multiple implementations from various hardware vendors, a rich development environment, which leverages the large community of windows developers, and a core set of ”Windows Companion” applications. These ”Companion Apps” make WinCE useful from the get-go not as a stand-alone PDA but as an extension of the office environment.

Built-in applications come in two categories: (1) Personal Productivity Apps: These include the compact yet capable ”Pocket Word”, ”Pocket Excel” and the Calendar/Contact/Task manager apps. (2) Communications/Networking Apps: These include the capable ”Pocket Internet Explorer” and the extensible ”Inbox” e-mail front end.

All of these applications combine to provide an environment which feels ”intuitive” for Windows9x and NT4.x users.

Messaging protocols

Various Internet Messaging Protocols are mentioned throughout this paper. Although all are used for messaging purposes, functions supported by these protocols do not necessarily match but in many cases complement each other. Given that these protocols came about in an evolutionary fashion over time, this makes sense. The following table illustrates this point.

In Table 15.1, the number of ”X”es in each box denote to what extent a particular function is supported by a particular protocol.

It is important to note that the scope of EMSD protocols was deliberately limited to the primary requirement for mobile messaging which is: ”submission and delivery of time critical important messages.” EMSD is designed to complement existing mailbox access protocols such as IMAP.

Although there are proprietary implementations of messaging protocols offered by various vendors over various wireless networks, EMSD is the only open efficient messaging protocol available. The openness of EMSD is a key attribute to help expand the wireless messaging, just as SMTP and POP helped to establish and expand the (now large) Internet e-mail industry years ago.

15.4 CDPD, EMSD and Windows CE: High Level Architecture

15.4.1 EMSD and WinCE Messaging

WinCE OS architecture supports a ”local message store” which can be used by any application to send/retrieve messages through a set of defined APIs.

Each message in the local message store has many attributes associated with it, such as the ”Mail Folder” it belongs to (Inbox, Sent Items, ...), the ”Mail Transport Service” responsible for handling the message (SMTP/POP3, Fax, EMSD,...), and other miscellaneous attributes.

WinCE OS architecture also supports the concept of a ”Mail Transport Service Provider” which works as an abstraction layer and handles sending/receiving of messages marked as handled by that ”Mail Transport Service.”

The ”MSInetEmail” Mail Transport Service Provider, which implements the POP3/SMTP message retrieval/submission protocols, is included with the base WinCE system. Consequently, for users with mailboxes on ISPs, Intranets, etc., WinCE offers a plug-and-play e-mail solution over dial-up IP connections.

The ”Inbox” application allows the user to choose which Mail Transport Service should be selected as the default one at any given time. In the WinCE architecture, the EMSD software would be just another ”Mail Transport Service” next to ”SMTP/POP3”, ”IMAP”, ”Fax” and so on.

EMSD is layered on top of UDP datagram service offered by the IP stack and as such, is oblivious to the how the IP connectivity is achieved. In other words, whether a direct serial link, a wireline modem or a CDPD modem is used is transparent to the EMSD layer. In fact, much of EMSD development and testing is done over LAN and direct serial links by simply adjusting a few tunable parameters.

The following block diagram illustrates the components involved as well as the layering of services.

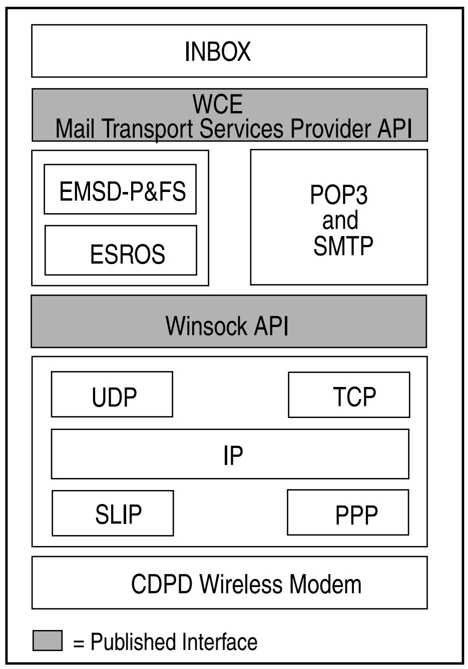


Figure 15.1: Components of WCE Mail Transport Service Provider with EMSD

As illustrated in Figure 15.1, upper and lower interfaces used by the EMSD user agent correspond to fully published WinCE APIs defined in WinCE development environment (SDK). It is also visible that the EMSD Mail Service Transport Provider is a peer to the Microsoft provided MSInetMail Mail Service Transport Provider. The WinCE published APIs enable any third party to develop a Mail Service Transport Provider.

15.4.2 WinCE and CDPD Modem integration

Today, all CDPD modems, including the PCMCIA types, expose a ”SLIP” interface on the serial port they are associated with. Some newer CDPD modems have started supporting the ”PPP” interface as well. Manufacturers of existing modems are also developing new firmware to add PPP support to their modems. WinCE comes with a built-in

IP stack and offers Winsock v1.1 subset functionality. Today, off-the-shelf, WinCE supports only ”PPP” over serial links, however add-on drivers for SLIP support are appearing from third parties.

For those CDPD modems supporting PPP, WinCE integration is straightforward. As far as WinCE is concerned, there is no difference between connecting to the CDPD network via that CDPD modem and connecting to an Internet ISP using a standard PCMCIA wireline modem: the same PPP interface is used.

15.4.3 EMSD Message Transfer Service and Back End Mailbox Issues

There are many scenarios in which EMSD can be used to provide messaging services, but for the purposes of this paper, we will consider only the scenario where the following are true:

- Target end user already has an e-mail mailbox.
- The mailbox mentioned is accessible via POP3/SMTP
- The end user would like to be able to access this mailbox via EMSD

In all cases, the ”EMSD Mail Transport Service Provider” in the WinCE will communicate directly with an EMSD Message Transfer Agent (MTA) which functions as a gateway between EMSD and SMTP. In other words, messages exchanged between the WinCE and the EMSD MTA are in the efficient EMSD format (in the airlink where EMSD’s attributes are needed). The EMSD MTA handles translating EMSD format messages to the Internet format (RFC-822) and sending them on, or translating incoming Internet messages destined at the EMSD WinCE node to EMSD format and handing them over using EMSD protocols.

The EMSD MTA can reside anywhere on the Internet, including the CDPD service provider, at an ISP or as Customer Premise Equipment (CPE) at the customer site. Because EMSD MTAs maintain their own subscriber base, all of the above schemes can be deployed simultaneously. For example, The Neda Customer Premise Message Center (http://www.neda.com/products/ETWP/DataSheets/etwp-cpmc-solaris.fm5.pdf) was designed as a CPE MTA.

When a user is going mobile, he/she can initiate forwarding of all or certain classes of messages (for example all URGENT messages) to the assigned EMSD address. A mailbox sorter scheme similar to the ”Procmail” utility fits well into this model.

If the system where the user’s mailbox resides is EMSD-aware or is the node where the EMSD-MTA entity is running, then a variety of features can be included. An example would be setting up the attributes of EMSD forwarding on the fly and delivering a message to a user only once...

All messages originated from the mobile WinCE unit would simply be sent via EMSD to the EMSD-MTA without any secondary filter processing. Filter processing at the user mailbox level is relevant only for delivery of messages coming from the Internet destined to the EMSD user.

15.5 Windows CE Inbox integration with EMSD

Once the EMSD Service Provider for WinCE Inbox is installed, all that is needed to be done is:

1. Establishing the IP network connectivity by using the appropriate ”dial-up networking” setup
2. Choosing ”EMSD” in the ”Service-¿Connect” from the Inbox menus, which are standard operations for Inbox.

15.6 End User Experience

15.6.1 Assumptions

- User is already familiar with Win95, e-mail concepts
- User has a mailbox on an Internet host, accessible via SMTP.
- User has acquired a WinCE unit.

15.6.2 Acquisition

- The same channel user purchased the WinCE from has ”CDPD Wireless e-mail kits.”
- Multiple kits can be set up by working with various CDPD modem manufacturers.
- Each kit would include modem, CDPD activation info and the EMSD add-on software.
- Alternatively (or as an extra service) the CDPD activation info and the EMSD add-on software can be made available on-line over the Internet.

15.6.3 Installation

1. Activate the CDPD modem.
2. Verify CDPD network connectivity via the activated modem using provided program.
3. Install the EMSD Mail Service Provider
4. Verify the EMSD setup using provided program.
5. Start forwarding of urgent (or other mailbox filter criteria) messages to EMSD at the host maintaining the user’s primary mailbox.
6. Use the standard WinCE Inbox program with the EMSD Mail Service Provider.

15.7 Conclusions

Integrating true Internet e-mail connectivity to the WinCE platform using EMSD over CDPD is very feasible and is a natural fit to the WinCE’s ”Companion” model.

Chapter 16

LEAP on Palm OS

16.1 Introduction

This is one of a series of articles that describes the implementation and integration issues involved in incorporating the LEAP protocols into particular PDA environments.

The starting point for incorporation of LEAP in PDAs consists of the Mobile Messaging application, and is based on the Efficient Mail Submission and Delivery (EMSD) protocol [**?**]. EMSD is the e-mail component of the LEAP family of protocols [**?**].

A complete description of how EMSD provides everything necessary to enable end-users to benefit from true end-to-end open mobile messaging based on patent-free protocols and open source and free software is provided in the article *Operation Whiteberry* [**?**]. The present article is part of the more general Operation WhiteBerry model. Before reading this article, the reader is strongly encouraged to read *Operation Whiteberry* so that he/she has a clear understanding of the general implementation framework.

It is our goal to make LEAP widespread on all PDAs. However, the incorporation of LEAP into each platform follows a particular approach and strategy. Each of the articles in this series outlines our strategy for a specific platform.

Windows CE Integration Strategy

A standard mail application called Inbox is bundled with all Windows CE devices. Microsoft has defined a generic mail transport service provider interface underneath Inbox, which allows alternative mail submission and delivery protocols to be integrated with Inbox. Microsoft has also defined a winsock interface which provides access to UDP[**?**] and TCP[**?**] by third party applications. These well-defined APIs allow a new mail submission and delivery protocol to be integrated with Inbox in the Windows CE environment.

Our strategy for integration of EMSD in the Windows CE environment has been to provide a complete Mail Transport Service provider underneath Inbox. We have done so by providing EMSD binary distribution packages which end-users can quickly and easily plug in, allowing immediate implementation of the WhiteBerry model.

Additionally, the entire source code package for EMSD on Windows CE is available subject to the Gnu General Public License (GPL). The availability of this software package in source form allows tight integration with wireless modems, and other customizations/enhancements.

The existing open-source implementation is available at http://www.mailmeanywhere.org.

16.1.1 LEAP on Open-Source PDAs

It is highly desirable that there exist complete open-source and free software platforms as alternatives to the commercial PDA platforms such as Windows CE and Palm OS.

At the present time two such open-source platforms are “Linux Based PDAs” and “eCos.” Recently there have been numerous announcements of new Embedded Linux support for PDAs and other handheld personal computing devices. Various suitable eCos platforms are also now in place.

See the article *LEAP on Linux PDAs* [**?**] for more details.

16.1.2 Palm OS Integration Strategy

Palm OS is an effective, general purpose mobile computing platform which, among other applications, comes bundled with a simple e-mail application called Mail. In contrast to the Windows CE Inbox application, however, the standard Palm OS mail application has only a very rudimentary set of features and capabilities. Also in contrast to Windows CE, the Palm OS does not include a set of well defined APIs to facilitate the integration of alternative mail protocols.

To complicate matters further, various third party mail user agents have now become widely available for Palm OS (largely because of the limitations of the standard Palm OS mail application), with a variety of different user interfaces. Because of this multiplicity of incompatible Palm OS mail applications, each integration of LEAP with Palm OS must be specific to a particular Palm OS mail user agent.

For these reasons our strategy for integration of the EMSD protocol engine in the Palm OS environment is different from that for Windows CE.

We start with a very basic example implementation of EMSD for Palm OS. This example implementation is specific to the standard mail application, based on file access.

Based on the availability of this package in source form, subject to the Gnu General Public License, a variety of Palm mail user agents can become EMSD-enabled very rapidly. To accomodate this further we have selected certain mail user agents which we consider particularly desirable, and we will participate in the integration of EMSD with these mail user agents.

16.2 Palm OS Mail User Agents

There are two basic alternative approaches to developing an e-mail application in the Palm OS environment. These are:

1. Separate Mail User Interface and Mail Transport Service
2. Integrated Mail User Interface and Mail Transport Service

Our goal in this article is to accommodate incorporation of EMSD as a Mail Transport Service provider in both models.

16.2.1 Separate Mail User Interface and Mail Transport Service

In the separate mail user interface and mail transport service model, the application is split into two parts. One part is responsible for transferring the mail message, while the other part is responsible for processing and displaying the message.

An example of this approach is the standard Palm Mail application. The basic distribution of Palm OS lacks mail transport services such as SMTP[?], POP[?], or IMAP[?]. It normally depends on its desktop synchronization capabilities to accomplish the actual sending and receiving. Messages written by the user are put into a queue; then upon synchronization with the desktop, the desktop counterpart sends the queued messages and also retrieves received messages.

However there are add-on packages, such as Top Gun Postman [?] that provide SMTP and POP/IMAP support within the Palm OS device. This enables users of the standard Mail application to send and receive messages directly from their devices via a wireless modem.

Incorporation of EMSD into packages such as Top Gun Postman can be rapidly accomplished as an add-on or replacement for SMTP/POP/IMAP. Figure 16.1 shows the components involved as well as the layering of services in this model.

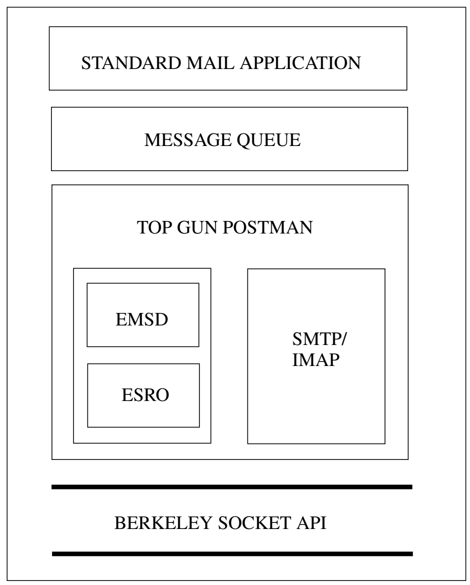


Figure 16.1: Example of Separate Mail Transfer Service for Palm OS

For example, the incorporation of EMSD support with Top Gun Postman can be very quick and easy because of the availaibility of EMSD protocol engines as a Palm OS library.

16.2.2 Integrated Mail User Interface and Mail Transport Service

In the integrated mail user interface and mail transport service model, the application is entirely self-contained.

There are numerous examples of this approach, since it offers greater convenience of operation and usage in the Palm OS environment. In particular MsgAgent [?] and DoodleMail [?] are very popular clients.

Figure 16.2 shows the components involved and the layering of services.

For example, the incorporation of EMSD support with MsgAgent can be very quick and easy because of the availaibility of EMSD protocol engines as a Palm OS library.

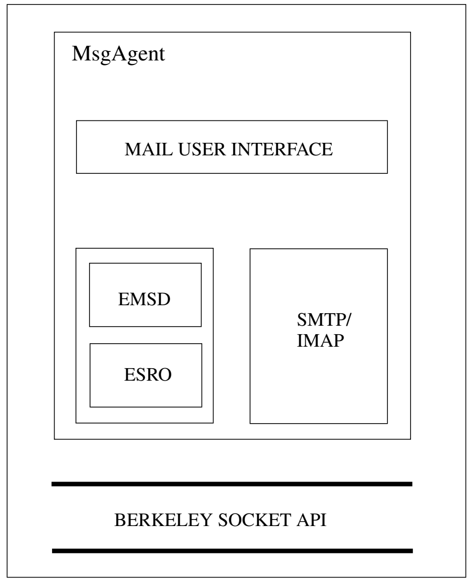


Figure 16.2: Example of Combined Mail Transfer Service for Palm OS

16.3 Invitation to Participate

As described in *Operation WhiteBerry*, the incorporation of EMSD in Palm OS mail packages presents enormous benefits.

We invite the developers of Palm OS mail application packages to incorporate EMSD into their software. EMSD protocol engines ready for the Palm OS environment are readily available, and can be easily integrated with your software. Complete implementations of EMSD in open-source form are available at http://www.mailmeanywhere.org.

Chapter 17

LEAP on Linux PDAs

17.1 Introduction

This is one of a series of articles that describes the implementation and integration issues involved in incorporating the LEAP protocols into particular PDA environments.

The starting point for incorporation of LEAP in PDAs consists of the Mobile Messaging application, and is based on the Efficient Mail Submission and Delivery (EMSD) protocol [**?**]. EMSD is the e-mail component of the LEAP family of protocols [**?**].

A complete description of how EMSD provides everything necessary to enable end-users to benefit from true end-to-end open mobile messaging based on patent-free protocols and open source and free software is provided in the article *Operation Whiteberry* [**?**]. The present article is part of the more general Operation WhiteBerry model. Before reading this article, the reader is strongly encouraged to read *Operation Whiteberry* so that he/she has a clear understanding of the general implementation framework.

It is our goal to make LEAP widespread on all PDAs. However, the incorporation of LEAP into each platform follows a particular approach and strategy. Each of the articles in this series outlines our strategy for a specific platform.

Windows CE Integration Strategy

A standard mail application called Inbox is bundled with all Windows CE devices. Microsoft has defined a generic mail transport service provider interface underneath Inbox, which allows alternative mail submission and delivery protocols to be integrated with Inbox. Microsoft has also defined a winsock interface which provides access to UDP[**?**] and TCP[**?**] by third party applications. These well-defined APIs allow a new mail submission and delivery protocol to be integrated with Inbox in the Windows CE environment.

Our strategy for integration of EMSD in the Windows CE environment has been to provide a complete Mail Transport Service provider underneath Inbox. We have done so by providing EMSD binary distribution packages which end-users can quickly and easily plug in, allowing immediate implementation of the WhiteBerry model.

Additionally, the entire source code package for EMSD on Windows CE is available subject to the Gnu General Public License (GPL). The availability of this software package in source form allows tight integration with wireless modems, and other customizations/enhancements.

The existing open-source implementation is available at http://www.mailmeanywhere.org.

See the article *EMSD on Windows CE* [**?**] for more details.

Palm OS Integration Strategy

Palm OS is an effective, general purpose mobile computing platform which, among other applications, comes bundled with a simple e-mail application called Mail. In contrast to the Windows CE Inbox application, however, the standard Palm OS mail application has only a very rudimentary set of features and capabilities. Also in contrast to Windows CE, the Palm OS does not include a set of well defined APIs to facilitate the integration of alternative mail protocols.

To complicate matters further, various third party mail user agents have now become widely available for Palm OS (largely because of the inadequacy of the standard Palm OS mail application), with a variety of different user interfaces. Because of this multiplicity of incompatible Palm OS mail applications, each integration of LEAP with Palm OS must be specific to a particular Palm OS mail user agent.

For these reasons our strategy for integration of the EMSD protocol engine in the Palm OS environment is different from that for Windows CE.

We start with a very basic example implementation of EMSD for Palm OS. This example implementation is specific to the standard mail application, based on file access.

Based on the availability of this package in source form, subject to the Gnu General Public License, a variety of Palm mail user agents can become EMSD-enabled very rapidly. To accomodate this further we have selected certain mail user agents which we consider particularly desirable, and we will participate in the integration of EMSD with these mail user agents.

See the article *LEAP on Palm OS* [**?**] for more details.

17.2 Integration Strategy for Open-Source PDAs

Reference implementations of the LEAP protocols are available as open-source and free software, subject to the Gnu General Public License (GPL).

It is highly desirable for there to exist useful and complete open-source and free software platforms as alternatives to the commercial PDA platforms such as Windows CE and Palm OS.

At the present time two such open-source platforms are of particular importance. These are:

1. Linux-based PDAs
2. eCOS

Our goal in this article is to accommodate the incorporation of the LEAP protocols, in particular EMSD as a Mail Transport Service provider, on both of the above platforms.

17.2.1 LEAP on Linux-Based PDAs

Recently there have been numerous announcements of new Embedded Linux support for PDAs and other handheld personal computing devices.

Such development efforts can benefit from efficient protocols which are specifically oriented towards the needs of miniaturized devices and mobile applications. As we have previously argued in the articles *The WAP Trap* [**?**] and *WAP Scraps* [**?**], the integration of pseudo-open protocols such as WAP with Embedded Linux is the wrong strategy. We are offering LEAP as an alternative [**?**].

Given the rapid emergence of Embedded Linux as a major "third alternative" to Palm OS and Microsoft Windows CE for handheld personal computing devices, we view the integration of LEAP with Embedded Linux as having special importance.

For a frequently updated overview of available and developing Linux-based PDA support, visit *The Linux-PDA and PDA-Linux Quick Reference Guide* at http://www.linuxdevices.com/articles/AT8728350077.html.

Because LEAP implementations on Linux and other Unix platforms are already in place, the incorporation of LEAP into Linux-based PDAs is simply a matter of portation.

The integration of EMSD as a Mail Transport Service provider with open-source Mail User Agents is often quite rapid. For example Pine, and GNUS and VM from Emacs, are major user agents for which EMSD integrations have already been completed.

The work of putting ESRO in kernel space, and providing a socket-based interface to ESRO, is currently in progress.

We consider the full and optimized integration of LEAP with Linux to be a strategic goal. We will continue to improve the quality of existing integrations, and the range of platforms for which LEAP and Linux support is provided.

17.2.2 LEAP on eCos Based Phones and PDAs

eCos is an open-source embedded operating system which is well in tune with the open-source and Linux software development framework. For more information (including source code), see http://sourceware.cygnus.com/ecos

Close integration of LEAP with eCos is highly desirable, because LEAP provides key applications that can greatly increase the utility and value of eCos. Note that on its own, eCos is just an operating system. But in combination with LEAP, eCos can provide Mobile Messaging and other powerful applications.

We intend to integrate LEAP with eCos in the context of a specific embedded hardware platform – specifically, the cell phone hardware platform. A cell phone platform running eCos and LEAP is extremely desirable.

If any such development effort is already underway, we would be pleased to hear about it. We will do all we can to cooperate in the creation of cell-phone-based open-source mobile applications and platforms. Please contact us at the e-mail address shown on the article title page.

17.3 Invitation to Participate

As described in *Operation WhiteBerry*, the incorporation of EMSD in open-source PDA platforms presents enormous benefits.

We invite the developers of open-source PDA platforms to incorporate EMSD into their software. EMSD protocol engines ready for the Linux environment are readily available, and can be easily integrated with your software. Complete implementations of EMSD in open-source form are available at http://www.mailmeanywhere.org.

Chapter 18

Trying out LEAP

All the technologies, products and services required to implement LEAP-based messaging solutions are complete and ready to go. LEAP is not vaporware; it is real and available for implementation *now*.

Anyone who is interested in LEAP is invited to try a hands-on demonstration for themselves. All the components required to allow anyone to try out our products and services are in place. For complete details, see the announcement entitled:

```
          Announcing Free Availability of
             Neda Communication Inc.'s
Enhanced Two-Way Paging (ETWP) Products and Services
            For WindowsCE Hand-Held PCs
             over CDPD and Wireless IP

               Neda-ETWP-wce-gold-1.2

                   June 12, 1998
```

This announcement is available at
http://www.neda.com/products/Neda-ETWP/gold-1.2/wce/announce/announce.html[**?**].

This announcement provides complete information about EVERYTHING that you need to turn your WinCE device into a fully functional Mobile Messaging device. Specifically, it describes:

1. Buying a CDPD modem and getting a CDPD account from your cellular service provider.
2. Configuring your WinCE device to work with your CDPD modem.
3. Getting an Enhanced Two-Way Paging (ETWP) account from Neda.
4. Loading Neda's Enhanced Two-Way Paging product into your WinCE device.
5. Sending and receiving messages.

Chapter 19

WhiteBerry and Bluetooth

19.1 Introduction

In this article we will describe how the EMSD (Efficient Mail Submission & Delivery)[**?**] protocol can be used to provide powerful new messaging capabilities. The capabilities we describe are based entirely on existing technologies, and on existing open and free protocols. We will describe two major new applications:

- Use of EMSD to provide true Mobile Messaging capability to the user who habitually carries a cell phone and a PDA device, without requiring additional hardware components. In particular, the model we describe does not require the customary add-on wireless modem for the PDA, so that the cost and inconvenience of this component can be eliminated. We refer to this model as the **WhiteBerry/{Bluetooth}** messaging solution.

- Use of EMSD to provide a general **Mail Notification** functionality, which can provide the user with immediate notification of the arrival of certain types of messages in the user's back-end Message Center.

This article is a follow-on to a previous Manifesto article entitled *Use of EMSD for Mail Notification* [**?**]. In that article we describe the Mail Notification model in technical terms; in this article we describe its implementation in a specific messaging context.

The models we describe in this paper are part of a more general industry movement which we call **Operation WhiteBerry** [**?**]. As described in Section 20.4, the Operation WhiteBerry movement is supported by a comprehensive set of development tools and resources. In particular, free open-source software implementations of the EMSD protocol are immediately available for a wide variety of end-user devices and Message Center platforms. All this software is readily available at the **MailMeAnywhere** open-source software distribution center at `http://www.MailMeAnywhere.org`.

The models we describe are thus not merely theoretical, but can be implemented immediately and without difficulty using existing software components.

19.1.1 Bluetooth vs. {Bluetooth}

In Section 19.3 we describe how true Mobile Messaging can be implemented based on a cell phone and a PDA. A key component of this model consists of a technology for short range, low power wireless communications between the cell phone and the PDA. At present there are two candidate technologies which address this requirement: 802.11, and Bluetooth.

The 802.11 [?] standard for wireless LANs (WLANs) is an IEEE Media Access Control (MAC) specification which was completed in 1997. The latest step in the evolution of the specification has been the ratification of the 802.11b or High Rate, providing data rates of up to 11 Mbps which can reach distances of hundreds of feet. Products based on 802.11 are now available in a variety of forms, and are experiencing widespread usage. Interoperability amongst the 802.11 products is common place and certification processes such as Wi-Fi™(wireless fidelity) which assurance of interoperability are in place. This widespread availability and usage has brought significant economies of scale benefits at both component and device level, greatly reducing the cost of 802.11 solutions. 802.11 does not specifically address the low power end of short range wireless communications, e.g. for communications with small battery operated devices. However it is to be expected that low power flavors of 802.11 will be developed in the future.

Bluetooth [?] is generally intended as replacement for wires and links between devices like laptops, cell phones, headsets and handheld computers. Bluetooth was intended to be a low-cost, low-power wireless technology for short-range radio communication. It has been under development for more than three years. Bluetooth provides fast and reliable digital transmission of both voice and data. Unlike infrared technology, Bluetooth is not limited to line-of-sight communication. Some Bluetooth products are available, but the price remains high. For more information visit the Bluetooth website at `http://www.bluetooth.com/`.

There are many in the wireless industry who argue that Bluetooth is no longer a viable option for short range wireless communications, since 802.11 is already in widespread usage and has lower cost; see for example the article *Bye, Bye Bluetooth* [?]. This may very well be the case for wireless communications in general, but it is not yet clear what will be the winning technology in the extremely low power short range environment. Bluetooth may yet play a role in addressing the requirements of extremely low power devices such as battery-operated cell phones and miniaturized PDAs. Furthermore, it is reasonable to expect pieces of Bluetooth technology (e.g. physical layer, usage profiles) to be merged in with 802.11 solutions.

However, this debate, and the eventual technological outcome, are irrelevant to our current purposes. What is clear is that there is a need for short range, low power wireless connectivity, and that *some* technology will eventually occupy this space, whether it be Bluetooth, 802.11 or something else altogether.

The model we describe in Section 19.3 requires cell phone/PDA connectivity, but it is independent of the particular technology used to accomplish this. We refer to Bluetooth in the title of this article for the sake of catchy wordplay; however the model we present is in no way locked in to Bluetooth.

Since repeated use of the wording "Bluetooth, or similar technology for short range wireless communications" will quickly become wearisome to the reader, we will use the notation {Bluetooth} throughout this article as shorthand for this. Specifically, we define {Bluetooth} to mean:

> *Bluetooth, or a Bluetooth-like technology that can provide short range, low power, inexpensive wireless communications between a cell phone and PDA.*

On those occasions when we really do mean Bluetooth specifically, we will say Bluetooth.

19.1.2 Industry Characteristics and Trends

There are various industry characteristics and trends that act to enable or facilitate the models we describe in this paper. The following characteristics or trends are of particular relevance:

- **{Bluetooth} support increasing on cell phones**. Support for Bluetooth itself is becoming increasingly common on cell phones, for example to enable the use of cordless hands-free headsets and other close-proximity components. We expect that this trend will continue, and that support for Bluetooth or similar technology will eventually become the norm for cell phones.
- **{Bluetooth} support increasing on PDAs**. Bluetooth hardware is also becoming increasingly common on PDAs, for example to enable cordless synchronization with a desktop PC. We expect this trend to continue, and

that Bluetooth or similar technology will eventually become the norm for PDAs.

- **IMAP (Internet Message Access Protocol) support increasing on PDAs**. IMAP Client software is becoming increasingly widespread on PDAs, for example to support e-mail synchronization of the PDA with a desktop mailbox application. We expect that IMAP support will eventually become the norm for PDAs.

- **EMSD (Efficient Mail Submission & Delivery) support to increase in Message Centers**. As the open WhiteBerry solution becomes widespread throughout the Mobile Messaging industry, we expect that EMSD support will increasingly be provided in Message Centers.

- **Essential Mobile Messaging requirements: Unconscious carry and push-mode delivery**. Market experience has now demonstrated beyond any doubt that the correct Mobile Messaging model includes two essential characteristics: unconscious carry, meaning that messaging is provided on a device that can be carried without moment-to-moment awareness of its presence; and push mode delivery, meaning that messages are delivered directly to the recipient without requiring an explicit poll or retrieval action. Any viable messaging solution must have these two characteristics.

- **Cell phone and PDA a viable device configuration**. The mobile professional of today can choose to carry any of a wide variety of communications and productivity devices. A common choice among users is to carry separate cell phone and PDA devices. This user may also equip himself with Mobile Messaging capability, implemented either via the cell phone, via the PDA, or by means of a specialized messaging device.

  Though other options are commonly exercised, we will assume throughout this article that the separate cell phone and PDA option is and will remain a viable and important option in the mobile marketplace.

  There already exists a large body of speculative literature describing the pros and cons of various device types and configurations, and predicting their future success or failure. The purpose of this article is not to add to this. For those who are interested, however, we provide a justification for this assumption in the appendix to this article.

19.2 What is WhiteBerry?

WhiteBerry is an open Mobile Messaging solution, based on a set of open, patent-free protocols called the **Lightweight & Efficient Application Protocols**, or **LEAP**. LEAP is a family of high-performance, efficient protocols that are ideal for mobile and wireless applications. For more information about the LEAP protocols see the LEAP Forum website at `http://www.LeapForum.org`.

The Mobile Messaging landscape of today consists of a number of closed, competing solutions, such as the popular BlackBerry system from Research in Motion (RIM). These existing solutions are based on closed and/or proprietary protocols, and are limited to a specific set of components which do not interoperate with those of any other solution.

The WhiteBerry solution, by contrast, is an open industry model, based on generic industry-wide component interoperability. Under the WhiteBerry scenario, Mobile Messaging functionality is provided by a multi-vendor set of products and services, and the necessary cross-vendor interoperability is guaranteed by the openness and integrity of the underlying LEAP protocols. WhiteBerry thus provides an open alternative to existing successful solutions such as BlackBerry.

Since the underlying protocols are completely open, WhiteBerry allows market entry, cooperation and competion at every point within the Mobile Messaging technology chain. The result is greatly increased business opportunities throughout the Mobile Messaging industry, increased competition, and better solutions for the end user. In particular, the two EMSD applications we describe in this paper represent major business opportunities.

A detailed description of the WhiteBerry solution is provided in the Manifesto article *Operation WhiteBerry* [**?**]. This article is available on the LEAP Forum website at http://www.LeapForum.org/operationWhiteberry/index.html, and also on the Free Protocols Website at http://www.freeprotocols.org/operationWhiteberry/index.html.

In the context of the present article, the key characteristic of the WhiteBerry solution is that it allows independent selection of each component of the Mobile Messaging solution. In particular, the end user or systems integrator may specify any appropriate component to provide airlink functionality for the mobile device.

19.3 The WhiteBerry/{Bluetooth} Messaging Solution

As we discuss at length in Appendix A.1, the modern professional can equip himself with a variety of mobile communications and productivity devices. But a common choice is for the user to carry two specialized devices: a cell phone for voice communications, and a wireless-enabled PDA for Mobile Messaging.

A major disadvantage of this configuration is that the user must carry and maintain battery charge for three distinct components: cell phone, PDA and add-on modem. But note that this user is in fact carrying around two airlink components: one in the cell phone, and one in the form of the modem. Both of these components perform essentially the same function. The user, or the systems integrator, may well wonder if this component duplication can be avoided.

The basic premise of the WhiteBerry/{Bluetooth} solution, simply stated, is to use the cell phone as the airlink component for the PDA, and use Bluetooth or similar wireless technology for local communication between these two devices. Under the WhiteBerry/{Bluetooth} scenario the add-on modem for the PDA is unnecessary, and can be dispensed with altogether.

Since the WhiteBerry solution provides generic component interoperability, it allows any appropriate component to provide airlink functionality for the Mobile Messaging device. In particular, this functionality need not be provided by the customary wireless modem; it can just as well be provided by a cell phone. All that is required is for the cell phone to be able to communicate with the PDA – and this is something that can very conveniently be accomplished by means of Bluetooth wireless technology, or its equivalent.

The general operation of this model is shown in Figure 19.1. Under the WhiteBerry/{Bluetooth} scenario the user maintains his cell phone and PDA as two physically distinct devices, but the cell phone provides airlink functionality for both voice and data messaging. The user uses the cell phone for voice communications as usual; while Mobile Messaging capability is provided by means of integration of the cell phone with the PDA via {Bluetooth}.

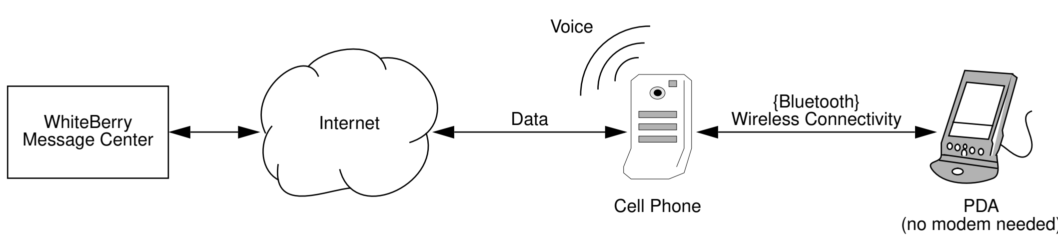


Figure 19.1: The WhiteBerry/{Bluetooth} Solution

An immediate advantage of the WhiteBerry/{Bluetooth} solution is that it eliminates the cost, size and inconvenience of the wireless modem for the PDA, replacing this with inexpensive miniaturized Bluetooth or equivalent hardware. The only disadvantage of this is that the Mobile Messaging functionality of the PDA requires physical proximity of the cell phone. If the two devices are separated too far, or if the cell phone battery runs out, the Mobile Messaging functionality of the PDA is lost. (But not, it is important to note, its PDA functionality.)

19.3.1 Basic WhiteBerry/{Bluetooth} Implementation Architecture

Figure 19.2 shows a basic implementation architecture for the WhiteBerry/{Bluetooth} model. This implementation is basic in the sense that it requires the least amount of device integration. In particular, the basic implementation requires only minimal cell phone integration, and no PDA integration at all.

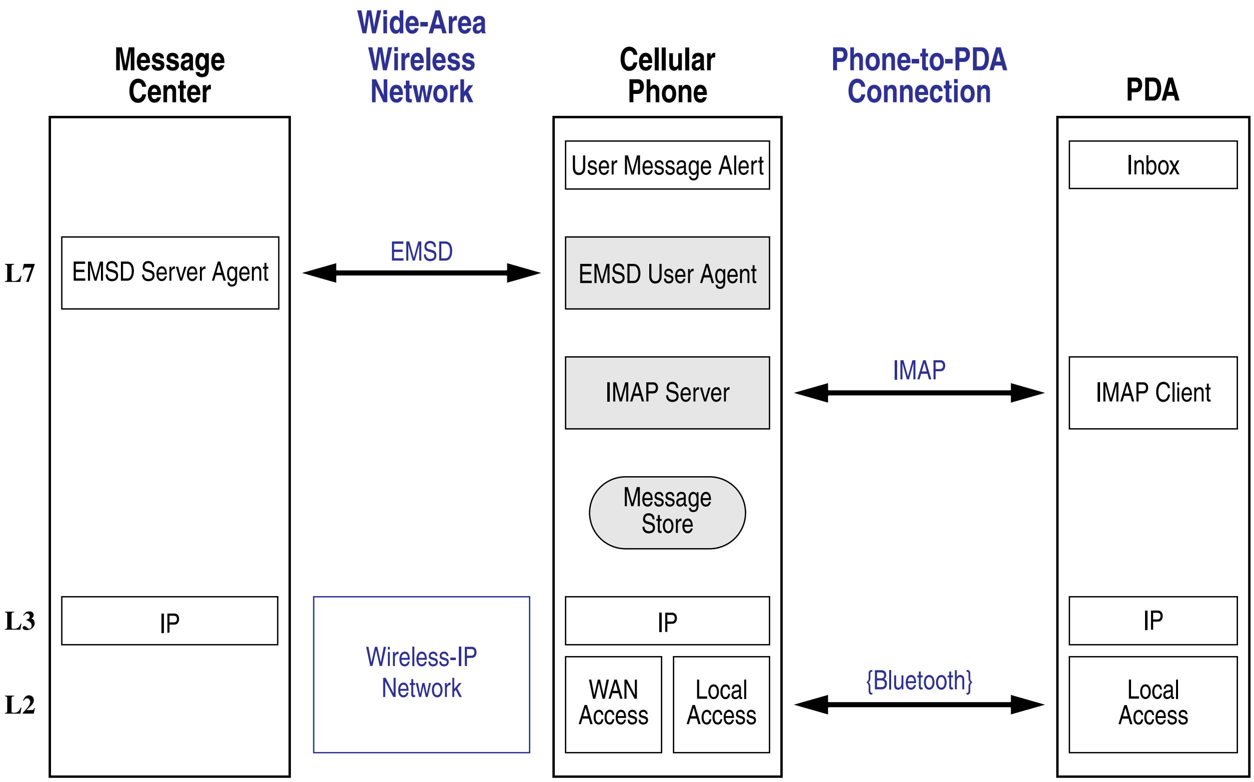


Figure 19.2: Basic Implementation Architecture.
Device components which are not commonly bundled are shown shaded.

On the Wide-Area Network (WAN) side of the cell phone, message transfer between the message center and the cell phone takes place via **EMSD** [?], [?]. EMSD, or the **Efficient Mail Submission & Delivery** protocol, is the e-mail component of the LEAP family of protocols. EMSD is highly optimized for efficient delivery of short e-mail messages. Given the bandwidth constraints of wide-area wireless networks, EMSD is therefore ideal for this application. For complete details on EMSD see the *LEAP Manifesto* article *EMSD: The LEAP E-Mail Component* [?], or visit the EMSD website at `http://www.EMSD.org`.

Since EMSD supports the push mode of message delivery, the cell phone can accept incoming messages at any time, and at any point within the network coverage area. Communications at Layers 2 and 3 takes place via IP over any appropriate Wireless-IP network.

On the local side of the cell phone, message transfer between the phone and the PDA is handled by IMAP [?] or other appropriate protocol. Layer 2 communications is implemented by means of Bluetooth, or any alternative technology appropriate to this.

Under the WhiteBerry/{Bluetooth} scenario the cell phone normally remains on at all times, so that it is always ready to receive messages. The PDA may be turned on or off as desired by the user.

When a message arrives at the message center it is delivered to the cell phone via the EMSD protocol. The cell phone stores the incoming message, then alerts the user that a new message has arrived. The cell phone may alert the user in any of the usual ways, i.e. by means of an audible (beep, buzz, ring) visual (flash) or tactile (vibration) indication. The user may then turn on the PDA and retrieve the message from the cell phone via the IMAP protocol.

The basic implementation requires that the cell phone and PDA include all the components shown in Figure 19.2. Note that the WhiteBerry/{Bluetooth} solution is based entirely on readily available, open and free protocols, that conform fully to the Internet end-to-end principle. Specifically, it is based on TCP/IP, IMAP and EMSD.

All those components which are not commonly bundled with the cell phone and PDA are shown shaded in this figure; all unshaded components we consider to be standard or near-standard issue for most modern devices. In particular, we consider {Bluetooth} support to be the norm for both the cell phone and the PDA, and IMAP Client software to be the norm for the PDA. These components are increasingly being provided in cell phone and PDAs; a trend we expect to continue.

In terms of device integration, therefore, the basic implementation requires only that the cell phone be equipped with the EMSD User Agent protocol engine, a minimal IMAP Server protocol engine, and a Message Store. The EMSD and IMAP protocol engines must also be properly integrated with the cell phone User Message Alert and {Bluetooth} capabilities. As described below, open-source and free software for both the EMSD and IMAP protocol engines is readily available.

Note that the Message Store requirements of the cell phone are quite limited, since it need only provide short-term storage of short e-mail messages for a single user. The memory requirements for this are well within the design parameters for most modern cell phones.

Note also that the basic implementation *requires no PDA integration whatsoever* – the PDA can be used entirely as is, without requiring any hardware or software modifications. This means that the basic implementation can be carried out independently by any cell phone manufacturer; no joint integration with a PDA manufacturer is required. The designed-in functionality of the PDA is intended to allow general e-mail communication with a desktop system or Message Center. But the cell phone manufacturer can use this same functionality to replace the intended desktop system or Message Center with its own cell phone – and the PDA is none the wiser.

Finally, it is worth noting that the WhiteBerry/{Bluetooth} model allows the use of *any* appropriate device as the airlink component – not just a cell phone. We have framed the discussion in terms of a cell phone as the airlink device since this is a particularly convenient and logical model. However, the role of the cell phone in Figure 19.2 could be played by any appropriate airlink device.

19.3.2 A Word About SMS

The cell phone manufacturer may note that SMS could also be used to provide push-mode message delivery to the phone. While it is true that SMS can provide the same functionality as EMSD, SMS is not an IP-based Internet protocol, and therefore has no long-term future in the Internet domain. In particular SMS, unlike EMSD, integrates extremely poorly with existing Internet e-mail protocols.

19.4 Mail Notification

A basic shortcoming of conventional e-mail protocols such as IMAP is that they do not support the **push** mode of message delivery, in which the message is delivered directly to the end-user device. Instead, the user must initiate an explicit **pull** action to retrieve new messages from the Message Center.

This is one of the shortcomings of existing protocols that EMSD addresses: EMSD fully supports the push mode of delivery. In addition, EMSD has been designed with a major emphasis on efficiency; it is highly optimized for the efficient delivery of "short" Internet e-mail messages – meaning text messages up to approximately 4 kbytes in length.

Although the great majority of daily e-mail messages fall within this category, there is no inherent upper limit to the size of e-mail messages. Any given e-mail can be extremely large, particulary if large attachment files are included with the message. However, the efficiency characteristics of EMSD do not extend to long messages; and in terms of efficiency, IMAP (because of TCP, sliding windows etc.) is more suitable for long messages than EMSD.

Thus we see that in the case of long messages, there is a functional lacuna in the existing e-mail protocol framework. EMSD can push the message to the recipient but cannot deliver it efficiently; while IMAP can deliver the message efficiently, but cannot push it to the recipient.

The ideal solution to this would be a well-integrated family of protocols, designed *ab initio* to provide full support for these, and other, messaging requirements. However, no such integrated protocol family exists at present. Nevertheless, the delivery of long messages can be properly managed by the integration of existing, though independently designed, protocols such as EMSD and IMAP.

In particular, long message delivery can be greatly enhanced by by means of a **Mail Notification** service, which would alert the user to the arrival of the message at the Message Center. Such a service can readily be provided by EMSD, and used in conjunction with a suitable message delivery protocol such as IMAP, this can greatly improve the timeliness of delivery of long messages.

The general operation of the Mail Notification service is shown in Figure 20.1. If a short message arrives at the Message Center, it is delivered directly to the mobile device via the EMSD protocol, and the device then alerts the user to its arrival. In this case no explicit notification is required, since notification is implicit in the arrival of the message itself.

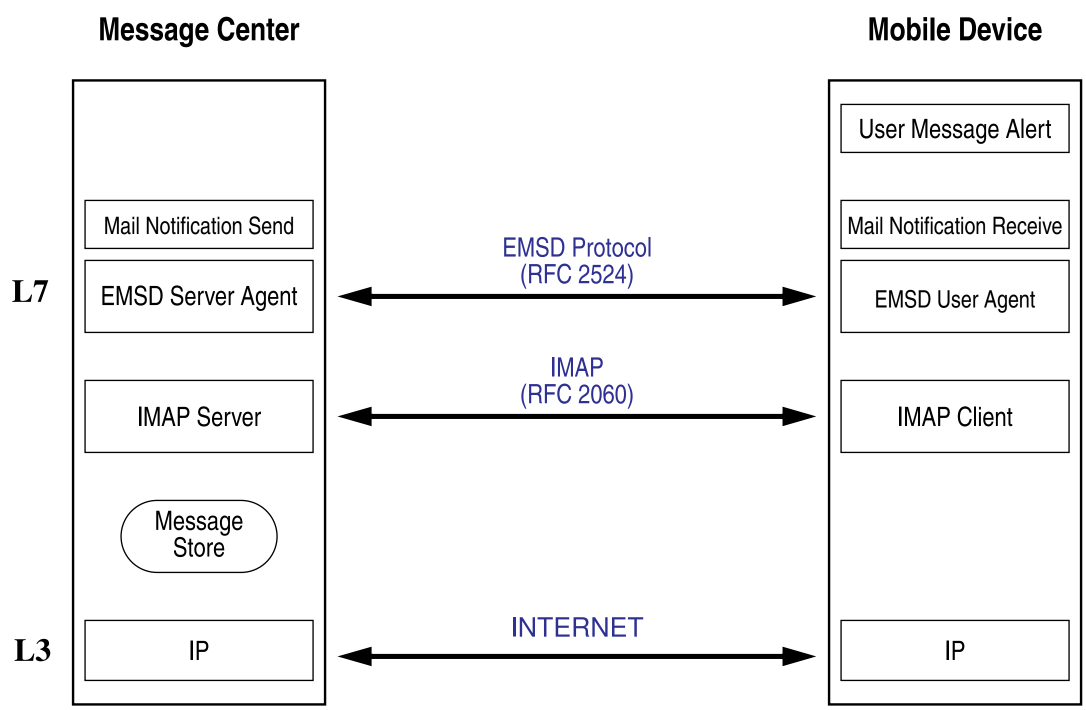


Figure 19.3: General Mail Notification Model

If a long message arrives at the Message Center, then the EMSD protocol is used to send a **Mail Notification** to the device. This could take the form of a generic notification, simply flagging the arrival of a long message at the Message Center; alternatively the notification could take the form of a message digest for perusal by the user. In the former case the Message Center would require an appropriate Mail Notification Send function, and the mobile device would require a complementary Mail Notification Receive function.

Regardless of the nature of the Mail Notification, the mobile device would then alert the user to the availability of the long message at the Message Center. The user can then retrieve the message at his discretion, either sight unseen (in the case of a generic notification), or on the basis of the message digest. Retrieval of the long message, of course, takes place via the IMAP protocol.

The above operational model could be described as a "push-assisted" mode of operation – i.e. the notification is pushed to the user, but the user must then pull the message itself. Alternatively, receipt of the notification could trigger automatic retrieval of the message by the device. In this case, the combination of Mail Notification via EMSD and message retrieval via IMAP is functionally equivalent to true push mode delivery of long messages.

19.5 Mail Notification in the WhiteBerry/{Bluetooth} Model

Figure 19.4 shows how the above Mail Notification model can be implemented in the context of the WhiteBerry/{Bluetooth} solution, thereby bringing enhanced Mobile Messaging functionality to this solution.

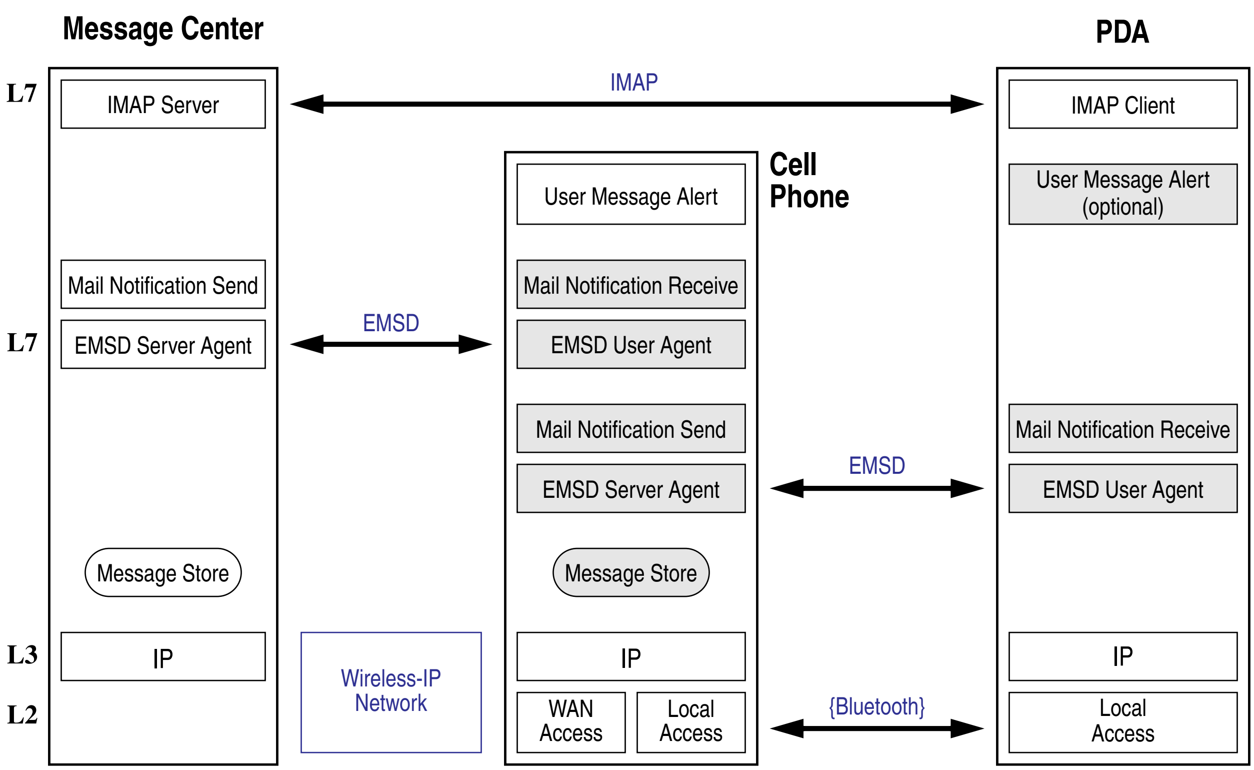


Figure 19.4: Mail Notification in the WhiteBerry/{Bluetooth} Model.
Device components which are not commonly bundled are shown shaded.

As in the case of the basic WhiteBerry/{Bluetooth} implementation, the cell phone normally remains on at all times, while the PDA may be turned on or off as desired by the user.

Whenever a short message arrives at the Message Center it is delivered to the cell phone via EMSD, then immediately transferred from the cell phone to the PDA, again via EMSD. If the PDA is off, the push mode delivery by the cell phone causes it to wake up automatically, so that the message is now immediately available to the user. No pull action is required by the user or the PDA.

The user can be alerted to the arrival of the message either by the cell phone or by the PDA, provided the PDA has suitable user alert functionality.

Whenever a long message arrives at the Message Center a notification of this is sent to the cell phone via EMSD, then immediately relayed to the PDA, also via EMSD. The notification could take the form either of a generic message arrival flag, or a message digest for review by the user. In the former case the Message Center, cell phone and PDA all require the appropriate Mail Notification Send and Mail Notification Receive functions.

If the PDA is off, the push mode delivery of the notification causes it to wake up, and the PDA then retrieves the long message from the Message Center via the IMAP protocol. Note that retrieval of the message takes place by direct IMAP communications between the PDA and the Message Center, as indicated in Figure 19.4. The cell phone is bypassed entirely at Layer 7, and provides only Layer 3 connectivity via IP. In other words, IP packets are routed through the cell phone transparently to Layer 7.

Retrieval of the message by the PDA may be initiated in any of the ways previously discussed: either by the user at his discretion sight unseen (in the case of a generic notification); or by the user at his discretion on the basis of the message digest; or receipt of the notification could trigger automatic retrieval of the message by the PDA.

As in the case of short message delivery, the user can be alerted to the availability/arrival of the long message either by the cell phone or by the PDA.

This implementation requires that the cell phone and PDA include all the components shown in Figure 19.4. Note that the implementation is based on the same readily available, open and free protocols as the basic WhiteBerry/{Bluetooth} implemenation; namely, TCP/IP, IMAP and EMSD.

As before, all those components which are not commonly pre-bundled are shown shaded in Figure 19.4. In addition to the components required for the basic WhiteBerry/{Bluetooth} implementation, the cell phone must now also be equipped with an EMSD Server Agent to deliver notifications and messages to the PDA. If the notifications are to take the form of a generic message arrival flag, then the cell phone also requires both a Mail Notification Receive function and a Mail Notification Send function.

In its turn the PDA requires an EMSD User Agent, and (possibly) a Mail Notification Receive function. If the PDA is to be used as the user alert device then it will also require an appropriate User Message Alert functionality; otherwise the user alert can be issued by the cell phone.

19.6 Development Framework and Resources

Those who find the WhiteBerry/{Bluetooth} scenario interesting or compelling are invited to participate in its implementation and deployment.

The WhiteBerry/{Bluetooth} solution is part of a more general industry effort which we call **Operation WhiteBerry**. A comprehensive development framework exists to assist organizations and individuals who wish to participate in Operation WhiteBerry in general, or the WhiteBerry/{Bluetooth} solution in particular. This framework provides a complete set of development resources, including:

- The MailMeAnywhere open-source software distribution center and development forum at `http://www.MailMeAnywhere.org`.
- The LEAP Forum at `http://www.LeapForum.org`
- The EMSD Organization at `http://www.EMSD.org`
- The IMAP Connection at `http://www.imap.org/`

Each of these forums hosts several mailing lists which facilitate various forms of participation. `MailMeAnywhere` hosts a public forum for the collective development and enhancement of LEAP protocol engines and integration tools, while `EMSD.org` is a development forum for the EMSD protocol.

`LeapForum.org` is oriented towards coordinating usage of the LEAP protocols from an industry and business development perspective; it provides a means for organizations and individuals to announce their participation in Operation WhiteBerry, seek out partners, and coordinate cooperative effort.

19.6.1 Development Support from Neda Communications, Inc.

In addition, development support is immediately available from Neda Communications, Inc. As the primary developer of the LEAP protocols, Neda is intimately familiar with all aspects of the WhiteBerry model, including the EMSD applications described in this paper.

Neda can provide assistance to manufacturers and systems designers at every stage of integration and implementation of WhiteBerry/EMSD-based solutions. Neda provides a comprehensive set of support services, including sales of software licenses, assistance with portation, and consulting services. For complete details, visit the Neda website at `http://www.neda.com/`.

19.6.2 Invitation to Cell Phone Manufacturers

We particularly encourage cell phone manufacturers such as Motorola, Nokia, Ericsson and others to do the simple engineering concept validation work required to prove the viability and ease of implementation of the basic WhiteBerry/{Bluetooth} model. This requires nothing more than:

- Acquiring the EMSD User Agent software in source form
- Acquiring a basic IMAP Server Agent implementation in source form
- Integrating these two protocol engines with the cell phone software

The EMSD User Agent source code is readily available at `http://www.MailMeAnywhere.org`. Alternative versions of the IMAP Server Agent software are available in source form from both the University of Washington and Carnegie-Mellon University; both versions are also readily available at `http://www.MailMeAnywhere.org`.

A complete package of resources required to emulate the basic implementation using a desktop PC in place of the cell phone is also available at `http://www.MailMeAnywhere.org`. This can be used for demonstration purposes, and as an example or starting point for software development purposes.

19.6.3 Open-Source Software Licensing

Both the EMSD and IMAP software is freely available in source code form. In particular, the EMSD software is available under the terms of the GNU General Public License (GPL). The GPL places no restrictions on use of the software for the development and proof-of-concept purposes described above. The GPL model is also very well-suited to allowing and encouraging widespread implementation of EMSD and IMAP on open PDA platforms. Specifically, the GPL is oriented towards environments which permit the integration of open-source software components.

However, the cell phone is not such an environment. On the contrary, cell phones have traditionally been regarded as closed consumer devices, for which the source code is almost always a proprietary trade secret.

The GPL places certain restrictions on commercial usage of the software, including its usage in closed devices, and such usage may be in violation of the GPL. Phone manufacturers are advised that they may need to acquire professional license agreements in order to make use of the EMSD and IMAP protocol engines in commercial contexts. Such commercial (non-GPL) licenses are readily available for use in cell phones and other closed environments.

Chapter 20

Use of EMSD for Mail Notification

20.1 Introduction

In this article we will make the case that there is a general need within the Mobile Messaging industry for a **Mail Notification** functionality, which would provide the user with immediate notification of the arrival of certain types of messages in the user's back-end Message Center. We will then describe how the required Mail Notification functionality can be implemented, based entirely on existing open protocols and technologies.

As we describe in Section 20.4, the proposed Mail Notification model is supported by a comprehensive set of development tools and resources. In particular, free open-source software implementations of the underlying LEAP protocols are immediately available for a wide variety of Message Center and Mail User Agent platforms. All this software is readily available at the **MailMeAnywhere** open-source software distribution center at `http://www.MailMeAnywhere.org`.

The model we describe in this article is thus not merely theoretical, but can be implemented immediately and without difficulty using existing software components.

20.1.1 Intended Audience

This article is directed towards Internet protocol engineers – it describes the Mail Notification model in technical engineering terms.

For a description of an implementation of this model in a specific context, see the Manifesto article *WhiteBerry and Bluetooth* [**?**]. That article may be considered a sister article to this one – while this article is directed towards an engineering audience, *WhiteBerry and Bluetooth* is directed towards a business development audience.

20.2 The EMSD Protocol

The Mail Notification model is based on the the **Efficient Mail Submission & Delivery** protocol, or **EMSD** [**?**], [**?**]. EMSD is a member of a broader family of high-performance, efficient protocols called the **Lightweight & Efficient Application Protocols**, or **LEAP**. For more information about the LEAP protocols in general see the LEAP Forum website at `http://www.LeapForum.org`.

EMSD is a messaging protocol that has been designed with a major emphasis on efficiency. In addition, EMSD fully supports the **push** mode of delivery, in which messages are delivered directly to the Mail User Agent, without requiring an explicit retrieval action by the user.

In common with other LEAP protocols, EMSD is truly open, RFC published, and patent-free; and declarations regarding the patent-free nature of EMSD have been made to the Free Protocols Foundation. For complete details see the *LEAP Manifesto* article *EMSD: The LEAP E-Mail Component* [**?**], or visit the EMSD website at `http://www.EMSD.org`.

EMSD has been designed to complement the existing Internet protocol framework in a variety of ways. In addition to the Mail Notification model described in this paper, EMSD is also the basis of an open Mobile Messaging model. This model is described in detail in another Manifesto article entitled *Operation WhiteBerry* [**?**].

20.3 Mail Notification

A basic shortcoming of conventional e-mail protocols such as IMAP (Internet Message Access Protocol) [**?**] is that they do not support the **push** mode of message delivery. Instead, the user must initiate an explicit **pull** action to retrieve new messages from the Message Center.

As noted above, EMSD fully supports the push mode of delivery, and therefore properly addresses this shortcoming. Also as noted above, EMSD is a highly efficient messaging protocol. Specifically, it is optimized for the efficient delivery of "short" Internet e-mail messages – meaning text messages up to approximately 4 kbytes in length.

Though the great majority of daily e-mail messages fall within this category, there is no inherent upper limit to the size of e-mail messages. Any given e-mail can be extremely large, particulary if large attachment files are included with the message. The efficiency characteristics of EMSD do not extend to long messages; and in terms of efficiency, IMAP (because of TCP[**?**], sliding windows etc.) is more suitable for long messages than EMSD.

Thus we see that in the case of long messages, there is a functional lacuna in the existing e-mail protocol framework. EMSD can push the message to the recipient but cannot deliver it efficiently; while IMAP can deliver the message efficiently, but cannot push it to the recipient.

This deficiency is particularly problematic in the Mobile Messaging arena, which has inherent requirements for both urgent push-mode **and** efficient delivery. Immediate delivery of time-critical messages is an essential requirement of the mobile user; while the constraints of wireless networks and miniaturized mobile devices demand highly efficient transfer of data.

The ideal solution to this would be a protocol which is suitable for long messages and which also supports the push mode of delivery, but no such protocol exists at present. In its absence, the delivery of long messages can be greatly enhanced by means of a **Mail Notification** service, which would alert the user to the arrival of the message at the Message Center.

As we will see below, such a service can readily be implemented based on EMSD. Used in conjunction with a suitable message delivery protocol such as IMAP, this can greatly improve the timeliness of delivery of long messages.

20.3.1 Mail Notification Implementation Model

The Mail Notification implementation model is shown in Figure 20.1.

In the case of short messages, EMSD can do it all – it can push the message directly to the recipient, and can do so efficiently. Therefore whenever a short message arrives at the Message Center, it is simply delivered directly to the Mail User Agent via the EMSD protocol. The message is now immediately available to the user, and the Mail User Agent can alert the user to this in any of the usual ways – e.g. by means of a window pop-up, audible alert, or flashing icon.

It is in the case of long messages that a notification mechanism is required. If a long message arrives at the Message Center, then the EMSD protocol is used to send a **Mail Notification** to the Mail User Agent. This could take the form of a generic notification flag, simply reporting the arrival of a long message at the Message Center; alternatively the

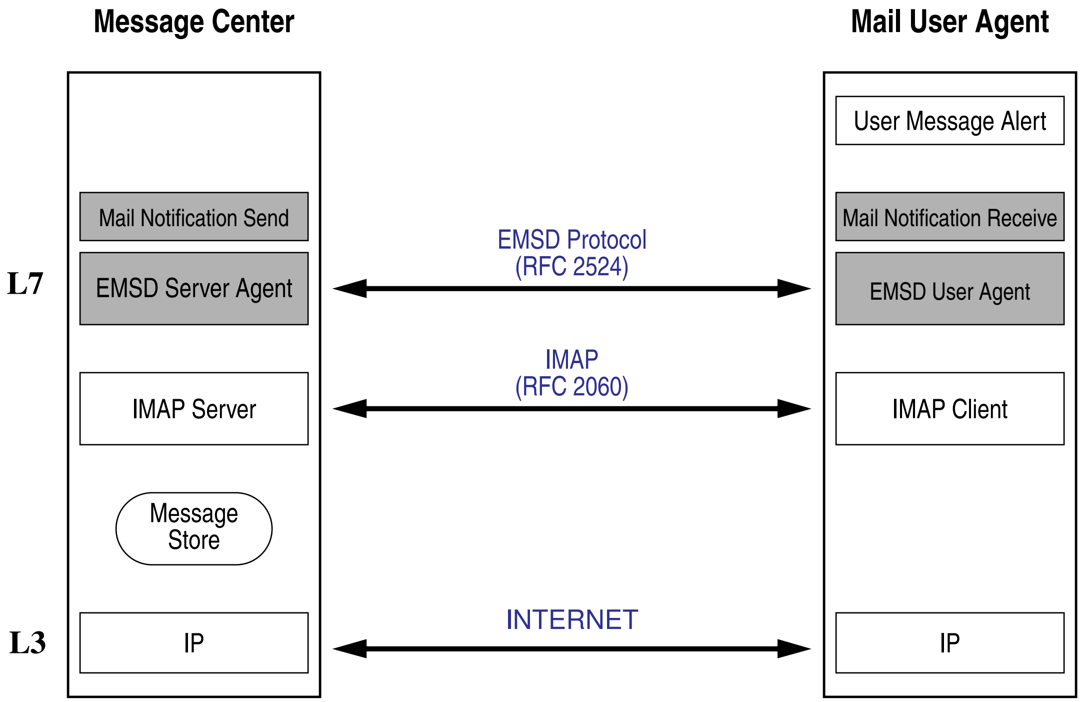


Figure 20.1: Mail Notification Model

notification could take the form of a message digest for perusal by the user. In the former case the Message Center requires an appropriate Mail Notification Send function, and the Mail User Agent requires a complementary Mail Notification Receive function, as shown in Figure 20.1.

The Mail User Agent can respond to the notification in either of two basic ways:

1. Receipt of the notification can trigger automatic retrieval of the message by the Mail User Agent.

2. The Mail User Agent can merely alert the user to the availability of the long message at the Message Center; it is then the responsibility of the user to retrieve the message at his discretion, either sight unseen (in the case of a generic notification), or on the basis of the message digest.

In either case, retrieval of the long message takes place via IMAP, or other appropriate long-message protocol.

The first mode of operation is most appropriate for a desktop messaging environment, which can readily accept delivery of messages of arbitrary size, so that the user need not be involved in the delivery process. Note that in this mode of operation, the combination of Mail Notification via EMSD and message retrieval via IMAP is functionally equivalent to true push mode delivery for long messages.

In the case of a mobile messaging environment, automatic delivery of a long message to a limited-capability mobile device may not be appropriate; and in this case the second option may be more appropriate. This operational model could be described as a "push-assisted" mode of operation – i.e. the notification is pushed to the user, but the user must then pull the message itself.

Note that the implementation of Figure 20.1 is based entirely on readily available, open and free protocols; specifically EMSD, IMAP and TCP/IP. Layer 7 communications takes place via EMSD and IMAP; while communications at Layer 3 takes place via IP.

The integration requirements for the implementation of Figure 20.1 are particularly simple. Those components which require integration are shown shaded in the figure; all unshaded components are already commonly integrated in modern Message Centers and Mail User Agents.

This implementation thus requires only that the Message Center be equipped with the EMSD Server Agent protocol engine and a suitable Mail Notification Send function; and that the Mail User Agent be equipped with the EMSD User Agent protocol engine and a Mail Notification Receive function. The EMSD and IMAP protocol engines must also be properly integrated with the User Message Alert function of the Mail User Agent.

As described below, open-source and free software for both the EMSD and IMAP protocol engines is readily available.

20.3.2 Mail Notification in Mobile Environments

The Mail Notification model is particularly advantageous in the Mobile Messaging environment, where long messages cannot be handled in a wholly satisfactory way be either EMSD or IMAP.

The general model of Figure 20.1 translates fully into the mobile domain, in which the Mail User Agent takes the form of a miniature handheld device such as a cell phone or PDA. There are however, some additional considerations in the case of mobile devices.

The mobile device may be turned on or off as desired by the user. However the wireless modem component of the device will normally remain on at all times, so that the device can accept incoming messages at any time, and at any point within the network coverage area.

Whenever a short message arrives at the Message Center, it is delivered directly to the mobile device via the EMSD protocol. If the device is off, the push mode delivery causes it to wake up automatically, so that the message is now immediately available to the user. The device can then alert the user to this in any of the usual ways for mobile devices, i.e. by means of an audible (beep, buzz, ring) visual (flash) or tactile (vibration) indication.

Whenever a long message arrives at the Message Center, a notification of this is sent to the mobile device via the EMSD protocol. Again, if the device is off, the push mode delivery of the notification causes it to wake up and alert the user to the availability of the long message at the Message Center.

The user can then retrieve the message at his discretion, either sight unseen (in the case of a generic notification), or on the basis of the message digest. Retrieval of the long message takes place via IMAP or other appropriate protocol. Automatic retrieval of the long message will not normally be an appropriate mode of operation for miniaturized mobile devices.

The integration requirements for mobile devices are the similar to those of generic Mail User Agents, except that the Mail Notification Receive function may be considered to be optional, since this is not required if the notification takes the form of a message digest.

As in the case of generic Mail User Agents, IMAP is also becoming the norm on PDAs and other mobile devices. IMAP Client software is becoming increasingly widespread on devices, for example to support e-mail synchronization of the device with a desktop mailbox application.

In terms of integration, therefore, the mobile device requires only an EMSD User Agent protocol engine and (optionally) a Mail Notification Receive function. The EMSD and IMAP protocol engines must also be properly integrated with the device User Message Alert function.

20.4 Development Framework and Resources

Those who find the Mail Notification model interesting or compelling are invited to participate in its implementation and deployment.

The Mail Notification model is part of a more general industry effort which we call **Operation WhiteBerry**. A detailed description of Operation WhiteBerry is provided in the Manifesto article *Operation WhiteBerry* [**?**]. This article is available on the LEAP Forum website at
`http://www.LeapForum.org/operationWhiteberry/index.html`, and also on the Free Protocols Website at `http://www.freeprotocols.org/operationWhiteberry/index.html`.

A comprehensive development framework exists to assist organizations and individuals who wish to participate in Operation WhiteBerry in general, or the Mail Notification model in particular. This framework provides a complete set of development resources, including:

- The MailMeAnywhere open-source software distribution center and development forum at `http://www.MailMeAnywhere.org`.
- The LEAP Forum at `http://www.LeapForum.org`
- The EMSD Organization at `http://www.EMSD.org`

Each of these forums hosts several mailing lists which facilitate various forms of participation. `MailMeAnywhere` hosts a public forum for the collective development and enhancement of LEAP protocol engines and integration tools, while `EMSD.org` is a development forum for the EMSD protocol.

`LeapForum.org` is oriented towards coordinating usage of the LEAP protocols from a business development perspective; it provides a means for organizations and individuals to announce their participation in Operation WhiteBerry, seek out partners, and coordinate cooperative effort.

The EMSD User Agent source code is readily available at `http://www.MailMeAnywhere.org`. Alternative versions of the IMAP Server Agent software are available in source form from both the University of Washington and Carnegie-Mellon University; both versions are also readily available at `http://www.MailMeAnywhere.org`.

20.4.1 Open-Source Software Availability

Both the EMSD and IMAP software is freely available in source code form. In particular, the EMSD software is available under the terms of the GNU General Public License (GPL). The GPL places no restrictions on use of the software for development and proof-of-concept purposes.

However the GPL places certain restrictions on commercial usage of the software, and such usage may be in violation of the GPL. Device manufacturers are advised that they may need to purchase professional license agreements in order to make use of the EMSD and IMAP protocol engines in commercial contexts.

Chapter 21

Lessons From History: Comparitive Case Studies

21.1 Introduction

In general, protocols can have widely differing degrees of industry tenure. Some protocols achieve widespread adoption and usage, and persist as long-term industry standards. Others never achieve widespread acceptance, or else have very short lifetimes before becoming obsolete or being eclipsed by competing protocols.

There is no definitive set of circumstances which will guarantee either the success or failure of any particular protocol. Protocols succeed or fail as a result of a number of technical, cultural and business factors, each of which has complex effects and unpredictable results. The technical merits of the protocol play an important role; however, technical superiority alone is no guarantee of success.

The LEAP Manifesto is about a particular set of protocols: the Lightweight & Efficient Application Protocols, or LEAP. These protocols have been designed to address a particular industry need: the need for a set of highly efficient communications protocols. They are intended to be the foundation of an entirely new industry: the Mobile Messaging industry.

It is our hope and our intention that the protocols succeed in this purpose. It is our hope that these protocols become enduring industry standards, that they play a key role in the growth of the Mobile Messaging industry, and that they bring long-lasting benefits to the industry and the consumer.

Those factors which influence the success or failure of protocols are therefore of great interest to us. In this article, we identify and discuss those characteristics of protocols which contribute to their success or failure. In this regard we can learn a great deal from history. This article therefore also includes several case studies of successful and failed protocols.

21.2 Characteristics of Successful Protocols

Although there are no guarantees, history shows that successful protocols tend have certain characteristics in common. By a ”successful” protocol, we mean one which becomes accepted as an industry standard in the face of competing protocols, endures as an standard in the long term, and serves to promote the growth of the industry.

Some of the characteristics which improve a protocol’s chances are:

1. **Good Technical Design.** The more a protocol satisfies the technical requirements of the industry, the better its chances. This means that the protocol must primarily be an engineering construct, not a business one.

2. **Open Development and Maintenance Processes.** It is preferable for the protocol to be developed and maintained by open processes. This means that mechanisms should exist for public commentary on the protocol, and the protocol maintenance process should allow the participation of all constituencies that are affected by the protocol.

3. **Open Availability Process.** The protocol should be published and made available in a way that ensures that it is freely, easily and permanently accessible to anyone who wishes to use it.

4. **Freedom from Usage Restrictions.** There should be no restrictions on the use of the protocol. Anyone who wishes to base an application on the protocol should be able to do so without legal or financial hindrance.

Not all successful protocols have all these attributes. Nevertheless, the history of protocol development demonstrates that the more a protocol conforms to these attributes, the more likely it is to become an enduring industry standard.

The characteristics of various protocols are listed in Table 21.1. The protocols are arranged in groups, where each group represents a set of competing protocols. The table lists the major characteristics of each protocol, along with its ultimate success or failure. Note that there is only one eventual winner in each group.

Note how certain characteristics correlate well with eventual protocol success – for example, in all cases the winning protocol was free of usage restrictions. Other characteristics do not appear to influence the eventual outcome – for example, endorsement by a standards organization does not greatly increase a protocol's chances.

Use	Protocol	Usage/ Publications	Publications/ Availability	Open Source Implementation	Commercial Implementation	Development/ Maintenance	Standards Body Blessing	Long-term Success
Architecture	TCP/IP	Free	RFC / Open	Many	Many	Open	IETF (After)	Success
	SNA/DECnet	Licensed	Closed	None	Few	Closed	None	Fail
	ISO/OSI	Free	ISO/Limited	Few	Many	Limited	ISO	Fail
E-Mail	SMTP	Free	RFC / Open	Many	Many	Open	IETF (After)	Success
	X.400	Free	ITU/Limited	Few	Many	Limited	ITU	Fail
	MS:Mail CC:Mail	Licensed	Closed	None	Few	Closed	None	Fail
Security	S/MIME	Free	RFC / Open	Few	Few	Open	IETF	Fail
	PGP	Free	RFC / Open	Many	Few	Open	None	Success
WWW	HTTP/HTML	Free	RFC / Open	Many	Many	W3, Limited	None (W3 Later)	Success
Wireless	LEAP	Free	RFC / Open	Many	Few	Open	None	?
	WAP	Licensed/ Patented	Self/Limited	Few	Few (More)	Limited (WAP Forum)	None	?

Table 21.1: Protocol Success Stories

In the remainder of this article we provide case studies of several successful and failed protocols.

	Web Industry	**Mobile Messaging Industry**
Prerequisites	Wired Networks, Multimedia PCs, ...	Wireless-IP Networks, Palmtop PCs, ...
Open Standards	HTTP/HTML published as RFC-1945 & RFC-1866	ESRO/EMSD published as RFC-2188 & RFC-2524
Standards Organization	W3 Consortium	EMSD.org
Client Software	Browsers on all platforms freely available	User-Agent software for Windows CE etc. freely available
Server Software	Plug & Play, Freely available	Plug & Play, Freely available, or reasonable licensing
Service Providers	mosaic.com, etc.	ByName.net, etc.
Open-Source	Apache, Netscape	Ready for Open-Source

Table 21.2: Web Industry vs. Mobile Messaging Industry

21.3 Case Study I: The World Wide Web

Perhaps the best known example of a network industry based on a set of open protocols is the World Wide Web. In this section we identify the key factors that contributed to the extraordinary success of the Web Industry.

Our use of the LEAP protocols to catalyze the growth of the Mobile Messaging industry is to some extent modelled on the World Wide Web. We have identified many of the key factors that led to the success of the Web, and we have implemented those same factors in the case of the LEAP protocols and the Mobile Messaging industry. In many ways, we are attempting to reproduce the same circumstances in the Mobile Messaging industry that prevailed in the Web Industry, with the expectation that this will generate similar results.

Throughout this case study, as we discuss each factor that led to the success of the Web Industry, we will identify the parallel factor that we believe will lead to the success of the Mobile Messaging industry.

The factors that made the Web successful are summarized in Table 21.2, along with the corresponding factors for the Mobile Messaging industry.

21.3.1 Prerequisites

The Web Industry required that certain technological prerequisites be in place. These included such things as wired networks and multimedia PCs. The Web Industry could not come into widespread existence until these prerequisites were in place. For this reason the Web could not have materialized any earlier than 1993.

Similarly, the Mobile Messaging Industry has its own set of technological prerequisites. These include widespread and reliable wireless networks, affordable and miniaturized modems, and affordable and miniaturized mobile devices. All of these prerequisites did not become fully satisfied until 1999. Therefore the Mobile Messaging industry could not have materialized any earlier than 1999.

21.3.2 Open Protocol Specifications

The Web protocols are open and are based on the HTTP and HTML specifications, initially published as Informational RFC-1945[**?**] and RFC-1866[**?**] respectively.

Likewise, our Mobile Messaging protocols are completely open and are based on the ESRO and EMSD specifications, published as Informational RFC-2188[**?**] and Informational RFC-2524[**?**].

Hypertext multi-media information retrieval systems based on proprietary protocols existed before the Web came about. Now that HTML is widespread these proprietary protocols are all considered irrelevant.

Similary, many wireless e-mail and mobile messaging systems based on proprietary protocols exist today. We believe that they will likewise be considered irrelevant when LEAP becomes widespread.

There is another interesting parallel between the Web and Mobile Messaging protocols. The IESG inserted a critical note into the HTTP RFC cited above. More recently, the IESG has done exactly the same thing to both Mobile Messaging RFCs cited above. We regard this inclusion of critical notes into foreign (i.e. not originated by the IESG) RFCs to be a manifestation of the IESG's prejudice against external protocols – a frame of mind sometimes referred to as Non Invented Here syndrome.

21.3.3 Open Standards Organization

The Web Industry had the W3 Consortium. The Mobile Messaging industry has ESRO.org and EMSD.org.

21.3.4 Widespread Client Software

At an early stage in the evolution of the Web, client software was made widespread for all major platforms.

Similarly, a key requirement for the development of the Mobile Messaging industry will be to ensure that implementations of the LEAP protocols are made available for all major handheld devices. Specifically, for:

- PalmPilot
- Windows CE
- EPOC

We have made our implementations of device (i.e. client side) software for Windows CE and Win-32 available through Neda's website.

21.3.5 Widespread Server Software

At an early stage in the evolution of the Web, server software was made widespread for most major platforms.

Similarly, a key requirement for the development of the Mobile Messaging industry will be the ready availability of LEAP server software for platforms such as the Sun Solaris and Windows NT.

Our implementation of LEAP server software is readily available for the Sun Solaris platform now, and will become available for Windows NT shortly.

21.3.6 Open-Source Software

The availability of open-source software is extremely important for the creation of network industries.

Reference implementations of Web protocols were made widespread at an early stage in the evolution of the Web Industry.

Open-source software will play a similarly crucial role in the Mobile Messaging industry. Full implementations of published LEAP protocols will be made available as free software in open-source form. Our strategy is to release the software in stages as it becomes ready and available. The first piece to be released is our Open C Platform [**?**].

The next piece will be the ESRO protocol engine, followed by the EMSD device-side protocol engine, followed by the Message Center EMSD protocol engine. Further, we will integrate the LEAP protocol engines with various other free software. The distribution mechanism for the software will be through http://www.MailMeAnywhere.org.

21.3.7 Service Providers

At an early stage in the evolution of the Web, its industry leaders actively participated in the creation of web-based information and services. In a sense they acted as early Service Providers.

Similarly, it will be necessary to provide an initial set of Subscriber Services based on the LEAP protocols. To satisfy this need we are hosting a number of LEAP-based subscriber services.

21.4 Case Study II: Pretty Good Privacy

The history of the PGP (Pretty Good Privacy) data encryption system provides an excellent example of the success of a protocol developed entirely outside the traditional Standards Organization processes. PGP was essentially the creation of a single man: Phil Zimmermann. Armed with a vision and a belief in its value, Zimmermann single-handedly made PGP the dominant consumer encryption application – displacing the IETF alternatives in the process.

PGP is a protocol for electronic data encryption, which at the time of its development and deployment was in direct competition with S/MIME, a protocol being developed for the same purpose by the IETF. The key differences between PGP and S/MIME are summarized in Table **??**.

One of the major advantages that PGP enjoyed was being the creation of a single person. Small groups have an inherent advantage over large ones in any cooperative venture; as the size of a group grows, the difficulties of communication and coordination become increasingly challenging. Phil Zimmermann took this advantage to the limit: as a one-man operation, he enjoyed maximum efficiences of communication and coordination.

This is to be contrasted with S/MIME, which was developed by the IETF using classical Standards Organization processes. Under these processes, protocols are developed by committees, in the form of IETF Working Groups. Though this is a very reasonable way to conduct cooperative effort, it inevitably suffers from the friction associated with group action: communication overhead, time required to resolve misunderstandings or disagreements, and so on. Phil Zimmermann enjoyed an agility and an efficiency of action that the IETF process could not possibly match.

Both PGP and S/MIME were open protocols, without any usage restrictions, and so neither protocol had any advantage in terms of openness. Both protocols were also eventually published as RFCs. However, it is interesting to note that S/MIME, enjoying its privilege as an in-house IETF protocol, sailed through an early RFC publication process. PGP, on the other hand, was not published as an RFC until much later, when it had become clear that PGP had achieved widespread acceptance.

A further significant difference between PGP and S/MIME was the extent to which the two protocols were implemented in the form of open-source software. Both protocols were implemented as open-source; however PGP had much wider open-source support than S/MIME. This undoubtedly contributed significantly to the success of PGP, and this attests to the power of open-source in encouraging protocol acceptance.

S/MIME was developed and endorsed by the IETF, a formal Standards Organization. PGP enjoyed no such endorsement, and was developed entirely independently of any formal standards body. In spite of this, PGP has now become the *de facto* world-wide standard for electronic data encryption.

PGP is certainly not an isolated case. HTTP, the protocol which forms the basis of world-wide Internet communications, was also developed and achieved prominence independently of any formal standards body. These and many other protocols have become industry standards despite their lack of official endorsement.

The conclusions that we can draw from the history of PGP and other protocols are that standards organizations do not have an exclusive monopoly on creativity and innovation, and that official endorsement is not a prerequisite for protocol success.

The case of PGP and many other protocols supports our view that in general, innovation comes from small groups or individuals with vision, and not from committees, working groups, and Standards Organizations.

Phil Zimmermann has been an inspiration to every individual or small group with an idea they believe in, but who find themselves at odds with an entrenched Standards Organization. We believe the history of LEAP will provide similar inspiration.

Part IV

The Mobile Messaging Industry

Chapter 22

The Mobile Messaging Industry

22.1 Introduction

In recent years we have seen dramatic and rapid changes in the nature of interpersonal communications. As little as ten years ago, land-line telephone was the dominant medium for quick-turnaround, informal, interpersonal communication. In less than a decade, two new technologies have radically changed the character of person-to-person communications. The first of these technologies is e-mail. The second is the mobile telephone.

Ten years ago, these technologies were highly specialized communications capabilities, known to and used by a relatively small number of people. In just the last decade, usage of these technologies has grown explosively, propelling them from relative obscurity to mainstream acceptance among the population at large. E-mail and cell phones have been enthusiastically embraced by the public and by businesses, and have now become an everyday, indispensable aspect of modern life.

22.2 The Next Big Thing: Mobile Messaging

E-mail and cell phones each provide obvious and significant advantages over traditional land-line telephone communication, and this accounts for their extraordinary success. Among other things, e-mail provides the ability to conduct interpersonal communications on an asynchronous basis, and also provides the qualitative advantages of the written, over the spoken, word. And cell phones provide the self-evident advantages of mobile, real-time voice communications.

Individually, the major advantage of each of these two technologies is not shared by the other. The correct combination of the two technologies, however, can provide the major advantages of both. We refer to this combined technology as **Mobile Messaging**. Formally, we define Mobile Messaging as *the ability to send and receive e-mail messages from a mobile, hand-held or hand-carried device*. This capability is also sometimes referred to as Wireless Messaging, or Mobile E-mail.

The relationship between Mobile Messaging and conventional e-mail is somewhat similar to that between mobile phones and land-line phones: in each case mobility is being brought to a pre-existing, fixed-location, communications technology.

Mobile Messaging adds another component to a person's array of interpersonal communication options. Along with other enduring communications media such as conventional telephone and e-mail, voice mail, Fax etc., it provides a unique set of advantages to the user.

Early: Specialized	Later: Generalized
Medical Industry	Mobile Professionals
Public Safety	Apartment Managers
Emergency Professionals	Expectant Fathers
Drug Dealers	Rich Brats (90210)
Field Service Technicians	Soccer Moms
Financial Industry	

Table 22.1: Mobile Messaging End-Users

22.2.1 The Mobile Messaging End-User

Usage of Mobile Messaging among end-users is expected to evolve in a similar way to the historical evolution of One-Way Paging. As was the case with One-Way Paging, early usage of Mobile Messaging is expected to be mostly among specialized users, such as emergency service professionals and field service professionals.

Usage is then expected to spread quickly to more general users such as mobile professionals, soccer moms, and so on. Some examples of typical early specialized, and later generalized, users are provide in Table 22.1.

22.3 Comparison to Paging

Modern desktop e-mail applications typically expect a fast Internet connection, and always expect a large amount of disk space in order to store e-mail messages of arbitrary size. The general trend in desktop e-mail has been towards being able to exchange ever larger and more complex messages, including graphics, sound, and video.

The paging network resides at the other end of the messaging spectrum. The data transfer capability of paging has progressed from purely numeric messages to short alphanumeric messages, but still remains very limited. The first paging systems were one-way only, but recently some proprietary two-way paging systems have become available, in which the paging device can both send and receive messages.

In terms of data transfer capabilities, Mobile Messaging occupies an intermediate role between conventional desktop e-mail and two-way paging. Table 22.2 compares the basic characteristics of desktop e-mail, paging, and Mobile Messaging. Note that e-mail and paging each have critical advantages that the other does not. For the transfer of ordinary (i.e. not excessively long) messages, however, Mobile Messaging provides all the principal advantages of both desktop e-mail and paging.

22.4 Timeliness of Mobile Messaging

All the basic technological and cultural requirements are now in place to allow the Mobile Messaging industry to come into existence and reach its full potential. The basic requirements for widespread Mobile Messaging are:

- Wireless Networks: must be reliable, widespread and affordable
- Wireless Modems: must be available in mobility-enabling form factors, and affordable
- Mobile Devices: must be available and affordable
- End-Users: must be ready to accept Mobile Messaging capability

	Desktop E-Mail	Traditional Paging	Mobile Messaging
Message Structures (From:, Subject:, Date:)	Yes	No	Yes
Mainstream & Open (Any device works with any message center)	Yes	No	Yes
Intranet Provider Supplied Account (generally)	Yes	No	Yes
Service Provider (ISP) Supplied Account	Yes	Yes	Yes
Mobile & Wearable	No	Yes	Yes
Urgent/Pushed Message Delivery	No	Yes	Yes
Access Through Ordinary Telephone	No	Yes	Yes
Good For Long Messages	Yes	No	No

Table 22.2: Mobile Messaging vs. Traditional Paging

All of these pre-requisites now exist. Wireless networks are widely established and growing. Wireless modems are now fully capable of supporting Mobile Messaging. A multiplicity of end-user devices suitable for Mobile Messaging exist, such as PalmPilot, Windows CE devices and pagers.

And finally, the cultural climate is ripe: end-users are familiar with and have accepted the benefits of e-mail and cell phones. The step to acceptance of mobile e-mail is a very short one.

22.5 Market Forecasts

The LEAP Manifesto is not a business plan, and our concern is not the making of business profit. Instead, throughout the series of articles that make up the manifesto, we are concerned with what is required to create and build a healthy and vigorous Mobile Messaging industry, that will bring the greatest benefits to the industry participants and to the consumer. For this reason, quantitative market forecasts are not of any great relevance.

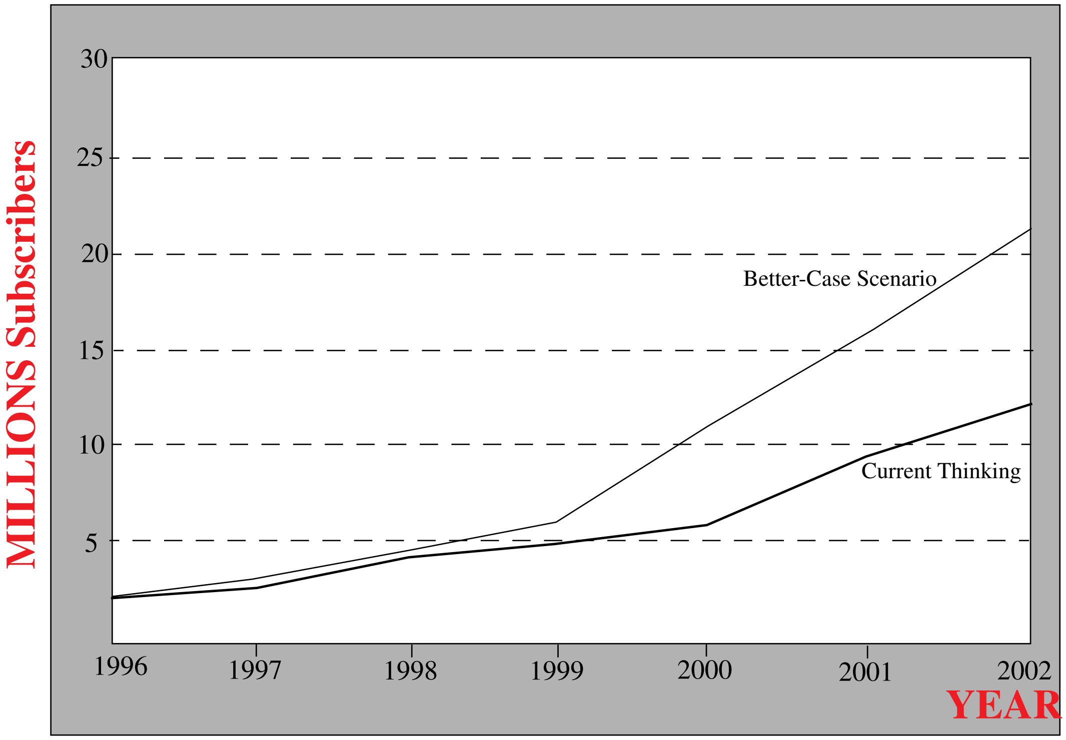
For those interested in such matters, however, we can say that the market for Mobile Messaging is extremely large. Market surveys repeatedly show that electronic messaging is the dominant application for both local area networks and wide area networks. Intranet operators and Service Providers repeatedly cite e-mail as the number one user requirement. Similarly, in the wireless arena, there is general consensus that Mobile Messaging is the primary application for wide area wireless data networks.

With regard to the actual size of the market: nobody knows. The Mobile Messaging industry is extremely complex, from both a technological and a business standpoint, and any market projection is at best an order-of-magnitude estimate based on a particular set of assumptions. Nevertheless, there is certainly no shortage of market projections, and all agree that the forecast for growth of wireless data and mobile messaging is phenomenal. In the following paragraphs we provide a few example data points.

Figure **??** shows the projections of the Yankee Group for the growth in numbers of wireless and mobile data subscribers. In this figure, the thin (lower) line represents the current thinking of the Yankee Group. The thick (upper) line represents their better case forecast scenario.

In their 1998 study, the Yankee Group estimated that in 1999 there were already almost 7 million wireless and mobile data subscribers. They forecast that by the year 2002 this number would increase to over 21 million subscribers, using numerous independent networks. Every one of those subscribers is likely to use Mobile Messaging in some form. The Yankee Group also forecast that sales of two-way pagers will reach 5 million in 2000, 24 million in 2001, and 54 million in 2003.

Wireless Mobile Data Market Forecast



SOURCE: the Yankee Group, 1998

Figure 22.1: Wireless Mobile Data Subscribers

A recent study by Killen & Associates estimates that the North American market alone for carrier services, equipment and software for wireless internet applications will jump from $2.7 billion in 1999 to $37.5 billion by 2002. They estimate that the global revenue for wireless access to the Internet and Intranet centered services, equipment and software will jump from $2.2 billion in 1999 to $10.0 billion by 2002.

Figure **??** shows the projections of Killen & Associates for the growth of global revenue for wireless access to the Internet and Intranet centered services, equipment and software.

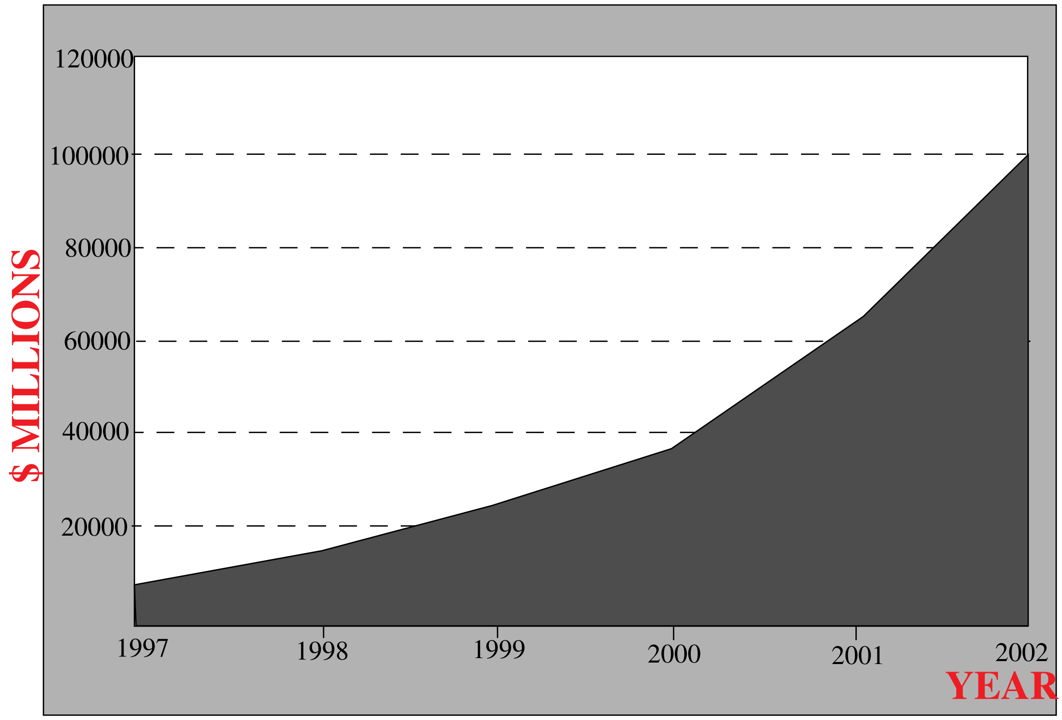
Whichever set of numbers you choose to believe, it is clear that the Mobile Messaging industry is destined for a spectacular future. The consequences of widespread wireless data communications capability are going to be enormous, in technological, cultural, and economic terms.

22.6 Current Status of the Mobile Messaging Industry

Despite its technological timeliness and enormous potential market size, the Mobile Messaging industry has yet to be realized in any meaningful way.

Numerous vendors claim to have some sort of Mobile Messaging product and service in place. Since most wireless networks have already converged towards IP, it is often fairly easy to get short messages to users over wireless networks, using simple custom protocols. These various systems all claim to be open; however, none of them work together, and they cannot make use of each other's assets. The result of this is that the Mobile Messaging arena is

The Global Revenue for wireless access to the Internet and Intranet-Centered Services,
Equipment, and Software: 1997 to 2002



SOURCE: Killen & Associates, Inc.

Figure 22.2: Global Wireless Internet Revenues

populated by various closed, non-interoperating products and services, and is currently dominated by a relatively small number of closed systems. None of these closed systems has a long-term future.

Most Mobile Messaging services are built on existing infrastructures and capabilities, and are available through paging and cellular phone companies as an extension of current services. However, building wireless messaging capability with protocols such as ReFLEX and IS-136 is at best a stopgap measure, because of the inherent size and structure limitations of those protocols. Moreover, these protocols are not IP based, and cannot benefit fully from the coming IP convergence.

Research In Motion (RIM; www.rim.net) has created an excellent, though closed, Mobile Messaging solution with its Blackberry(tm) mobile e-mail system. While RIM is currently leading the industry with its combined product and service offering, other manufacturers and service providers are moving quickly to catch up.

22.7 Differences among Mobile Messaging Providers

In short, the Mobile Messaging industry is seeing general agreement on the general concept of e-mail, but a complete lack of agreement or standardization on common technologies and protocols. Mobile Messaging providers are distinguished not only by differences in their services and products, but also by differences in:

- Physical networks and methods of data transfer

- Interface software
- Point-to-point routing software
- End-to-end and data reliability software

This high degree of heterogeneity and lack of interoperability within the industry acts to constrain its growth. Users are compelled to invest in one of several closed solutions from a particular group of service providers, and must then remain committed to that group of service providers.

Growth of the industry would be greatly enhanced if users were able to invest in a set of Mobile Messaging hardware products, then move freely from one service provider to another. This reduces the capital investment risk to the purchaser, thereby facilitating and encouraging his investment. Also, by decoupling the various components of a Mobile Messaging solution from one another, competition within each component is encouraged, resulting in better products and services. All of this serves to increase the vitality and growth rate of the industry as a whole.

22.8 The Fundamental Obstacle: Lack of Inter-Operability

Existing implementations of Mobile Messaging are based on closed and proprietary systems, services and devices. End-to-end transmission of delivery of messages requires smooth interoperability among networks, service providers, and devices. This is accomplished by means of protocols, which define the required interoperability. However, today's Mobile Messaging solutions are based on protocols which are closed and proprietary to the networks and/or service providers, and which cannot interoperate with other networks. Therefore the message centers and devices for a particular Mobile Messaging solution do not interoperate with those for a different solution and/or network.

While RIM's Blackberry(tm) service provides a model for wireless messaging, its closed design inhibits its adoption by the industry at large.

The current status of the industry is illustrated in Figure 22.3, which shows the interoperability relationships among various systems, networks, and devices. As the figure shows, each Mobile Messaging system is closed and unable to interoperate with the other systems.

22.9 The Key Enabling Requirement: A Standard Protocol

This lack of standardization within the industry inhibits its growth. In order for the industry to reach its potential, there must be complete interoperability among all components of a complete Mobile Messaging solution. All of the current and future wireless networks must interoperate easily, like the Internet itself, as a single network.

In order for this to happen, the Mobile Messaging industry must adopt a single, standard set of open protocols. Ultimately, network industries are based on protocols, not on specific products or services. The key enabling and driving force is the underlying protocol. Until a single open standard is adopted, growth of the Mobile Messaging industry will remain inhibited.

22.10 Protocol Requirements

The correct protocol will define a universal set of rules which provide an efficient, high-reliability, end-to-end data path for Mobile Messaging.

Existing Internet e-mail protocols are not adequate for Mobile Messaging. They are not bandwidth efficient, nor do they properly support the ”push” model for delivery of time-critical messages. In the “push” model, the message

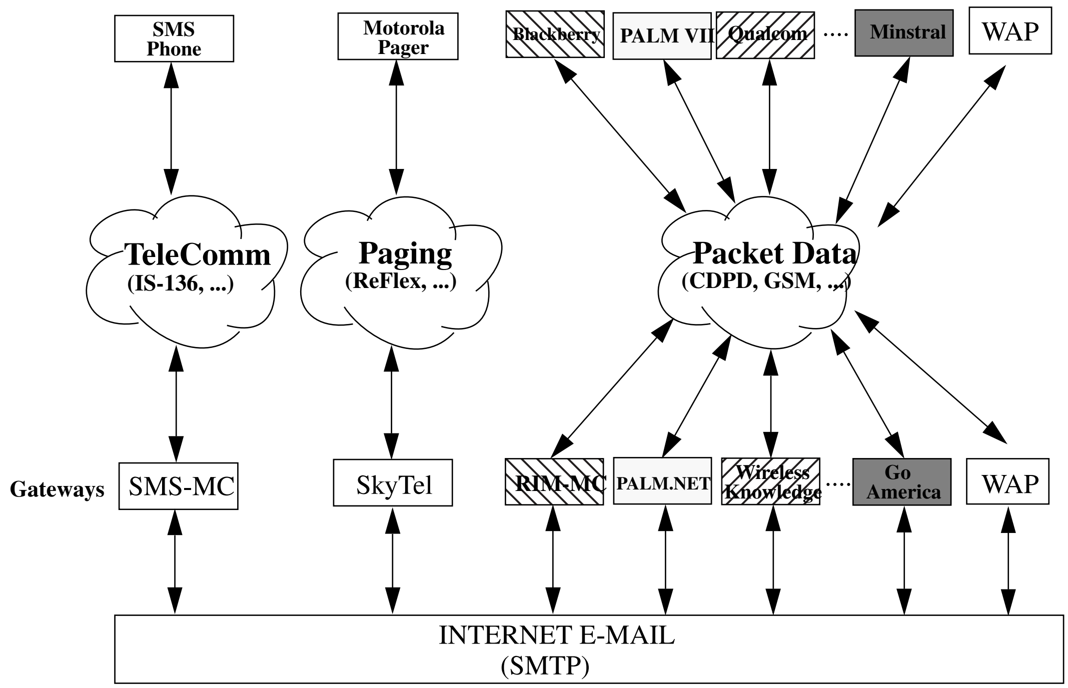


Figure 22.3: Existing Mobile Messaging Systems

is delivered directly and immediately to the recipient, like a pager message. In the case of existing Internet e-mail protocols, messages must wait on a server until they are picked up or "pulled" by an explicit action on the part of the recipient.

At Layer 3, it is clear that all data networks, both wired and wireless, are now converging on Internet Protocol (IP). The importance of Internet access ensures that this convergence will continue. Therefore a basic requirement of the protocol is that it define message transmission over wireless IP.

In addition, the protocols must adequately address the particular requirements of Mobile Messaging; specifically, the need for bandwidth efficiency and battery efficiency of mobile platforms.

In order to be effective as an industry catalyst, the protocols must be open. This means several things. First and foremost, this means that they must be patent-free, so that there are no restrictions associated with use and implementation of the protocols. This also means that they must be published as Internet RFCs. RFC publication ensures unrestricted and permanent availability of the protocols. The protocols must also be maintained by public working groups, to ensure that the protocols are open to commentary and influence by any industry constituency that may be affected by them.

Based on all these requirements, it is clear that the needed set of protocols is missing. We claim that the LEAP protocols, described in detail elsewhere in *The LEAP Manifesto*, are the answer to this need.

The effect that the LEAP protocols will have on the Mobile Messaging industry is illustrated in Figure 22.4. As this figure shows, LEAP will provide complete openness of interoperability among Mobile Messaging devices, message centers, and wireless networks. This is in sharp contrast to the closed, non-interoperating model of Figure 22.3.

LEAP will thus have the effect of unifying the entire Mobile Messaging industry under a set of open Internet Protocol ("IP") standards and protocols so that, in the manner of the World Wide Web, all of the Mobile Messaging networks will effectively operate as one.

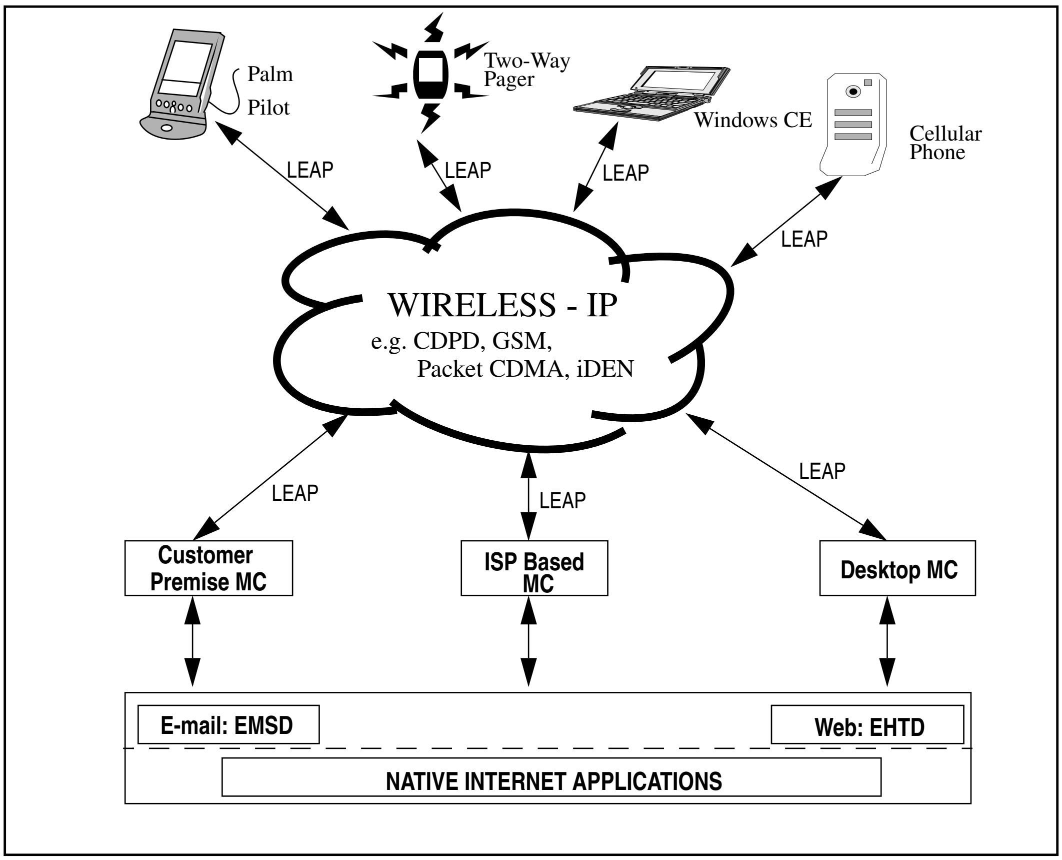


Figure 22.4: Open Mobile Messaging

Part V

APPENDIX

Appendix A

WhiteBerry and Bluetooth: General Information

A.1 Device Options for the Mobile Professional

The mobile professional of today can choose to carry any of a wide variety of communications and productivity devices. Throughout this article, however, we make the basic assumption that carrying separate cell phone and PDA devices will remain a viable and important option for a large class of mobile users.

In this appendix we provide a general discussion of the device options available to the mobile end-user, and we provide justification for this assumption.

A.1.1 Device Options for Cell Phones and PDAs: Integrated vs. Specialist

Today's mobile professional has at least three major communications and productivity requirements:

- **Voice communications**. The user requires voice communications for synchronous person-to-person interaction, and also for asynchronous voice messaging. This requirement is usually provided by the user's cell phone.
- **Personal information management (PIM)**. The user requires the usual set of information management tools, including calendar, address book, memopad and task list. These are most often provided by some sort of PDA (Personal Digital Assistant) device.
- **Mobile Messaging**. For those situations in which data messaging is more suitable or appropriate than voice interaction, the user requires Mobile Messaging capability, in particular e-mail. This can be provided by a stand-alone wireless e-mail device, or by integration of the e-mail application into the user's cell phone or PDA.

If we leave Mobile Messaging out of the equation for the moment, and focus on just the voice communications and PIM requirements, our mobile professional has two basic choices:

- Carry separate and independent cell phone and PDA devices
- Carry an integrated cell phone/PDA device

Until quite recently, the mobile professional would have been obliged to carry a physically separate cell phone and PDA, since neither of these could provide the essential functionality of the other. Now, however, there are a variety of integrated cell phone/PDA appliances available in the marketplace, which combine voice communications and PIM capabilities into a single integrated device.

This phone/PDA integration has been approached from both the phone side and the PDA side. From the phone side, a number of cell phone manufacturers have marketed integrated phone/PDA models, including the Ericsson R380s, Kyocera Smartphone and QCP 6035, Mitsubishi/Trium Mondo, Motorola Accompli 009 and Accompli A6188, Nokia 9000, Microsoft/Sagem WA3050, and Samsung SCH-i201. And from the PDA side, Handspring has recently marketed its VisorPhone device, a snap-on module that turns its Visor handheld PDA device into a cell phone.

The great advantage of the integrated cell phone/PDA is the obvious convenience to the user of having both of these functions provided by a single device. However, this convenience is offset by certain disadvantages.

The cell phone/PDA is in many ways an integration mismarriage. Note that from a user interface point of view, cell phones are primarily ear and mouth devices, while PDAs are primarily hand and eye devices. For this reason the user interface and form factor requirements of the cell phone and the PDA are fundamentally different, and frequently in conflict. Furthermore, the cell phone represents the culture and traditions of the telecommunications industry, while the PDA represents those of the data communications industry. For all these reasons, marrying these two devices together requires major engineering compromises and tradeoffs.

As a result of this, the integrated phone/PDA lacks the form-follows-function elegance of the best of the cell phones, or the best of the PDAs. Though some of the models listed above demonstrate considerable ingenuity of design, there is no integrated phone/PDA available today that can match the functional or aesthetic integrity of either a specialized cell phone, or a specialized PDA.

This, of course, is a manifestation of a general principle: the features and performance of a multi-purpose device can never match those of a specialized device, when judged in the specialist arena. The Swiss Army Knife has undoubtedly earned its place in the world, but there are many situations in which the user is better off with a toolbox full of specialist devices.

In addition to this general consideration, there are various other drawbacks to the use of an integrated cell phone/PDA. For example, a dead battery is a constant hazard for any cell phone user, because of the inherent high-power requirements of cell phone usage. In this eventuality the multi-device user can simply use a payphone or a client's phone, and look up the necessary numbers in his PDA address book. The integrated device user, on the other hand, has a much bigger problem, since he has now also lost his address book functionality.

A detailed comparison of integrated versus specialist cell phone/PDA functionality is outside the scope of this article. Suffice it to say that each has its own advantages and disadvantages, and neither is superior to the other under all circumstances. Which of the the two options is most appropriate for a given user depends on the usage patterns and preferences of that particular user.

Users who place a premium on convenience of portability will be more inclined to carry an integrated device, while those who place a premium on device specialization will be more inclined to carry independent cell phone and PDA devices. The integrated device and multi-device options are both viable alternatives, and each has its place in the mobile device marketplace.

A.1.2 Device Options for Mobile Messaging

Let us now add Mobile Messaging back into the mix. In terms of device integration, the mobile professional has four major options:

1. **Messaging via specialist device**. I.e. a specialized Mobile Messaging device, such as the RIM BlackBerry device

2. **Messaging via cell phone**. I.e. a cell phone with messaging capability, such as a WAP-enabled phone, or a phone with SMS capability

3. **Messaging via PDA**. I.e. a wireless-enabled PDA with bundled e-mail application, such as the integrated Palm PDA/Novatel wireless modem marketed by OmniSky

4. **Messaging via integrated phone/PDA**. I.e. an integrated cell phone/PDA with bundled e-mail application, such as the Nokia 9000 device

All of these are viable options, and all are in use today. Each has its own set of advantages and disadvantages, but in this article we will focus our attention on option 3 above, in which the user receives Mobile Messaging functionality via his PDA.

The great advantage of this is that, unlike the cell phone/PDA integration, this is a particularly smooth integration. The PDA is essentially a general-purpose data processing and user I/O device; and from this perspective e-mail is just another software application that fits perfectly into the PDA paradigm.

Certainly, integration of e-mail into a PDA is far more logical and practical than integration of e-mail into a cell phone, so for those users who choose to carry a cell phone and a PDA, option 3 above usually makes more sense than option 2.

Thus for the user who chooses to carry a PDA, there is little to be gained by also carrying a specialized wireless e-mail device – whatever advantage the specialist device can offer is likely to be more than offset by the portability convenience of having the e-mail functionality provided by the PDA.

Users who carry an integrated cell phone/PDA device will, of course, be most inclined to select option 4 above. But for those users who prefer to carry separate cell phone and PDA devices, we see that option 3 provides several major advantages, and for this reason is the preferred choice among mobile professionals.

A.1.3 Mobile Messaging via PDA: One Major Disadvantage

However, Mobile Messaging via the PDA has one major disadvantage. The PDA must be wireless-enabled, and this requires an additional device: the add-on wireless modem. This is inconvenient for the user in two respects. First, the add-on modem more than doubles the weight and dimensions of the PDA, greatly diminishing its portability, and relegating it from shirt pocket to jacket pocket. Second, the user must now maintain battery charge for two separate devices, requiring the use of two charging cradles, etc.

Eventually the modem functionality may become absorbed into and an integral part of the PDA, but we believe that this development is still some time away. For the moment the PDA and the modem are two separate devices, and will remain so for the immediate future.

In this article we describe how Mobile Messaging capability can be provided to the user who carries separate cell phone and PDA devices, while eliminating the modem requirement altogether.

A.1.4 Summary: Cell Phone/PDA Integration a Viable Option

As we have seen, the mobile professional can equip himself with voice, PIM and messaging capabilities in a variety of ways, including various levels of integration. The WhiteBerry/Bluetooth scenario we describe in this article is one more of these.

We do not claim to be able to predict which of these various alternatives will be more or less successful in the mobile marketplace. But we believe that each alternative is a viable option, and each has its place. The particular device configuration a user selects will depend on the specific requirements and preferences of that user.

In particular, the WhiteBerry/Bluetooth solution is appropriate for those users who want the performance and features of best-in-class, specialized cell phone and PDA devices, and who wish to have Mobile Messaging capability provided via the PDA.

Appendix B

Glossary of Terms

– A –

access control	A network function that determines whether a particular mobile device is permitted to access the network.
adaptive retransmission algorithms	Used by self-adjusting timers to determine and dynamically set timers to effectively adjust data traffic in the event the link is slower than usual due to congestion or other network conditions.
Advanced Mobile Phone System (AMPS)	The North American analog cellular phone system. The spectrum allocated to AMPS is shared by two cellular phone companies in each area or region (geographic market). This system was deployed during the 1980s and today it and its variants represent nearly 85% of the cellular voice systems installed throughout the world.
AMPS	See Advanced Mobile Phone System.
analog signal	A continuously variable waveform (such as a sound wave) that can represent an infinite number of values; wireline telephone systems use analog signals to provide voice communication. The term "analog" is used to refer to telephone transmission and/or switching that is not digital. Analog signals are amplified rather than regenerated.
ANSI	American National Standards Institute. A group that certifies organizations developing US standards for the information processing industry. ANSI accredited groups participate in defining network protocol standards.
API	Application Program Interface.
Application Layer	Layer 7 in the Open Systems Interconnect (OSI) 7-Layer Network Layer Reference Model. Provides protocols and services required by particular user-designed application processes. Functions pertaining to particular user requirements and application service elements that can be used by more than one application are contained in this layer.
application program interface (API)	The interface between the subscriber unit and the multiple application subsystem. Generally consists of simple commands such as "open a connection." These commands are translated by the enabling layer into commands that either control the network connection or communicate data across the network.

ASCII	American Standard Code for Information Interchange.
ASN.1	Abstract Syntax Notation One. A specification language used to define interoperable interfaces, based on standard application layer protocols such as Message Handling System (MHS) or Common Management Information Protocol (CMIP).

– B –

backbone	The backbone is the part of the communications network that interconnects Local Area Networks (LANs) together inside a building or across a city or country. LANs are connected to the backbone via bridges and/or routers and the backbone serves as a communications highway for LAN-to-LAN traffic. A backbone can be a LAN, a Wide Area Network (WAN), or a combination of both dedicated to providing connectivity between subnetworks in an Enterprise-wide network.
bandwidth	The bandwidth determines the rate at which information can be sent through a channel: the greater the bandwidth, the more information can be sent in a given amount of time.
bps (bits per second)	A measure of data transmission speed.
broadband personal communications standards (PCS)	Consists of 120 MHz of new spectrum available for new cellular networks. Also known as wideband PCS.

– C –

CCITT	Consultative Committee for International Telegraph and Telephone. Now known as the International Telecommunications Union- Telecommunications Systems (ITU-TS).
CDMA	Code Division Multiple Access. Defines how a single channel can be allocated to support multiple users simultaneously.
CDPD	Cellular Digital Packet Data.
CDPD Forum	An industry trade association supporting the development of the Cellular Digital Packet Data (CDPD) commercial marketplace.
Cell	The geographic region in which Radio Frequency (RF) transmission from one fixed transmission site can be received at acceptable levels of signal strength using an omnidirectional or a directional antenna. In a cellular network, a cell represents a geographic area within which a particular bandwidth of radio frequency channels can be received at adequate signal strength.

Cellular Digital Packet Data (CDPD)	A wide-area, wireless data networking technology. CDPD is an open standard for using existing cellular networks for wireless data transmission. Packets of data are sent along channels of the cellular network.
cellular modems	A device that combines data modem and cellular telephone transceiver technologies in a single unit. This allows a user to transfer data on the cellular network without the use of a separate cellular telephone.
channel	A 30 kHz cellular voice channel. A voice-grade transmission facility with defined frequency response, gain, and bandwidth. Also, a path of communication, either electrical or electromagnetic, between two or more points. Also, channel is used to describe the distribution path into the market for value-added resellers.
circuit	The physical connection (or path) of channels, conductors and equipment between two given points through which an electric current may be established. Includes both sending and receiving capabilities.
circuit switched	The process of setting up and keeping a circuit open between two or more users, such that the users have exclusive and full use of the circuit until the connection is released.
CLNP	Connectionless Network Protocol.
CLNS	Connectionless Network Service.
clone	A mobile device, typically a cellular telephone, that claims to possess the same address identifier as another mobile device.
CMIP	Common Management Information Protocol [CCITT-X.711].
CMIS	Common Management Information Service [CCITT-X.710].
CMISE	Common Management Information Service Element.
code division multiple access (CDMA)	Defines how a single channel can be allocated to support multiple users simultaneously so that all mobile users transmit on the same frequency with different pseudo-random signals. Originally developed to support voice. One of two digital cellular standards in the US.
Comite Consultatif International Telegraphique et Telephonique (CCITT)	Consultative Committee for International Telegraph and Telephone. A standards organization that devises and proposes recommendations for international communications. Refer also to American National Standards Institute (ANSI).

compression	The removal of redundant or unnecessary information from a set of data.
confidentiality	A security process that ensures that the content of a transmitted message cannot be determined except by the possessor of a key associated with the message, even if the transmitted bits are observed.
connectionless network protocol (CLNP)	The standard OSI connectionless network protocol.
connectionless network service (CLNS)	An International Standards Organization (ISO) internetwork service.
COTS	Connection-Oriented Transport Service.
CRC	Cyclic Redundancy Check. These are used for the purposes of error detection.
CSMA/CD	Carrier Sense Multiple Access with Carrier Detection.
cyclic redundancy check (CRC)	A computational means to detect errors in frames transmitted between devices. The mathematical function is computed at the originating device before a frame is transmitted. Its numerical value is computed based on the content of the frame. This value is compared with a recomputed value of the function at the destination device. Refer also to Frame Check Sequence (FCS).

– D –

data compression	Reducing the size of a file of data by eliminating unnecessary information, such as blanks and redundant data.
data encryption	The processing of data under a secret key so that the original data can only be determined by a recipient in possession of the secret key.
Data Link Layer	Layer 2 in the Open Systems Interconnect (OSI) 7-Layer Network Layer Reference Model. Provides communication among two or more systems. The Data Link Layer performs error checking, addressing, and other functions that are required to ensure accurate data transmission between adjacent systems.
datagram	A finite-length packet with sufficient information to be independently routed from source to destination.
decompression	The restoration of redundant data that was removed through compression.

decryption	The processing of received data by the possessor of a secret key in such a way that the intended contents of the data are restored.
decryption key	The secret key required to decrypt received data.
default route	A routing table entry which is used to direct any data addressed to any network numbers not explicitly listed in the routing table.
demultiplexing	The process of distributing data received in a shared data stream to the several entities that share the data stream. Demultiplexing can occur at several layers of a protocol stack. Refer to Multiplexing.
DES	Data Encryption Standard. An encryption/decryption algorithm defined in FIPS Publication 46.
Diffie-Hellman EKE	A procedure to exchange electronic keys between communicating devices. Used in CDPD.
Digital Sense Multiple Access With Collision Detection (DSMA/CD)	A technique used by Cellular Digital Packet Data (CDPD) to arbitrate access to the reverse channel stream by more than one Mobile End System (M-ES).
DNS	Domain Name System.
Domain Naming System (DNS)	A mechanism used in the Internet for translating names of host computers into addresses. It is a network database system that provides translation between host names and addresses.

– E –

EHTD	Efficient Hyper Text Delivery: the component of the LEAP protocols which facilitates web browsing.
EMSD	Efficient Mail Submission & Delivery: the component of LEAP that addresses the Mobile Messaging application. EMSD is a messaging protocol that is highly optimized for the submission and delivery of short Internet e-mail messages.
EMSD-FS	EMSD Format Standard. EMSD-FS is a non-textual form of compact encoding of Internet e-mail (RFC-822) messages, which facilitates efficient message transfer.
EMSD-P	EMSD Protocol. EMSD-P is responsible for wrapping a limited size EMSD-FS message in a point-to-point envelope, and submitting or delivering it.

EMSD-SA	EMSD Server Agent.
EMSD-UA	EMSD User Agent.
encryption	The processing of data under a secret key in such a way that the original data can only be determined by a recipient in possession of the secret key. The process reduces the possibility of eavesdropping.
encryption key	The secret key used to perform the encryption process.
end system (ES)	Any computer not considered to be performing routing or bridging functions. In Internet terminology, an ES is a host; ESs are the actual physical and logical end nodes that exchange information.
end-to-end communications	Data delivered between a source and destination endpoint.
end-to-end connection	Connections between the source system and the destination system.
end-to-end data confidentiality	The provision of data confidentiality between the sender and receiver of a communication.
ESRO	Efficient Short Remote Operations: the efficient transport layer protocol for several LEAP applications. ESRO is a reliable connectionless transport mechanism, forming the foundation for the development of efficient protocols when TCP is too much and UDP is too little.

– F –

file transfer protocol (FTP)	Common Internet file transfer program that uses Transmission Control Protocol/Internet Protocol (TCP/IP)
firewall	A security router that filters traffic based on addresses.
FPF	Free Protocols Foundation. FPF is an independent public forum, dedicated to the support of patent-free protocols.
frame	A Link Protocol Data Unit.
FTP	File Transfer Protocol.
full-duplex	Capability of simultaneously transmitting and receiving.

– G –

gateway	A protocol conversion facility allowing connection of dissimilar communication systems.
GHz	GigaHertz, Hz x 109 (One million cycles per second).
Global Positioning System (GPS)	A system of satellites orbiting the earth used for navigation and location. Position on earth is determined by referencing multiple satellites in the system whose orbits are precisely known and thus provide a reference frame for measuring distances.
Global Standard For Mobile Communications (GSM)	A digital cellular standard used primarily in Europe and some other countries. Primarily a Time Division Multiple Access (TDMA) cellular voice service, though data services are offered over GSM. Formerly called Groupe Speciale Mobile.
GSM	Global Standard for Mobile communication.

– H –

half-duplex	Capability of transmitting or receiving, but not both simultaneously.
handoff	The process of passing a Radio Frequency (RF) connection to another available channel either at the originating cell site, a neighboring cell site, or in another service provider's coverage area. As the subscriber moves from one location to another, the connection is continually handed off from channel to channel and from cell site to cell site.
host	The computer providing file transfer and communications services to client computers.

– I –

ICMP	Internet Control Message Protocol.
IEEE	Institute for Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
intermediate system (IS)	A node that is connected to more than one subnetwork with a primary role of forwarding data from one subnetwork to another. In Internet terminology, ISs are known as routers that relay data packets from ESs to their intended destinations.
International Standards Organization (ISO)	Organization that determines standards for international and national data communications. The US representative to the ISO is the American National Standards Institute (ANSI).

Internet	A set of networks using a set of network-independent protocols for communication that have been defined under the aegis of the Defense Advanced Research Projects Agency.
Internet (Internetwork)	(1) The global collection of interconnected local, mid-level, and wide-area networks that use Internet Protocol (IP) as the network layer protocol. (2) Any connection of two or more local or wide-area networks.
Internet Protocol (IP)	A connectionless protocol that requires no prior call setup between the source and destination systems.
IP	Internet Protocol.
IS	Intermediate System.
IS-95	The CDMA (Code Division Multiple Access) standard.
ISO	International Standards Organization.

– J –

– K –

Kbps	Kilobits per second.
key	A quantity required for the execution of security algorithms that is intended to be used only by legitimate entities.
kHz	kiloHertz, Hz x 1000.

– L –

LAN	Local Area Network. A network that takes advantage of the proximity of computers to offer relatively efficient, higher speed communications than long-haul or wide-area networks. Electronic mail, messaging, file sharing and database access are common LAN applications.
latency	The slight delay in data delivery through a network.
LEAP	Lightweight & Efficient Application Protocols. LEAP is a set of high-performance, efficient protocols which are ideal for mobile and wireless applications.

Location Directory (LD)	The repository of information specifying the current Forwarding Address of a collection of mobile hosts to be accessed by the redirectors. For Cellular Digital Packet Data (CDPD), refer to Mobile Network Location Protocol And Home Mobile Data Intermediate System.

– M –

MAC	Medium Access Control.
MAC Layer	That layer of a distributed communications system concerned with the control of access to a medium that is shared between two or more entities.
message	A meaningful unit of information. A Protocol Data Unit (PDU) of defined format and purpose. Refers to the basic unit of data being transmitted.
Message Handling System (MHS)	A supplementary service that implements Message Store and Forward Message Handling. MHS is used within the Cellular Digital Packet Data (CDPD) network. MHS provides for distribution lists, return receipts and allows for gateways to proprietary message systems, including alternative delivery mechanisms such as telex or fax.
message transfer agent (MTA)	Performs Message Routing (e.g., to other MTAs) and message delivery to User Agents and Message Stores.
messaging API (MAPI)	An application protocol.
middleware	Software that isolates applications from the protocol requirements, hardware, and actual operations of a particular network. May perform protocol conversions between disparate systems. Refers to any software that resides between a client program and a server program. Middleware includes protocol stacks, network operating systems, and messaging systems.
MIME	Multipurpose Internet Mail Extensions. MIME is a specification for enhancing the capabilities of standard Internet e-mail.
modem	MOdulator-DEModulator. A device that translates computer information for transmission on both cellular and regular telephone lines.
modulation	The analog waveform of a transmitted digital signal.
Motorola Integrated Radio System (MIRS)	Technology developed by Motorola. ESMR is based on this technology. Nextel is the largest company deploying ESMR. (Now called iDEN).

MSC — Mobile Switching Center.

MTA — Message Transfer Agent.

MTS — Message Transfer System.

multicast — A one-way point to multi-point routing and relaying service that allows a message to be addressed and delivered to all members of a group, regardless of their current location.

multipath fading — The singal degradation that occurs when multiple copies of the same radio signal arrives at the receiver through different reflected paths. The interference of these signals, each having traveled a different distance, result in phase and amplitude variations. The radio signal processing in both the base station and mobile units have to be designed to tolerate a certain level of multipath fading.

multiplexing — The process of mixing data originating from several entities into a single shared data stream. Multiplexing can occur at several layers of a protocol stack.

– N –

Network Protocol Data Unit (NPDU) — Network Layer Protocol Data Unit. The NPDU comprises the Network Layer Service Data Unit and the Network Layer Protocol Control Information.

Network Service Access Point (NSAP) — The NSAP is an abstraction of a point where the Network Layer service user and the Network Layer service provider meet. Each Network Entity may be accessed through one or more NSAPs by its users. Each NSAP has one or more NSAP-Addresses.

node — A node is an entity that participates in network communication. Examples of nodes are End Systems and Intermediate Systems.

NPDU — Network Protocol Data Unit.

NSAP — Network Service Access Point.

NSAP-Address — The NSAP-Address uniquely identifies a Network Service Access Point (NSAP).

NSAP-Selector — A component of an Network Service Access Point (NSAP)-Address used to select the Network Layer service user. The NSAP-Selector is sometimes referred to as a Transport-Selector; however, a user of the Network Layer need not be a transport service.

– O –

open system	A system with publicly known protocols or architectures, one that requires no special protocols and is simple to implement.
Open Systems Interconnection (OSI) 7-Layer Network Layer Reference Model	An international standardization program facilitated by International Standards Organization (ISO) and Consultative Committee for International Telegraph and Telephone (CCITT) to develop standards for data networking that facilitate multivendor equipment interoperability. OSI includes the following seven layers: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, and Application Layer.
OSI	Open Systems Interconnection. A set of protocols designed to be an international standard method for connecting different types of computers and networks. Europe has done most of the work developing OSI.
OSI reference model	An ”outline” of OSI that defines the seven layers of the model and their functions. Sometimes used to help describe other networks. Refer to Open Systems Interconnection (OSI) 7-Layer Network Layer Reference Model.

– P –

packet	Network Protocol Data Unit
Patent-free protocol	A protocol which is functionally free from software patents.
PCS	Personal Communications Services. A wireless technology being developed in the US. Narrowband PCS refers to a new spectrum available for paging services. The digital signals of PCS are a series of rapid pulses, unlike analog’s steady flow.
PDA	Personal Digital Assistant.
PDU	Protocol Data Unit.
Personal Communications Services (PCS)	Refers to the next generation of cellular systems. Will include data services.
Personal Digital Assistant (PDA)	A hand-held or custom-built Mobile End System (M-ES).
PGP	Pretty Good Privacy. PGP is a freeware software which is used worldwide for e-mail encryption and authentication including file encryption and peer-to-peer Virtual Private Network (VPN).

Physical Layer	Layer 1 in the Open Systems Interconnect (OSI) 7-Layer Network Layer Reference Model. That layer of a distributed communications system concerned with the actual transmission of data across a physical medium. The physical layer provides a physical connection for transmission of data between two data link entities.
ping	An ICMP message that is sent out to a known IP address. If the signal arrives at the known address, it returns to its point of origin, confirming connectivity.
point of presence	An entry point for connection into the network.
Point-to-Point Protocol (PPP)	An Internet protocol that provides a method for transmitting datagrams over serial point-to-point links.
PPP	Point to Point Protocol,
primitive	An abstraction representing the communication between layer entities operating at different layers or between a layer entity and the management entity controlling it. There are four types of primitives in the OSI (Open Systems Interconnection) model)-request, indication, response, and confirm.
protocol	A set of procedures for exchanging information between peer entities. A formal description of message formats and the rules two computers must follow to exchange those messages. Protocols can describe low-level details of machine-to-machine interfaces, such as the order in which bits are sent across wire, or high level exchanges between allocation programs, such as the manner in which two programs transfer files across the Internet.
Protocol Control Information (PCI)	The part of the Protocol Data Unit that contains addressing, control, flags, and optional information needed by peer Layer Service Providers to provide the service. Refer to protocol data unit and service data unit.
Protocol Data Unit (PDU)	The unit of information that is exchanged by peer Layer Service Providers to implement the protocol that offers the service. The PDU consists of a Protocol Control Information and Service Data Unit.
PSTN	Public Switched Telephone Network. Actual telephone wires.
public switched telephone network (PSTN)	Public telephone network.

– Q –

QOS	Quality of service.

– R –

radio frequency (RF)	The frequency in Hertz (cycles per second) of the carrier frequency being transmitted or received.
RDP	Reliable Data Protocol.
reassembly	The process of combining a number of the Link Layer Service Data Unit (LSDU) into an SN-Data Protocol Data Unit (PDU) or SN-Unitdata PDU.
Reed-Solomon code	A forward error correcting code used in Cellular Digital Packet Data (CDPD).
Reed-Solomon forward error correction	Refer to Reed-Solomon code.
reliable sequenced delivery	The delivery of a set of Protocol Data Units (PDUs) from a source to a destination with no errors in any PDU, in the order transmitted, and without gaps or duplicates.
RFC	The internet's Request for Comments documents series. The RFCs are working notes of the Internet research and development community. A document in this series may be on any topic related to computer communication, and may be anything from a meeting report to the specification of a standard.
RFC Publication	The mainstream of Internet publishing procedure, ensuring that the protocols are freely, easily and permanently accessible to anyone who wishes to use them.
RIP	Routing Interchange Protocol. A protocol which may be used on internets to pass routing information between gateways. It is used on many Local Area Networks (LANs) and on some of the NSFNET intermediate level networks.
roam	Traveling outside of a local cellular coverage area.
ROSE	Remote Operations Services Element.
router	A special-purpose dedicated computer that attaches to two or more networks and examines received packets from one network. Based on the destination address of the packet, forwards it to the most appropriate next network.

RPC	Remote Procedure Call: specified in RFC-1831 [?] and RFC-1833 [?]

– S –

S/MIME	Secure MIME. S/MIME extends MIME and add security to the specification.
SAP-address	Service Access Point Address.
SNDCF	Subnetwork Dependent Convergence Function.
SDU	Service Data Unit.
secret key	A key whose value is not disclosed to unauthorized entities.
security	The set of functions concerned with ensuring that only authorized users of the network can use the network, and only in authorized ways.
segmentation	The process of fragmenting an SN-Data Protocol Data Unit (PDU) or SN-Unitdata PDU into a number of Link Service Data Units (LSDUs).
Serial Line Internet Protocol (SLIP)	A protocol used between a host and the service provider to transmit IP datagrams through a serial port.
Service Data Unit (SDU)	When a user of a layer service requests the Layer Service Provider to transfer data to a remote service user, the user data is termed a Service Data Unit. The Layer Service Provider attaches a small header, called the Protocol Control Information, to the user data. The combination of the Protocol Control Information and SDU is called a Protocol Data Unit.
SLIP	Serial Line Internet Protocol.
SNDCP	Subnetwork Dependent Convergence Protocol. A Network Layer protocol that supports subnetwork convergence.
SNMP	Simple Network Management Protocol. The Simple Network Management Protocol is the Internet's standard for remote monitoring and management of hosts, routers and other nodes and devices on a network.
Sockets Interface	The Sockets Interface, introduced in the early 1980s with the release of Berkeley UNIX, was the first consistent and well-defined application programming interface (API). It is used at the transport layer between Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) and the applications on a system. Since 1980, Sockets has been implemented on virtually every platform.

specialized mobile radio (SMR)	(1) A wireless data service that operates in the 806 to 821 MHz and 851 to 866 MHz bands. (2) Analog trunked radio systems primarily used for dispatch applications. (3) Dispatch radio systems combined into the national ESMR network.
synchronization	The process of achieving a common interpretation of a transmitted bit stream between more than one entity at the same point in the bit stream. Synchronization may be required at each layer of a protocol stack.
synchronous	The communications method where characters are spaced by time, not by start and stop bits. Synchronous transmission of a message requires fewer bits and less time than asynchronous transmission. Refer to asynchronous.

– T –

TCP	Transmission Control Protocol. A transport layer protocol for the Internet. A connection oriented stream protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol. A set of protocols developed to link computers and data communications equipment across many kinds of networks.
TDMA	Time Division Multiple Access.
Telecommunications Industry Association (TIA)	Standards organization.
telemetry	A communications system for the transmission of short digital or analog data that monitors status information for a remote process, function or device.
Telnet	The virtual terminal access protocol used by the Internet. It allows a user at one site to interact with a remote timesharing system at another site as if the user's terminal was connected directly to the remote computer.
threshold	The minimum value of a signal that can be detected by the system or sensor under consideration.
throughput	The actual amount of useful and non-redundant information that is transmitted or processed. Throughput is the end result of a data call. Throughput is a measure of the efficiency of a communications link.
TIA	Telecommunications Industries Association.

Time Division Multiple Access (TDMA)	A medium access control scheme where each member of a device population is assigned transmission slots separated in time.
Transmission Control Protocol (TCP)	A connection-oriented protocol that provides reliable end-to-end service. Responsible for ensuring data integrity end-to-end, given that Internet Protocol (IP) is a connectionless protocol.
Transport Layer	Layer 4 in the Open Systems Interconnect (OSI) 7-Layer Network Layer Reference Model. Transpor Protocol 4 (TP4), Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) reside at this layer. Connection-oriented service provides reliable and transparent transfer of data between cooperating session entities.

– U –

UDP	User Datagram Protocol.
User Agent (UA)	Related to the messaging supplementary service. A user agent is an application entity that acts on behalf of a user and interfaces to a local message transfer agent. A user agent provides submission and delivery services and other message handling operations such as composing, forwarding/distributing, replying, examining.
User Datagram Protocol (UDP)	A connectionless protocol that provides end-to-end communication with fewer capabilities than Transmission Control Protocol (TCP). Each UDP datagram is handled independently. UDP does not guarantee delivery of data.

– V –

vertical market application	An application that is industry-specific and typically very task-specific. Including systems for on-line reservations, point-of-sale, utility monitoring, or dispatch services. Such applications have been customized to meet the needs of a specific industry and its particular customers.
VMTP	Versatile Message Transaction Protocol

– W –

WAN	Wide Area Network. A network that links a wide geographical area.
WAP	Wireless Application Protocol. WAP is a specification for wireless data communications using hand-held devices such as mobile phones and palmtop computers.

wide-area-network (WAN)	A network that links a wide geographical area.
window segment size	A parameter used to control the flow of data across a connection.
wireless	Communication by means of electromagnetic radiation, particular in the Radio Frequency (RF) portion of the spectrum. Can refer to various technologies for transmitting data, including those used by ARDIS, RAM, or Cellular Digital Packet Data (CDPD).
wireless data network	A radio-based network for data transmission. Cellular Digital Packet Data (CDPD) is an example.
WTP	Wireless Transaction Protocol

– X –

X.25	A network-level interface protocol and the underlying Link Layer protocols for connecting to a packet-switching network. Routes packets of data through a network to destination nodes. Used to connect remote terminals to host systems; provides any-to-any connections for simultaneous users.

– Y –

– Z –

Index