

Wireless Application Development

**Issues and Guidelines
Paging Systems Emphasis**

Motorola Personal Networks Group

Motorola, Inc. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any warranties, express or implied, of merchantability or fitness for any specific purpose. Further, Motorola, Inc. reserves the right to revise this publication and to make any modifications to its content, at any time, without obligation to notify any party, person, or entity of such revisions or changes.

© 1999, 2000 Motorola, Inc. All Rights Reserved.

Personal Networks Group
3301 Quantum Boulevard
Boynton Beach, FL 33426-8622

T, Motorola, FLEX, ReFLEX, InFLEXion, and VoXML are trademarks or registered trademarks of Motorola, Inc.
All other brand or corporate names are trademarks of their respective owners.

Table of Contents

- 1.0 Introduction 1**
 - 1.1 Scope 1
 - 1.2 Opportunities 1
 - 1.3 Intended Audience 2
 - 1.4 Document Organization 2

- 2.0 Key Players in the Wireless World 3**
 - 2.1 Application Developers 3
 - 2.2 Paging Carriers 3
 - 2.3 Subscriber Device Suppliers 3
 - 2.4 Infrastructure Suppliers 4
 - 2.5 Information Content Providers 4
 - 2.6 Subscribers 4

- 3.0 Inside The Paging Infrastructure 5**
 - 3.1 System Overview 5
 - 3.2 Paging Terminals 11
 - 3.3 System Control Switches 13
 - 3.4 Transmitters 14
 - 3.5 Receivers 15
 - 3.6 Gateways and Servers 15
 - 3.7 Terminal and Distribution Networks 16

- 4.0 Subscriber Devices 19**

- 5.0 Protocols 21**
 - 5.1 Overview of Protocols 21
 - 5.2 Protocols Between Outside World and Paging Infrastructure 22
 - 5.2.1 From PSTN to Paging Infrastructure 22
 - 5.2.1.1 TNPP (Telocator Network Paging Protocol) 22
 - 5.2.1.2 TAP (Telocator Alphanumeric Protocol) 23
 - 5.2.1.3 SNPP (Simple Network Paging Protocol) 24
 - 5.2.2 Mail and Internet Protocols 24
 - 5.2.2.1 SMTP (Simple Mail Transfer Protocol) 24
 - 5.2.2.2 HTTP (Hypertext Transfer Protocol) 24
 - 5.3 Protocols Within Paging Infrastructure 25
 - 5.3.1 Terminal to Terminal 25
 - 5.3.1.1 TNPP (Telocator Network Paging Protocol) 25
 - 5.3.1.2 WMtp (Wireless Messaging Transfer Protocol) 25
 - 5.3.2 Terminal To System Control 29
 - 5.3.2.1 TNPP (Telocator Network Paging Protocol) 29
 - 5.3.2.2 WMtp (Wireless Messaging Transfer Protocol) 29
 - 5.3.3 System Control to Transmitters 30
 - 5.3.4 Receivers to System Control 30

5.4	Protocols Between Paging Infrastructure and Subscriber Devices	31
5.4.1	One-way Over The Air Paging Protocols.....	31
5.4.1.1	POCSAG (Post Office Code Standardization Advisory Group)	31
5.4.1.2	ERMES (European Radio Message System)	31
5.4.1.3	FLEX.....	31
5.4.2	Two-way Over The Air Paging Protocols.....	33
5.4.2.1	ReFLEX 50&&&&&&.....	33
5.4.2.2	ReFLEX 25.....	34
5.4.2.3	InFLEXion.....	35
6.0	Messaging Options.....	37
6.1	Broadcast Messaging	37
6.2	Peer to Peer Messaging	37
6.3	Multicast Messaging	37
7.0	Challenges of the Wireless World.....	39
7.1	Transmission Errors.....	39
7.2	Message Ordering	39
7.3	Coverage Problems	39
7.4	Message Latency.....	40
7.5	Network Capacity and Data Rate Limitations	40
7.6	Network Dependencies.....	41
7.7	Protocol Capability vs. Network and Device Capability	41
7.8	Security Issues.....	41
8.0	General Recommendations and Guidelines	43
8.1	Up-front Investigation.....	43
8.2	Application Architecture	43
8.3	Message Size and Data Packing.....	44
8.4	Data Representation	45
8.5	Data Compression and Encoding Techniques	45
8.6	Managing Latency.....	46
8.7	Error Handling.....	47
9.0	Future Trends.....	49
9.1	New Protocols.....	49
9.2	New Devices & Operating Systems.....	49
9.3	Support for Standards.....	49
10.0	Conclusions	51
11.0	Where to Find More Information	53
11.1	Internet Sites	53
11.2	Useful Text Books	53
11.3	Email Address	53
12.0	Acronyms and Glossary of Terms	55

1.0 Introduction

In the very recent past, major technology changes have been introduced which are transforming the way we live and work. Foremost among these are the technologies used to move information along the Internet, the “information super-highway”. Individuals and companies are in the early stages of understanding how to use the Internet to their advantage. Growth, over the last couple of years, has been staggering and promises to continue in this fashion for years to come.

As people become more dependent upon information in this mobile world, they are searching for ways to stay connected regardless of where they are. This explains the dramatic growth seen in the mobile communications products market. Motorola recognizes these trends and intends to be a major player in both the Internet and wireless communications markets.

One key to success in the wireless and Internet markets is identifying and delivering the information people want or need to use. This involves a number of key players, information content providers, public and private wireless carriers, suppliers of wireless infrastructure and subscriber devices, and software developers who can write the applications that extract, organize, and deliver the information to the user.

Tremendous opportunities exist for companies that successfully compete in the wireless and Internet markets. However, significant challenges exist to those who chose to enter the fray. This paper attempts to increase the reader’s understanding of paging systems. It also seeks to identify some of the challenges and pitfalls that application developers face in their attempts to create the wireless “killer apps” that will propel them to success, fame, and fortune.

1.1 Scope

This paper focuses on one piece of the wireless space, the paging sector. This is done primarily due to the author’s greater experience and familiarity with the networks and technology used in the paging industry. Paging systems and all the components needed to host wireless applications are in place today. It can be argued that paging networks are somewhat ahead of wireless voice networks in their ability to deliver data. Paging has always focused on sending data, whereas the primary focus of both cellular and Personal Communications System (PCS) networks has been to deliver voice.

1.2 Opportunities

Many developers ask why should they develop applications for a paging network? Why not just wait for higher speed technologies such as General Packet Radio Service (GPRS) and Third Generation (3G) networks? Several compelling reasons exist for developing applications for paging systems. First, paging networks are already in place with more capacity coming online in the very near future. Second, paging devices and services exist in the market today. Opportunities and demand for applications on paging networks are already there for those who know how to profit from them. Third, most experts expect that early implementations of new technologies will not deliver anywhere near the maximum theoretical data bandwidth. And fourth, availability of devices, billing issues, technological challenges and support by carriers may delay adoption of these new technologies.

Rather than wait for technologies that may not reach critical mass for several years, a compelling strategy would be to take advantage of current opportunities, develop applications for today's existing systems, and migrate to new technologies when the demand and system availability justify it. Many of the issues facing developers who plan to deploy their applications over paging networks are germane to any wireless network. Lessons learned in writing these applications can be carried over as applications are developed for the cellular and PCS networks.

1.3 Intended Audience

The primary audience for this document is software developers who are interested in developing wireless applications, particularly those applications that involve existing paging networks. In addition, this document will be useful for people who manage the networks and infrastructure used to deliver information to the end user. This document can help sensitize network and infrastructure managers to the issues faced by software developers in trying to write applications that are “well behaved” on the networks.

1.4 Document Organization

This document begins by profiling the key players in the wireless world. The issues facing these players help set the stage for discussions and recommendations that follow.

Next we describe the paging infrastructure in some detail. Many application developers are prone to believe that the infrastructure is simply a wireless pipe or connection between people and machines. The infrastructure has characteristics that must be understood in order to write successful applications. Following an introduction to the infrastructure, we discuss characteristics of wireless devices. These devices have properties that are different from traditional desktop computers that must be understood when designing wireless applications.

After we cover the paging infrastructure, we discuss protocols at a rather high level. The intent is to explain what they are and how they determine network and application behaviors. The intent is not to document the protocols. Protocols are described at the various points in the system????.

After we’ve discussed protocols, we focus on messaging options that should be considered when developing applications. Then we address the real challenges facing developers that are somewhat unique or at least more pronounced in the wireless environment. In addition to describing the challenges, we give some recommendations that should help ensure success in writing applications for the wireless world. This is followed up with a brief discussion of future trends, and ends with conclusions, a section that contains a list of acronyms and sources of additional information.

At several places within the document, we identify key issues or considerations of which developers should be aware. These are identified in italicized text in a framed box as illustrated below:

<i>Tip: This is an example of information that is especially useful to developers.</i>
--

Let’s begin by discussing key players.

2.0 Key Players in the Wireless World

Identifying key players in the wireless world can help you understand the issues that must be considered when developing applications for this environment.

2.1 Application Developers

Application developers are key players in linking data sources to the end user. Their role is to extract data from various sources and translate it into information that has value to a consumer. They must handle the technical details of extraction, summarization, transfer, and presentation. They must work within the constraints of the systems, networks, and devices used to process and deliver the information. They need to be knowledgeable about all of these components and how they interact.

Many developers are small companies or individuals who have embraced the promise of the wireless opportunity. Many have limited resources, both people and financial. Most have software development experience, but mainly in the wired world. Many have identified a niche market that they would like to exploit. They may have experience in a particular industry, but may have few contacts or experience with the other players in the wireless world.

Given their limited resources, developers cannot afford to develop applications using a trial and error approach. As the saying goes, “if you don’t have time (and resources) to do it right, how do you have time to do it over?”. Application developers must get it right the first time. A poorly written application will not be accepted, either by the carriers or by the consumer.

2.2 Paging Carriers

Paging carriers operate the infrastructure that delivers the payload from data sources to the subscriber. They are profit driven and closely scrutinized by the investment community. The services that carriers provide must cover the cost of doing business as well as generate a healthy profit. Therefore, they are very cautious when it comes to initiating new services that may interfere with their existing revenue generating services.

Carriers are aware of the growing interest in the two-way wireless Internet market and are interested in hosting the “killer apps” that can increase their revenue stream. However, hosting smart two-way products on their networks is a new challenge, one that carriers are approaching with appropriate caution. Carriers require some assurance that, once on the network, the “killer apps” will be well behaved and they cannot be used in an abusive manner. This means an application should not generate excessive network traffic that can interfere with other normal messaging.

Some people feel that carriers will simply expand capacity if the demand requires it. Therefore, they contend that capacity concerns are not that critical. The truth is carriers face numerous challenges in raising capital for new infrastructure, locating new transmitter and receiver sites, and acquiring additional RF spectrum. Even if the capital is provided and spectrum is available, it takes time to expand networks to bring additional capacity online. An application that requires too much bandwidth will simply not be accepted by paging carriers.

2.3 Subscriber Device Suppliers

Device suppliers generate revenue by selling wireless devices and are anxious to find and host “killer applications” that increase demand for their products. Therefore, they have a vested interest in working with the other key players to remove barriers to success. Since the technology needed to design and build wireless devices is very sophisticated, device suppliers tend to be large corporations.

While device suppliers may have resources that can be applied to address problems related to wireless communications, they do not control all the pieces of the system. For example, while device suppliers may be able to guide and influence the best advanced messaging configuration for the carrier's infrastructure, they do not control these networks. Similarly, they can try to influence infrastructure suppliers to add support for technologically advanced features, but they can't mandate or dictate the timing. Subscriber device suppliers do not have domain knowledge in all industries. Therefore, they must rely on other companies to identify and develop applications that make sense in a wireless environment. In short, device suppliers must work together with the other key players in order to achieve mutual success.

2.4 Infrastructure Suppliers

Infrastructure suppliers are benefited by technologies, like advanced two-way messaging, that create a demand for additional infrastructure. However, these advances in technology require continued product development. The development cycle for this equipment is typically many months. Infrastructure equipment is a capital investment that carriers expect will serve their needs for a very long time. Enormous challenges face infrastructure suppliers in trying to bring products to market that meet a quickly developing market opportunity, are highly reliable, and are backward compatible to existing systems and protocols.

Infrastructure products are very sophisticated. Therefore, infrastructure suppliers tend to be medium to large size companies. Paging infrastructure suppliers are not the only source of infrastructure components. Some paging carriers have actually developed pieces of their own infrastructure, rather than depend on the product development timing and priorities of established suppliers.

Infrastructure suppliers, subscriber device suppliers, and carriers have traditionally worked very closely together to establish standards and promote technologies that benefit all. The role of the independent developer is a new addition to the mix, and is yet to be well integrated into the symbiotic relationship.

2.5 Information Content Providers

Information is the fuel that drives the Internet revolution. Information content providers control this resource. These companies run the gamut from creators of information to suppliers or re-directors of information feeds. These companies can be very large organizations with staffs devoted to producing news, weather, sports, and other information. They can also be very small organizations or individuals who provide specialized information.

Some information like news and weather may be intended for public distribution and consumption. Other information may simply exist in company databases that support the mission of the business, but could be very useful if accessed wirelessly.

It is likely that some "killer apps" will need to use information controlled by information content providers. These applications may perhaps package and present the information in a new and improved manner.

2.6 Subscribers

Subscribers drive the market. They are the reason the other players exist. These consumers run the gamut from highly technical to very non-technical, from very demanding to easily satisfied, from heavy users to occasional users.

Successful products are those that are well designed for the intended user and that meet the needs of the majority of users. This means that easy to use is generally preferred over technically challenging, speedy execution and response is preferred over slow and sluggish response, simple is preferred over complex. The successful developer architects the "killer app" based upon a thorough understanding of the users' needs and behaviors. The application profile – in terms of the amount of data transferred over a period of time, latency requirements, etc. – should be determined and anticipated by developing a good understanding of the subscriber's needs and behaviors.

3.0 Inside The Paging Infrastructure

This section breaks down the paging infrastructure into components, explaining the functions of each. This knowledge is useful to a developer for several reasons. First, a basic understanding of paging infrastructure helps build the vocabulary so that developers can understand and communicate with carriers. Second, by understanding the components of an infrastructure, it's easier to see where bottlenecks can occur and understand the need for certain design recommendations. Finally, understanding the infrastructure reinforces the fact that there is more to paging technology than the “wireless information pipe” metaphor would suggest.

3.1 System Overview

A paging network is a collection of paging terminals, system controllers, transmitters, receivers and data links, all carefully engineered to make sure the paging system has optimal coverage and response capabilities with minimal interference. Sophisticated coverage mapping tools are often used to help determine infrastructure site spacing which is needed to maximize coverage and minimize interference.

Paging systems in some ways are similar to cellular networks, yet in some ways they are very different. Like cellular systems, virtually all paging networks, other than small building-wide systems, involve more than one transmitter. Paging networks, especially those that support one-way subscriber devices, rely on a simulcast capability to blanket an area where a subscriber is likely to be.

Since one-way subscriber devices have no way of telling the system where they are, several transmitters must send the same message over a wide area using the same frequency. This “fire and forget” operation is much different than the cell-based targeted delivery methods used in cellular systems. Radio signals can and do cause interference when different messages are transmitted using the same frequency at the same time. Therefore, it is important with paging systems that transmission timing be precisely controlled.

Theoretically, two-way messaging networks could target delivery of messages much like cellular networks. However in reality, paging carriers may use the same infrastructure to support both one-way and two-way subscribers. Furthermore, not all paging infrastructure supports single transmitter or cell-based targeted message delivery. Even if targeted messaging is available for two-way subscribers, there is still the issue of sending broadcast messages to many two-way subscribers.

The following figures show a few of the many possibilities for paging system zone coverage and frequency use. Notes addressing how interference can be avoided are also provided.

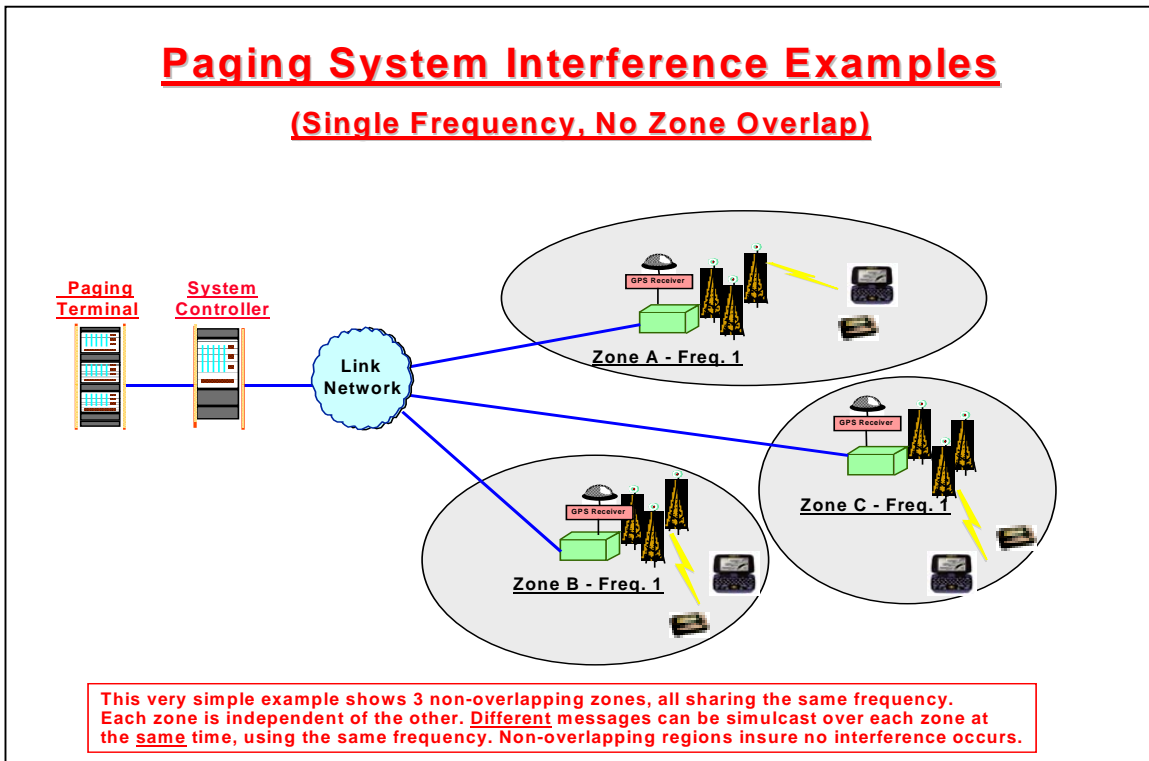


Figure 1. Non-overlapping Paging Systems, One Frequency

Figure 1 shows 3 non-overlapping zones in a system that has a single frequency. The zones might represent widely separated markets such as Miami, Chicago, and Dallas / Ft. Worth. Many transmitters cover each market. The geographic dispersion and relatively low power of the paging transmitters ensures that the different messages can be simulcast within each market at the same time using the single frequency without interference.

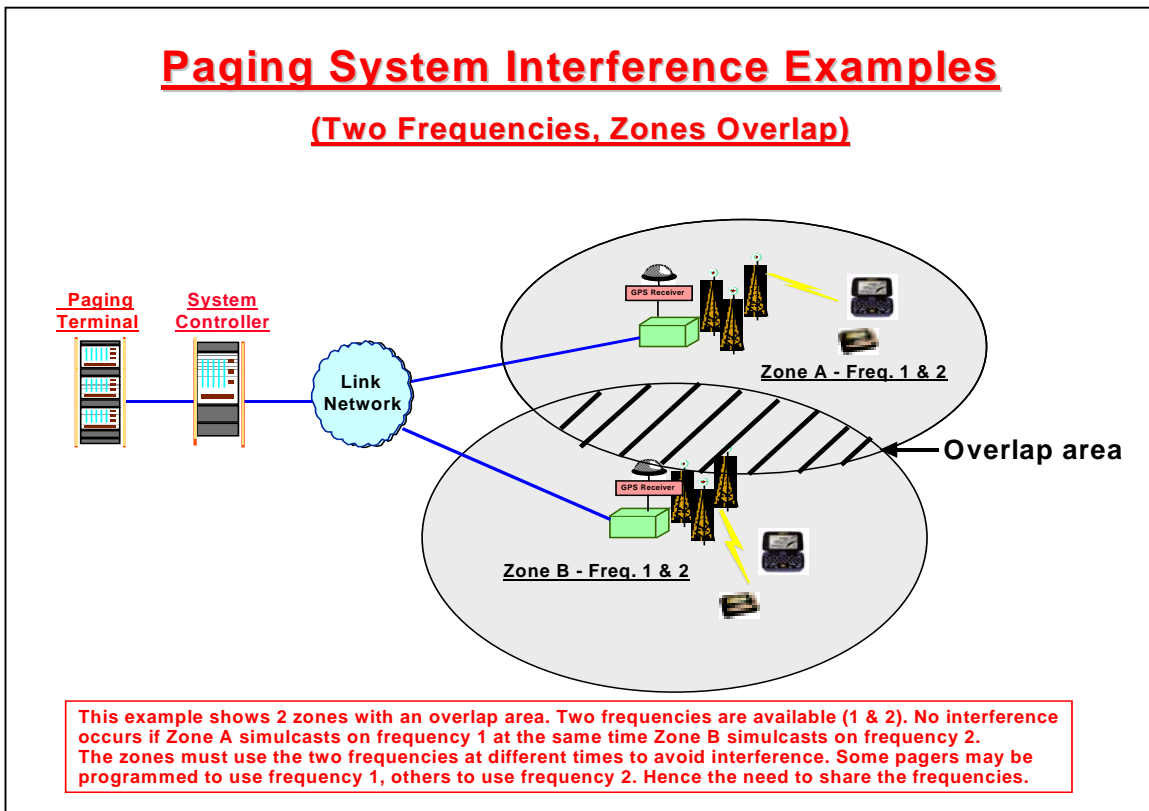


Figure 2. Overlapping Paging Systems, Two Frequencies

Figure 2 shows the case where two zones have an overlap region, but two separate frequencies are available to the paging operator. The example might be the Dallas and Ft. Worth, TX markets. Dallas could be Zone A, Ft. Worth could be Zone B. Because of the close proximity of the two cities, we have an overlap area. Because the operator has two separate frequencies, it's possible to alternate the use of the frequencies. Frequency 1 can be used in Zone A while Frequency 2 is used in Zone B. Then the zones can switch. Frequency 2 can be used in Zone A while Frequency 1 is used in Zone B. Both frequencies are needed in both zones because some pagers may be programmed for one specific frequency. Care must be taken to ensure frequency use is coordinated to prevent both zones from using the same frequency at the same time to send different messages.

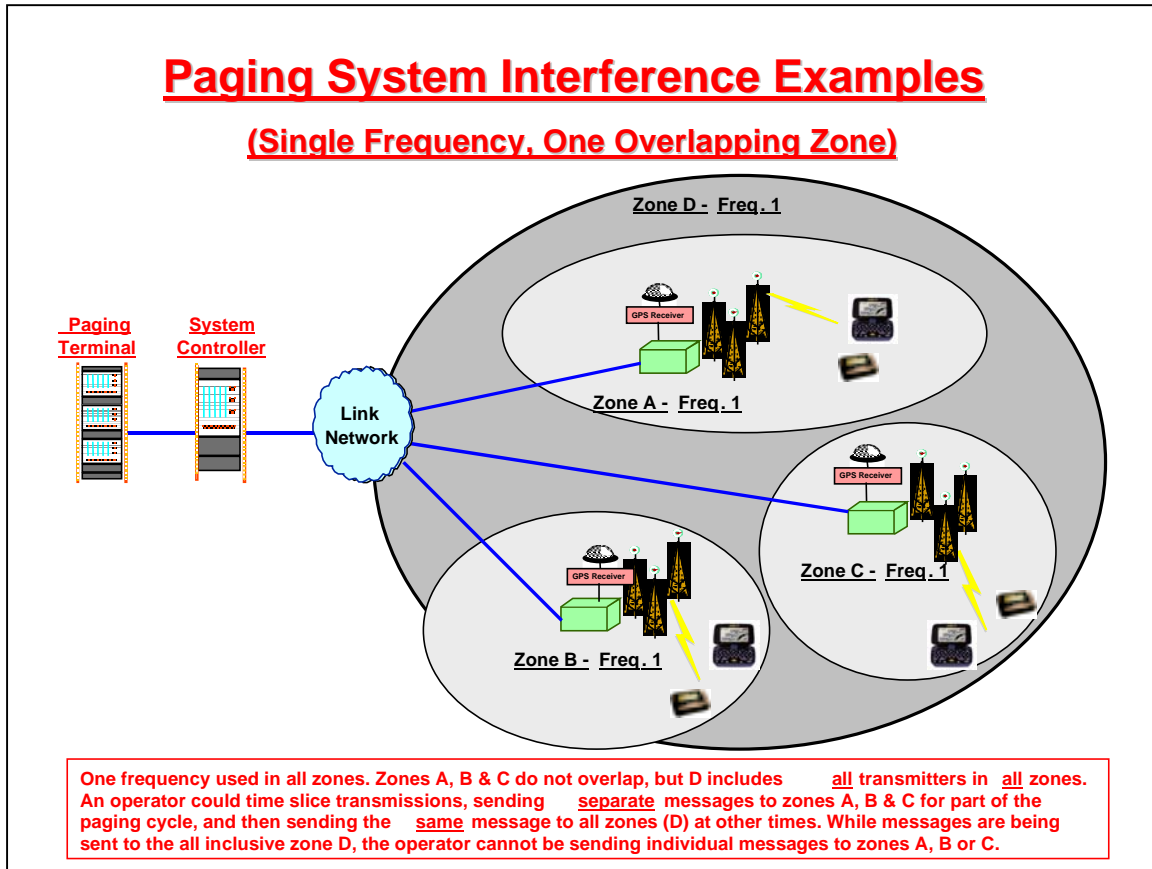


Figure 3. Overlapping Paging Systems, One Frequency

Figure 3 shows a system that is set up as 3 non-overlapping zones (A, B & C), and one super zone (D), which is actually the combination of all 3 zones. This could be the case in a region including several major cities such as New York, Philadelphia, and Washington. In this example, some subscribers pay for local service in their home city, others pay for a wide area service that covers the whole region. In this example the operator has a single frequency which must be shared. The operator can statically configure his scheduler to send separate messages to each of the three non-overlapping zones during part of the paging cycle and then send messages to the wide area subscribers during the remainder of the paging cycle. During the time messages are sent to the whole region, all transmitters are simulcasting the same information. This solution permits the carrier to send a single message to all three zones to be transmitted at the same time to reach subscribers who may be in any of the three zones.

Note that this example is just one way a carrier could provide wide area coverage using a single frequency. The other obvious alternative is to simply treat the regions as 3 non-overlapping zones, just like the example shown in Figure 1. In this case 3 separate copies of a message intended for a roaming subscriber could be sent, one to each zone. There is no requirement that the messages be transmitted at the same time since the regions do not overlap. In fact, depending upon the proportion of local to roaming subscribers or the balance of message traffic to each zone, this alternative may be preferable to the super zone solution. This is because the super zone solution usually requires that only roaming messages be simulcast during the time periods reserved for wide area (Zone D) coverage. If there are not enough messages to completely fill the (statically configured) reserved time slot for the super zone, the unused portion of the time period cannot be used to send local messages. Of course, if the system controller (the scheduler) is capable of dynamically adjust its scheduling to switch from wide area to local coverage when roaming message queues are empty, then this problem is alleviated. However, many paging systems do not implement these sophisticated scheduling algorithms. In the event that multiple system controllers are used to handle the

different zones, the likelihood that a super zone could be used is further diminished, since few system controllers contain the external synchronization timing and logic needed to implement this.

Paging systems are store and forward systems. They accept messages for delivery to paging devices, and store them for a brief period of time before delivery. This storage is necessary because pages must be scheduled for delivery. The storage interval is typically just a few seconds, but depends on a number of factors including the over the air protocols used, the system utilization at the time the message arrives, pager and system configuration settings. In the worse case, it could take several minutes for a message to be transmitted.

Paging systems are highly reliable, but are not bullet proof. It is possible for messages to be lost while in the paging infrastructure. This is rare but can occur due to system component failures or in some systems due to excessive network traffic. Excessive message traffic can have a strong impact on message latency. Users or applications that send excessive data through a paging infrastructure can cause significant delays for all messages that use the infrastructure.

A sample paging system is illustrated in the diagram below.

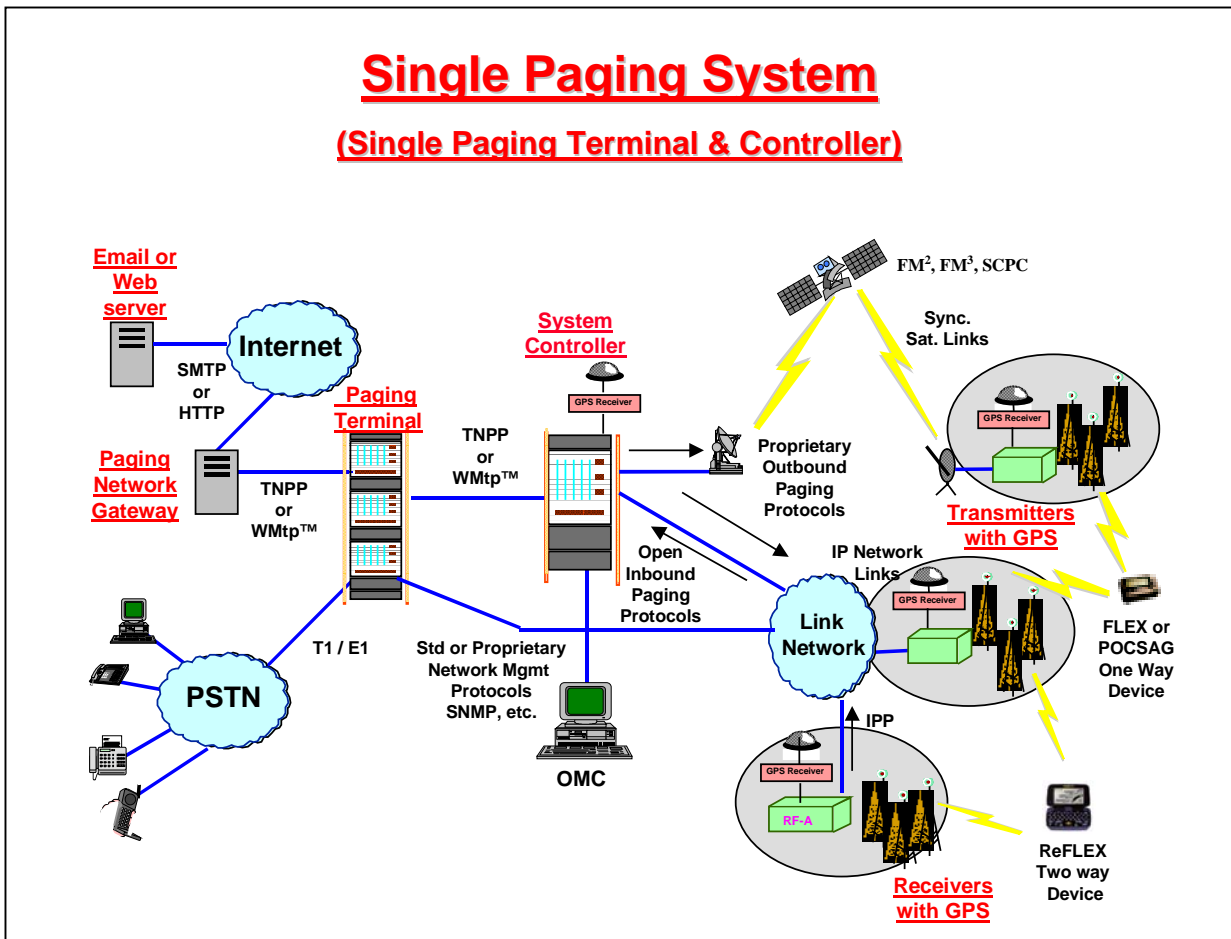


Figure 4. Single Paging System

Figure 4 provides an overview what a simple paging system contains. However, a few notes are warranted. First, not all paging systems have all the components shown. For example, one-way paging systems may not have the Internet access components and certainly will not have the receivers. Some systems have an external network management system, others handle the operations, maintenance and control (OMC) functions directly within the paging terminal or controller. Some suppliers combine the functions of the paging terminal and system controller into a single “box” or sub-system. While the figure shows both

satellite and link network connections to the transmitter site (typically IP), actual systems usually will have one type of distribution network, not both.

Figure 4 shows a collection of transmitters and receivers in small, localized areas. The illustration is misleading. In real systems transmitters and receivers are interspersed across the geographical area as needed to provide adequate inbound and outbound coverage.

Additional components needed to support the carrier's business have been omitted from Figure 4. For example, all paging systems have some provision or link to a billing system. Most systems need ancillary equipment such as operator terminals and printers, which are necessary for subscriber provisioning and system configuration.

Figure 4 illustrates a single paging system. This configuration might exist for very small paging operators who serve a small geographic area. However, larger systems are usually distributed and include many paging terminals and controllers. Each distributed system serves a portion of a carrier's total market. Figure 5 below shows an example of two interconnected paging systems.

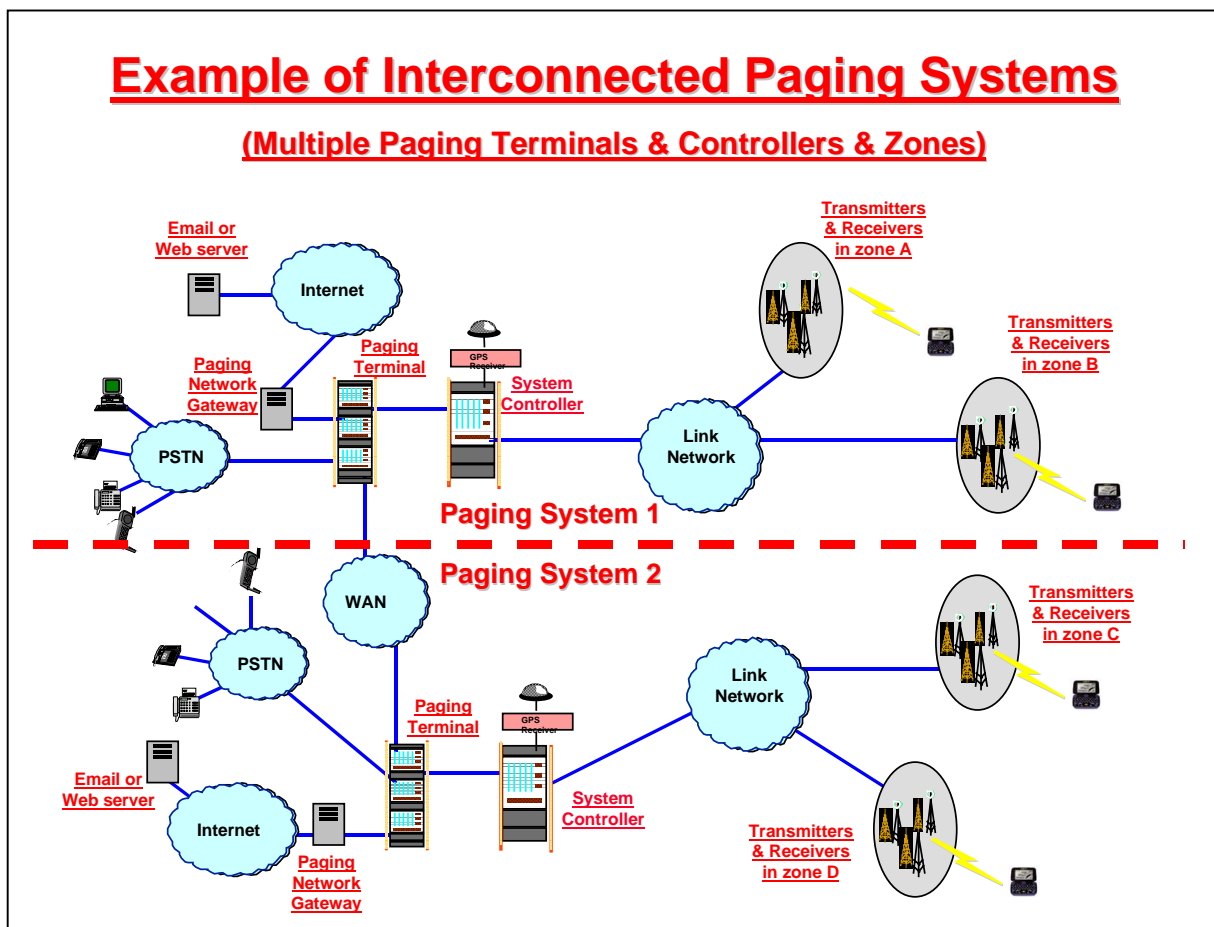


Figure 5. Interconnected Paging Systems.

A few explanatory notes are warranted for Figure 5. First, interconnection is not limited to two systems. Some large carriers provide nationwide coverage that requires many terminals and transmitters. Also, other interconnect possibilities exist. For example, some infrastructure suppliers permit many paging terminals to send traffic to a single system controller which then sends paging traffic to a wider collection of transmitters.

This section has served as a high level introduction and view of a paging system. Now it's time to explain the functions of the various components.

3.2 Paging Terminals

The paging terminal is usually the entry point to the paging system. It automates the front end paging function by connecting callers to the paging system, accepting and validating messaging requests, performing other paging administration services, and forwarding messages on to other sub-systems in the paging system. It often offers features in addition to basic paging such as integrated voice mail services. It provides interfaces to billing systems, operator assisted paging systems and various Internet gateways and servers.

Messages may originate from many sources. In the past, the most common origination point was the public switched telephone network (PSTN). With the introduction of alphanumeric and two-way messaging, sources for paging have expanded and now include specialized paging input devices, personal computers, sites on the Internet, and other pagers or wireless devices. Regardless of source, messages usually must pass through the paging terminal.

Paging terminals interface to the Public Switched Telephone Network (PSTN) to permit subscribers to dial up and send pages from their telephones using dual tone multi-frequency (DTMF). Messages entered from the PSTN are usually numeric, but various techniques have been used in some paging terminals to permit callers to send limited alpha messages from a telephone. However, sending alpha messages from a telephone keypad is tedious, generally requiring the use of various key combinations to select alpha characters.

To support alphanumeric messaging, many terminals support access to alpha entry devices, which may include PCs or other specialized terminals that can compose alpha messages. These messages may be sent directly to the paging terminal through a dial up modem connection, or a direct connection via a dedicated circuit. Some vendors support access to the paging terminal through a separate gateway.

With the advent of the Internet and the explosion of e-commerce, paging terminals need to accept messages from new sources such as email and the World Wide Web. Most paging terminals do not directly support the mail and Internet protocols such as SMTP and HTTP. The common solution to this problem is for infrastructure suppliers and other vendors to develop gateways that translate between the Internet protocols and the paging protocols supported by the paging terminals. This was illustrated in Figure 4.

Paging terminals contain information about paging subscribers. Generally the subscriber database is distributed across many paging terminals in a network. Each subscriber has a single "home" terminal which is where his service information or "profile" is stored. Subscriber profiles include their Personal Identification Numbers (PINs), information on the type of device they have, information regarding the services they are permitted to use, service limitations, and subscriber configuration parameters. Typical service options may include whether voice mail, message storage, etc. are used. Limitations may include the maximum number of messages permitted in some time frame, maximum message lengths, etc. Subscriber configuration options may include user-selected passwords, custom greetings, whether they can divert or forward messages, etc. ??? too many etc's - also next sentence

Paging terminals process requests for paging services regardless of the point of origin, PSTN, wide area network (WAN), etc. The process generally includes a voice response script or other menu-driven option of services from which a caller can make a selection. The subscriber to whom the caller wishes to send a message is provided?????, either automatically by the telephone network based on the telephone number dialed, or as entered by the caller. This information is used to validate the subscriber. If the subscriber information resides in the paging terminal that answers the call, the information is validated and used to determine what services and features the caller can request for the subscriber. If the subscriber's information is not located in the paging terminal that answers the call, but is located in another home terminal, then the paging terminal may be designed to forward the request to the home terminal for processing. If the subscriber information is not found anywhere in the network, the call is rejected.

One vital function normally handled by the home paging terminal is translating a subscriber ID into the capcode of the device used by the subscriber. Paging devices accept and decode messages addressed to their unique capcode, the address of the pager. Paging devices don't understand subscriber IDs, which are

assigned by service providers and may be either unique telephone numbers or PINs?????. Without this translation function, paging is not possible.

The paging terminal gathers all the necessary information from the caller including the message to be sent. It then sends this information along for further processing through the paging system. In some cases this means the information is sent to a separate system controller that schedules, batches, and encodes the information for delivery to transmitters. Some paging terminals include the queuing, batching, scheduling, and encoding functions in the terminal itself. Other architectures separate some or all of these functions from the terminal and require a separate system controller to perform these functions.

The paging terminal generally holds the message until the downstream system controller accepts it. The terminal may even store the message for later retrieval and re-transmission.

Once the message has been sent, paging terminals usually have an accounting function built in which records the call or message details for later billing. The information can be transferred to a separate billing system for processing. The billing interface is not shown in Figures 4 or 5.

One-way subscriber devices cannot tell the paging network their location because they don't have transmitters. For these subscriber devices, the paging carrier typically configures the subscriber record to include all the zones or areas where the subscriber may be located. The paging system instructs the system controller to simulcast messages to all these zones.

Some one-way systems, particularly in Asia, support a roaming feature whereby a subscriber can call into the paging terminal to tell it where he / she will be during a defined period. This permits the paging system to "target" message delivery to a smaller area.

Two-way messaging devices have an onboard transmitter. These devices can tell the paging system where they are so messages can be targeted to a particular area. (The paging device actually tells the paging system which transmitter's signal is the strongest and therefore which signal should be used to send messages to the device. This is possible because each transmitter sends a unique "color code" with the message which is detected by the paging device.) This capability theoretically can greatly increase network capacity since the system does not need to "light up" all transmitters when sending a message to a subscriber. However, actual implementations of "targeted delivery" in paging systems produces lower real capacity savings than in theoretical scenarios. This is because in some circumstances the system may still need to ask subscriber devices where they are before they send the message. These small "where are you" messages are generally simulcast over a wide area, thereby reducing some of the RF channel savings from targeted delivery.

Paging terminals that support two-way devices must keep track of subscriber locations so they can efficiently target message delivery. Two-way protocols generally use a request / response messaging paradigm and therefore the terminal must be able to match outbound commands and messages to inbound replies.

Because two-way subscriber devices can confirm receipt of messages, the paging infrastructure can better ensure message delivery. The paging terminal will hold on to two-way messages for some system configurable time until confirmation is received. If confirmation is not received within the configured time, which is typically several days, the message will be discarded. This is in contrast to the "fire and forget" approach taken for one-way messages.

Tip: Although some two-way paging service providers boast of "guaranteed delivery" for two-way messages, the truth is nothing is really "guaranteed". A pager that is never turned on will never get a message. Subscribers that are never in two-way coverage will never be able to send confirmation to the paging system. Developers should understand there are situations when messages may not be received even in systems which "guarantee delivery", and should deal with this possibility.

Paging terminals are generally available in either a redundant or non-redundant configuration. The paging terminal software contains the logic necessary to manage redundancy, to detect failures, and to switchover to the backup system. This logic is responsible for making sure the paging terminal continues to function when system hardware or software fails.

3.3 System Control Switches

System controllers perform the tasks of queuing, batching, encoding, and scheduling messages received from paging terminals for delivery to transmitter sites. They also handle processing of two-way inbound messages that originate from receiver sites. The system controller's task is to manage the distribution of messages to the many transmitters in such a manner as to optimize the use of the distribution links and over-the-air (OTA) radio frequency (RF) spectrum. Some infrastructure suppliers combine the system control function in the paging terminal equipment. Others keep this function separate as shown in Figure 4.

The system controller accepts inputs from one or more paging terminals. The messages are queued until they can be scheduled for delivery over the distribution network to the transmitters. The length of time the messages may remain in queue is a function of the supported over-the-air protocol and the message traffic destined for a particular area. In the case of one-way messages, the controller may have to send the messages to several dispersed geographic areas to reach the intended subscribers. For two-way messages, inbound messages help locate the subscriber, so the target area can be much smaller. Not all system controllers support both one and two-way messaging.

As discussed in the paging system overview, paging systems typically simulcast a message over a wide area to reach the intended subscriber. This generally involves more than a single transmitter. To avoid interference in a geographical area covered by more than one transmitter, the messages must be precisely timed so that they are simulcast from all the transmitters at exactly the same time. Newer protocols that support higher data rates typically have tighter timing constraints.

The system controller must compute the necessary transmitter "launch times", allowing for the time it takes to send the messages across the distribution network to the transmitter. This launch time is sent to each transmitter controller along with the message payload. The message must arrive at the transmitter controller just before the designated launch time so the transmitter controller has time to process the message and send it to the transmitter's power amplifier at precisely the right time. If the message arrives too early, it may cause the transmitter controller's input queue to overflow. If it arrives too late, the message will not be transmitted.

Messages that arrive from the paging terminal are sent using one of several paging terminal protocols. When they are scheduled and transmitted over the distribution network, they are encoded using a different protocol. The system controller performs this encoding. The encoding into a different protocol is necessary for many reasons, not the least of which is to provide some error detection and correction to protect message payloads when they are transmitted over the air.

In two-way paging systems, such as those used for the advanced messaging devices, the system controller plays a key role in locating subscribers and processing inbound messages received from receivers. The two-way protocols must handle user initiated inbound peer-to-peer messages. While it is possible to handle all inbound messages as randomly generated messages, similar to the way Ethernet LANs work, it is much more efficient to schedule inbound messages in order to minimize collisions and optimize the use of the inbound RF channels. The system controller handles this inbound message scheduling function. The two-way protocols define the messages necessary to request inbound transmissions and schedule the transmissions.???? The system controller must implement these complex message flows.

The most valuable resource available to paging carriers is their RF spectrum. It can be difficult and expensive to acquire additional spectrum. Therefore, the system controller plays a significant role in optimizing the use of this scarce resource by implementing sophisticated scheduling algorithms, both for the inbound and outbound RF channels.

The system controller may also encode, transmit, and receive system management and control messages over the same distribution links used for basic messaging. These messages include sending configuration information, requesting and receiving diagnostic information, and downloading new software to the transmitters and receivers. Different infrastructure systems handle the operations and maintenance functions differently. Some embed this functionality within the controller directly. Others leave it to a specialized network management system. Depending upon the network management protocols and the type

of distribution network used, it may be necessary for the system controller to handle the scheduling of these messages along with basic paging messages.

Data networks connect paging terminals to system controllers and system controllers to transmitters and receivers. The protocols used between the paging terminal and system controller and between the system controller and the transmitters / receivers generally determine the type of networks used. Some protocols require serial synchronous circuit switched or direct connect networks. Others require packet switched networks. The system controller must provide interfaces to the appropriate networks. Not all system controllers support all types of networks. Newer controllers tend to support IP based networks, which are gaining favor since they can be used to support text and voice messaging, as well as network administration.

System controllers generally come in either a redundant or non-redundant configuration. Redundant configurations better ensure that the paging system remains operational when system hardware or sub-systems fail. The controller software manages the failure detection and switchover functions.

3.4 Transmitters

Transmitters handle the over-the-air transfer of messages from the paging infrastructure to the subscriber's messaging devices. Paging transmitters range in power from less than 100 to around 300 watts. They must be located so as to provide adequate coverage over the intended service area. Typical commercial paging systems include hundreds of transmitters.

Transmitters are designed to operate in specific frequency ranges. Different frequencies are required in different regions of the world, since the governing bodies in each country set aside different frequencies for use by services such as paging, cellular telephone, radio, etc. This means that a paging transmitter designed for use in the United States may not be usable in Europe, since the allocated paging frequencies are different.

Paging systems operate at relatively low data rates, typically ranging from 1200 to 6400 bps on each outbound RF channel. Effective data rates are even lower since the over-the-air protocols include overhead needed for batching, error detection and correction.

Moving to higher data rates requires the paging carrier to install more transmitters with closer site spacing to achieve the same level of coverage as with lower data rates. This is a costly tradeoff, since it requires an investment in more paging infrastructure and more site rentals. This means that some markets may have lower data rates than others, depending upon the economics and customer demand.

Paging protocols have traditionally delivered messages over the air using 25 kHz outbound RF channels. The two-way protocols define 50 kHz paging channels. The ReFLEX™ 25 protocol specifies either 1 outbound channel in a 25 kHz band or 3 outbound channels in a 50 kHz band. ReFLEX 50 supports up to four outbound subchannels of 6400 bps in a 50 kHz band. InFLEXion™ specifies up to 3 carrier channels and 7 subchannels in a 50 kHz band. The transmitters are designed to support one or more paging protocols. Few, if any, transmitters support them all.

Modern paging transmitters may either be linear or Frequency Modulated (FM). FM transmitters transmit on a single frequency at a time. Linear transmitters may transmit on multiple frequencies at the same time. Linear transmitters may be needed to support the advanced voice paging protocols such as InFLEXion. FM transmitters tend to be much less expensive than linear transmitters.

Transmitters accept messages from system controllers and transmit them at the assigned time. The transmitters must support the distribution protocol used between the paging system controller and the transmitters. They must key up at the appropriate time, encode the messages according to the over-the-air protocol, and transmit the data at the precise time indicated by the controller.

Transmitters also must support operations and maintenance functions including the ability to accept configuration changes and new software downloads. In some cases, the configuration data is sent over the same distribution links as the paging data. Sometimes the configuration changes can be made over dialup links or directly at the transmitter site using a control panel on the transmitter.

Transmitters are actually made up of several modules. A transmitter controller connects to the distribution link to receive messages from the system controller. This component contains the processing logic needed to handle paging and control packets. It is the brain that controls the operation of the transmitter. The transmitter also includes one or more power amplifiers to generate the signal that is fed to the transmitter antenna. Transmitters generally have manual control interfaces that permit technicians to enter commands when they are performing on-site service. Some advance paging protocols have very tight timing requirements that can only be met by using a GPS receiver. Data communication equipment, such as routers or satellite receivers, are also needed. Some transmitter vendors include these communications devices within the transmitter controller others provide separate components. All of these components will be located at the transmitter site.

3.5 Receivers

Receivers are needed in two-way messaging networks. Like transmitters, they operate in specific frequency bands. They generally have very high sensitivity since they must detect messages from paging devices that may be operating at power levels well below one watt.

Receivers generally operate at inbound data speeds ranging from 1600 to 9600 bps. The effective inbound data rates may be much lower than this due to overhead needed for packetizing, error detection, etc. As was the case for transmitters, moving to higher inbound data rates requires an investment in a larger number of receivers, spaced closer together.

Receivers must support the inbound paging protocols used between the receiver and the system controller. These tend to be IP based protocols.

Receivers must also support operations and maintenance functions including accepting new configuration settings and new software downloads. These functions may be supported over dialup links, or packet data networks using IP based protocols.

Like transmitters, receivers are made up of many components. These include a controller that contains the necessary processing logic, circuitry that detects and processes received signals, data communications equipment, GPS receiver, and antennas.

3.6 Gateways and Servers

With the advent of two-way messaging comes the need for portals to and from the Internet, mail systems, corporate databases, and other repositories of electronic data. These portals are gateways or servers that manage protocol translation, handle the routing of information between external systems and the paging infrastructure, run database engines, etc. These systems may be owned and operated by the paging carrier or may be external to the carrier's operation????, perhaps being owned by an enterprise.

Servers and gateways tend to have very specialized functions. A particular application may depend upon several servers, each providing a particular function. For example, a paging carrier may have a server that accepts messages from an external mail server using Simple Mail Transfer Protocol (SMTP), and translates the email request into a form and protocol suited to the paging infrastructure. Examples of such a configuration are shown in Figure 6.

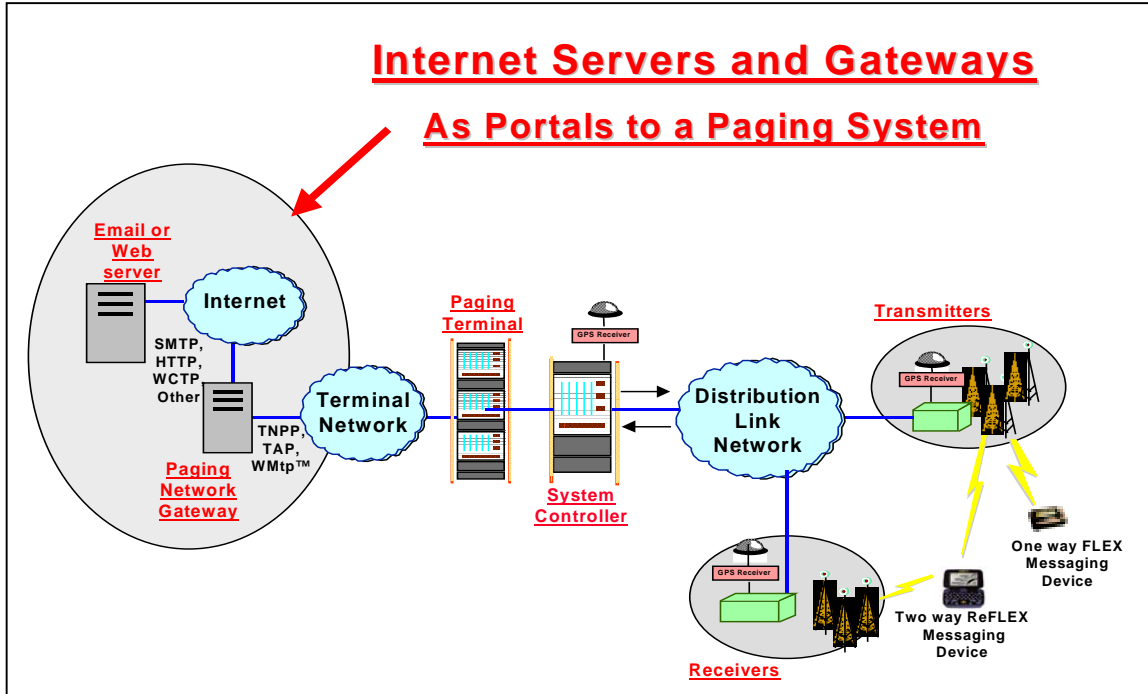


Figure 6. Servers and Gateways in a Paging System

Examples of servers and gateways used with paging systems are Motorola’s FLEX™ Messaging Server (FMS) and Glenayre’s GL3200 Data Applications Platform.

Tip: No single standard exists for routing messages originating from a two-way messaging device to a particular service or application in a server or gateway. These tend to be carrier specific. Some carriers use “virtual PINs”, others use IP sockets, etc. Carriers should be consulted for information concerning the method(s) they support. Some carriers provide their own Software Development Kits (SDKs) which provide Application Program Interfaces (APIs) for accessing their services.

3.7 Terminal and Distribution Networks

The networks used to connect paging terminals to each other, paging terminals to system controllers, and system controllers to transmitters and receivers can be very complex. The networks may be terrestrial wireline or wireless. Wireline networks may include dedicated digital links such as T1 / E1, analog lines, or X.25 and Frame Relay networks. Wireless networks may include satellite, microwave, or RF.

Networks may exist between any components in the paging system. Except for small test or on-premises systems, a distribution network always connects system controllers to transmitters and receivers. Terminal networks may exist to interconnect paging terminals in large, geographically dispersed paging systems. Paging terminals and system controllers are often co-located, but a link network may exist to connect them when co-location is not possible, such as when a system controller handles message traffic from many geographically dispersed markets, each served by one or more paging terminals.

The type of network used dictates the network equipment needed. The network may include modems, Channel Service Units / Digital Service Units (CSUs/DSUs), multiplexors, satellite receivers, routers, hubs, etc. The type of network needed is often dictated by the specific paging system components used. For example, a particular system controller may require synchronous digital circuit switched distribution networks. Another may support packet data networks running IP based protocols.

Network links often must carry more than one type of information and must support more than a single protocol. For example, many paging systems send paging traffic to transmitters using proprietary protocols based upon User Datagram Protocol / Internet Protocol (UDP/IP), and use the same networks to manage the transmitters using Simple Network Management Protocol (SNMP), File Transfer Protocol (ftp), telnet, etc.

Some networks cannot support certain protocols. For example, one proprietary protocol developed by Motorola, which is used to send paging and control information from some system controllers to some transmitters, requires that the link looks like a wire, that no bit stuffing is done, and that certain clocking requirements are met. This eliminates from consideration most packet switched networks in the distribution network.

4.0 Subscriber Devices

Wireless subscriber devices can include pagers, cellular or PCS phones, Personal Information Managers (PIMs), specialized terminals, etc. They typically operate in a limited frequency band and support a single or limited number of over-the-air protocols. This means that a given device may have a very regional focus. Device capabilities may be limited by the protocols and networks they support. ????

Most paging devices are designed to operate on a single frequency; they are not able to scan frequencies. The frequency is programmed into the pager in the factory and corresponds to a frequency the carrier is licensed to use. If a carrier acquires new frequencies, either new pagers must be ordered or old pagers must be re-programmed. This has implications as to how a carrier adds capacity to an existing paging system.

Paging subscriber devices run the gamut from simple one-way paging to complicated two-way messaging devices. In fact, they may even fall somewhere in between one and two-way. So called one and a half way pagers are devices that receive messages and automatically acknowledge reception back to the paging infrastructure. These devices have both transmitters and receivers, but are not capable of sending user data inbound??back to the network???. Some pagers permit the user to send canned but not free form messages. Some paging devices even support voice messaging.

Subscriber devices are designed to operate in rather hostile environments. They must be able to detect valid data and reject erroneous data. They generally feature excellent sensitivity needed for good in-building reception. They operate on battery power and must be designed with energy conservation in mind. For convenience they generally are designed to be small and light weight.

Paging subscriber devices have numerous characteristics that differentiate one from the other. Some are numeric-only, one-way devices. Others support alphanumeric text. Some support binary data. Some have just a few simple buttons, while others have complete QWERTY keyboards. Some are single line text displays. Others feature “full screen” graphics capable displays. Color displays may appear in devices in the not too distant future.

Some paging subscriber devices are fully programmable and have capabilities similar to miniature computers. In general, these devices are all two-way. These are the devices that are the primary focus of this paper.

Pagers and wireless messaging devices have a unique address associated with them that is used to select the specific device that is to receive a personal message. This address may be called the capcode or RIC (radio identification code). In addition to the individual address, many messaging devices support one or more broadcast addresses. These are shared addresses that permit many wireless devices to receive the same message. This is useful when an information service data feed, such as news, weather or sports, is to be received by many devices.

Many wireless messaging devices can be programmed over the air to add or delete broadcast addresses, change home address assignments, enable or disable features, etc. No standard exists among pagers for changing this information in the “code plug”. Therefore, carriers must send different data strings to different devices to change their configurations. The ability to change the “code plug” is dependent both on the capabilities of the pager and the paging infrastructure.

Tip: Motorola supports a Generic Over-The-Air Programming capability through its GOTAP server. This capability hides the non-standard nature of programming some pagers’ code plugs. GOTAP defines a set of generic commands to modify specific attributes or configurations that are code plug independent. The GOTAP server translates the generic commands into the device dependent strings that actually change the code plug.

It is important to note that a single paging network may host a variety of pagers, from the simple to the complex. The configuration of the paging network is influenced to some degree by the types of subscriber devices that must be supported. For example, some older paging protocols such as POCSAG are not constrained to send messages on frame boundaries. Newer higher speed protocols do have frame boundary

and timing constraints. When the paging network must support both, the paging channel must be divided between the two. Some paging infrastructure can be configured to favor one protocol and hence one class of pager over the other. The configuration choices made by the paging carrier can affect message latency and other behavior characteristics that could be important to an application developer.

5.0 Protocols

This section covers many of the protocols used in paging systems. It is not necessary to understand the details of the protocols to successfully develop applications for paging systems. The intent of this section is to give the reader an overview of the protocols and a general sense of the complexity involved in sending information across paging networks.

5.1 Overview of Protocols

Protocols are very simply the “rules of the road” that must be followed when exchanging information. They define the order of, content of, and constraints on information to be sent, plus they identify the optional and mandatory information that must be exchanged between two entities. They also define features that may be supported.

Protocols set the bounds of what is possible when information is exchanged. They greatly influence or determine attributes such as latency, capacity, efficiency, robustness, immunity to errors, and the type and amount of data that can be exchanged.

While it is not necessary for a wireless application developer to understand the details of all the protocols used within the end to end system, it is important to be aware of the role protocols play and to understand that protocols may limit what a developer can do.

Protocols tend to evolve over time. Individuals working on technical committees write protocols and update them as new capabilities are defined, or as clarifications are needed. Different paging carriers may support different paging protocols or different versions of the same protocols. This means that their networks have different capabilities and may support different features.

Protocols define the limits of what is allowed. Paging infrastructure suppliers are free to implement system components as long as they do not exceed bounds set by the protocols. However, it is very often the case that paging infrastructure suppliers do not implement or support all the features and capabilities allowed by the protocol.

Tip: It is not safe to assume that just because a paging system “supports” a particular version of a protocol that it supports all the features allowed by that version of protocol. For example, some paging protocols such as TNPP define codes used to identify various types of pagers. The paging infrastructure may support only a subset of the pagers defined in the protocol specification.

The following sections discuss a large number of protocols used at different points in a paging system. It is important to understand that every interface point where one protocol is handed off to another is a point of potential congestion and message latency. At each of these points, the system usually must store the messages in a queue prior to processing them.

Before beginning our discussion of protocols we should define a few terms and roles. First, the individual or system that accesses the paging system is referred to in the following text as the caller. The person or wireless device the caller wants to reach is called the subscriber. Callers may or may not be subscribers to a paging service.

The Figure 4 identifies various protocols used in paging systems. This section will not discuss all of these protocols, but will focus on those commonly encountered in a paging system. The protocols shown in the figure are not all encompassing. Many others have been defined over the years and are supported in various paging systems.????

5.2 Protocols Between Outside World and Paging Infrastructure

5.2.1 From PSTN to Paging Infrastructure

Several paging protocols have been defined over the years to deliver messages or information from devices and external systems to a paging system. The primary ones include Telocator Network Paging Protocol (TNPP) and Telocator Alphanumeric Protocol (TAP).

In addition to these machine to machine protocols, messages can be entered into a paging system from a telephone by a caller who responds to an automated voice response script. The paging terminal answers an incoming call, and through a series of voice prompts, obtains the information needed to send a message to a subscriber. The caller enters the requested information from the telephone keypad using Dual Tone Multi-Frequency (DTMF) tones. The methods and protocols used to manually enter page requests from a telephone will not be discussed further in this section.

5.2.1.3 TNPP (Telocator Network Paging Protocol)

TNPP is an ASCII character oriented protocol originally intended for transmission over RS-232 asynchronous data links utilizing speeds ranging from 300 to 9600 bps. Infrastructure suppliers have actually implemented the protocol over many types of data links besides RS-232 serial links, at speeds in excess of the maximum 9600 bps data rate allowed by the specification.

TNPP includes provisions for delivering information between two paging nodes and for forwarding information to other nodes. That is, TNPP supports intermediate nodes that act as routers or forwarding agents. Both possibilities are illustrated in Figure 7.

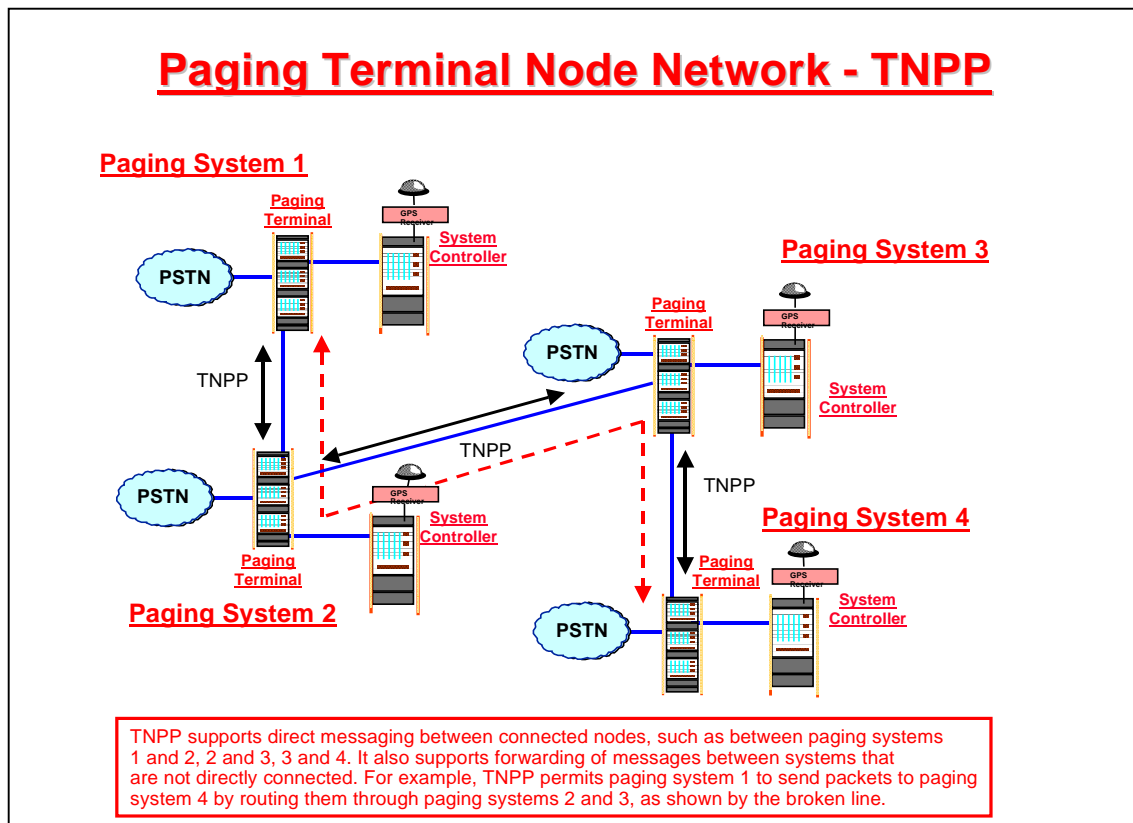


Figure 7. Message Routing in Paging System Using TNPP

TNPP sends information in different types of data blocks. Two types of data blocks are used for sending paging data. One is called a CAP page block which is used if the block contains all the signaling information needed to generate the page. The target pager's capcode is included in the message. This type of data block would normally be used between a paging terminal and a system controller. The other type of data block used for paging data is the ID page block. This type contains a subscriber ID rather than a capcode. This type of data block is normally used between paging terminals.

Tip: Remember, before a message can be sent to the pager, the subscriber ID must be converted to a capcode. This is normally done by a paging terminal.

TNPP defines three additional types of blocks used for purposes other than paging. They are COMMAND, DATA, and STATUS. The COMMAND block is used to send configuration and control commands to a destination. The DATA block can be used to send non-specific data to a destination. The STATUS block is used to report errors in equipment that may be attached to a node.

The protocol includes provisions for handling error conditions between nodes and between end points. CRC codes are generated on the data contained in TNPP packets. Nodes check the validity of CRCs, and request retransmission if errors are detected.

The maximum size of a TNPP packet is normally 1024 bytes. This length includes everything in the packet, including header, user data, start and end of text codes, CRC and other control characters. This means that the maximum amount of user data that can be sent in a packet is less than 1024. A reasonable average maximum packet size is around 1000 bytes. Many data blocks can be combined in a single packet as long as the maximum length of the packet is less than 1024 bytes.

TNPP version 3.8 defines a group call feature that permits a sender to specify a list of recipients to receive a single message. This would be useful for email messages that often have more than one recipient. Unfortunately, most paging terminals currently do not support this feature. This means that the paging terminals must send multiple copies of the same message in order to support mailing lists.

Many infrastructure equipment manufacturers have developed extensions to the TNPP specification. For example, some vendors support packet sizes up to 4096 bytes. These extensions can cause problems in getting various vendor's infrastructure equipment to "plug and play" together.

TNPP was not designed to be used for two-way messaging or for sending binary data. These are the main shortcomings of the protocol. Nevertheless, it is one of the most widely used terminal-to-terminal and terminal-to-control protocols currently in use in the paging industry.

5.2.1.3 TAP (Telocator Alphanumeric Protocol)

TAP is a protocol commonly used to send alpha text messages from devices such as PCs and page entry devices to pagers. It is an ASCII-based protocol that supports both an automated and a manual mode.

Each block of information sent to the paging system must be no longer than 256 characters, comprising not more than 250 information bytes plus 3 control characters and a 3-character checksum. Multiple blocks are allowed by the protocol. The protocol does not impose limits on numbers of blocks. However, user or carrier systems may impose limits on total message size or total number of blocks that may be sent in a single connection or session.

The protocol supports sending messages to pagers. Each page message block includes a Pager ID and a message. The Pager ID is typically the PIN of the subscriber who is to receive the message.

Any ASCII character with a value less than or equal to 0x7F (DEL) may be used in the message. Some older systems do not support non-printable ASCII characters less than 0x20 (SPACE). TAP does not support binary data types.

TAP is a simple one-way paging protocol. It may be used to send information to either one-way or two-way messaging devices, but it does not support the receipt of information from two-way messaging devices.

5.2.1.3 SNPP (Simple Network Paging Protocol)

This simple protocol is used by some carriers as a method of submitting pages from PCs or other systems on the Internet to their paging network. It is an ASCII-based protocol that uses IP as the network protocol. The protocol permits a system to send pages and to query their delivery status.

Two-way messaging extensions have been defined in version 3 of the protocol to permit the sender to set multiple choice response (MCRs) in the pager, to confirm message delivery and to query for responses. The protocol is defined in RFC 1861.??will they know where to find this??

5.2.2 Mail and Internet Protocols

This section briefly mentions protocols that are commonly used to support Internet and email applications. Gateways that exchange information with paging systems may support these protocols. Most paging terminals do not support these protocols directly.

5.2.2.1 SMTP (Simple Mail Transfer Protocol)

SMTP is a protocol that is widely used to send mail across a network. It is documented in RFC 821.

The protocol permits the exchange of standard ASCII text messages, but does not define methods for sending other forms of information and attachments. However, the subject of exchanging non-text information is covered in RFCs 1521 and 1522, which cover Multipurpose Internet Mail Extensions (MIME).

The protocol provides for a sender to establish a connection to a receiver, which may be the final or an intermediate node in the destination path. The sender uses various SMTP commands to open a connection, request that mail be sent, specify message recipients, define the message content, and close a connection. The recipient replies to all messages with error and return codes indicating its ability to process the request. The protocol includes provisions for forwarding mail to hosts other than the one specified by the sender. Provisions are made for expanding mailing lists, verifying mailing lists and sending mail directly to a recipient's terminal screen.

5.2.2.2 HTTP (Hypertext Transfer Protocol)

HTTP is a stateless, connectionless, object-oriented protocol that is documented in RFC 2068. It is well suited for Internet browsing. While it is commonly used to send ASCII text or HTML pages from one Internet site to a web browser or other user agent, the protocol is not limited to sending these formats. It supports MIME-type encoding and user-specified character sets.

HTTP requires a reliable connection and is typically implemented to run over TCP/IP networks. However, any network that guarantees in order message delivery such as TCP can support HTTP.

Few if any paging terminals implement HTTP directly in currently deployed systems. However, HTTP is implemented in some gateways that are used as portals to paging systems.

HTML is not well suited for wireless devices. Alternatives based on XML have been proposed for use in wireless devices. These include HDML (Handheld Markup Language), WML (Wireless Markup Language), and VoxML™ (Voice Markup Language). HTTP can be used to exchange information encoded in any of these markup languages.

5.3 Protocols Within Paging Infrastructure

5.3.1 Terminal to Terminal

5.2.1.3 TNPP (Telocator Network Paging Protocol)

TNPP, as described earlier in section 5.2.1.1, can be used to submit messages to a paging system from an external system that has implemented the protocol, such as an Operator Assisted Paging (OAP) system. It can be also be used to route paging messages between terminals. Finally, it can be used to send CAP pages from paging terminals to paging system controllers.

Generally, TNPP ID pages are sent to paging terminals where the subscriber ID is converted to the pager's capcode. TNPP cap pages are sent to the system controller for encoding and distribution to transmitters. If an external system has its own subscriber database which contains the subscriber's capcode, then it can send TNPP cap pages directly to the system controller, bypassing the paging terminal. These possibilities are shown in Figure 8.

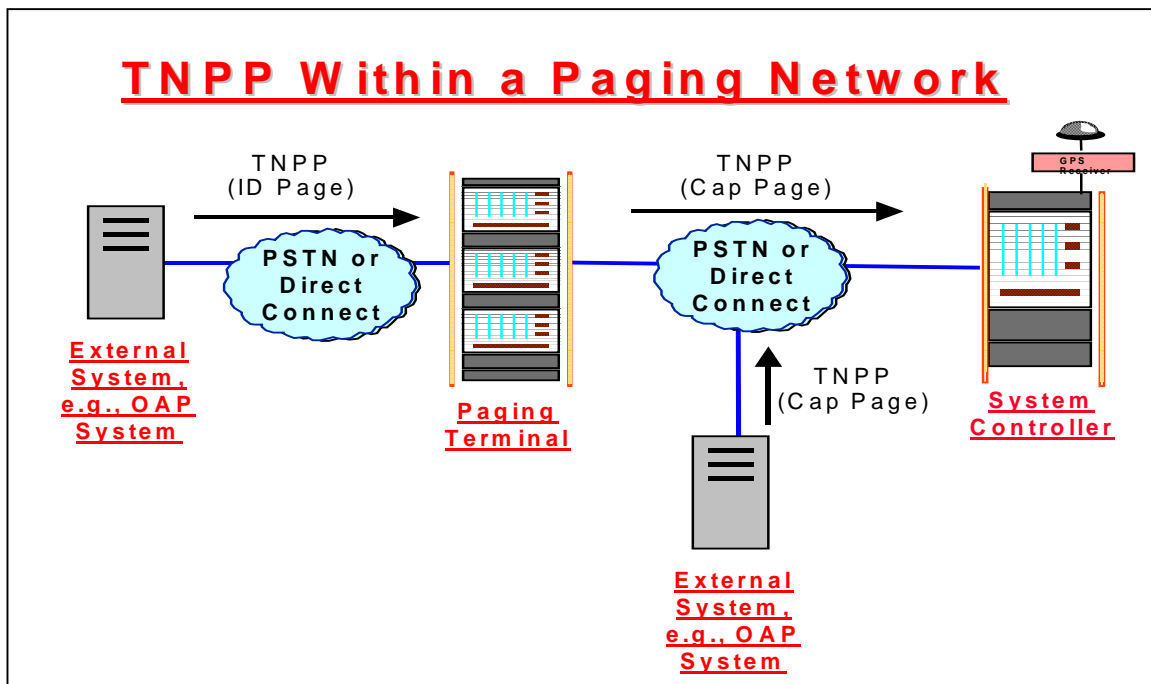


Figure 8. TNPP Within a Paging Network

5.3.1.2 WMtp (Wireless Messaging Transfer Protocol)

WMtp™ is a fairly new paging system protocol developed by Glenayre in consultation with Motorola. The protocol is designed for both one-way and two-way messaging. It was initially used to support voice messaging (using the InFLEXion protocol) in some US markets. Later it was used in Asia for one-way FLEX roaming systems. It is now being used to support two-way data (using ReFLEX) in several US markets. The protocol supports most one-way and two-way over the air paging protocols.

WMtp was designed to support the following features:

- Acknowledged one-way message delivery

- Acknowledged two-way message delivery
- Group delivery
- Targeted messaging to specific cells
- User response messages from a pager
- Automatic roaming
- Output congestion control

WMtp is an IP based protocol used to exchange numeric, alphanumeric and binary messages between terminals and system controllers in a paging system. It is currently specifies TCP/IP as the transport and network layer protocols, but the application layer protocol is independent of underlying layers. Any data link protocol that is able to carry TCP/IP may be used for WMtp. These include Point-to-Point Protocol (PPP), Serial Link Interface Protocol (SLIP), Frame Relay or X.25. WMtp supports both peer-to-peer and broadcast messaging.

WMtp sends and receives data as protocol data units (PDUs). A subset of the Remote Operation Service Element (ROSE) protocol is used to communicate between nodes in a WMtp network. ROSE packets are encoded using Abstract Syntax Notation One (ASN.1). The ROSE protocol is a request / response protocol that defines different classes of messages, some that require a response, some that don't. If a reply is required, a request message sent to a node will be answered by a reply in which the requested information is returned, or by an error or reject message indicating the request was erroneous or could not be handled by the node.

The use of ASN.1 permits WMtp nodes to exchange different data types transparently. That is, it resolves data type dependencies that may exist in different nodes, such as byte ordering. WMtp also supports the exchange of BLOB (Binary Large Object) data types. This permits it to handle digitized voice and other arbitrary binary data types.

The protocol defines roles played by various components in the paging system. For a particular messaging session, a paging terminal may be an input terminal (MS-I), a home terminal (MS-H) or both. Input terminals accept calls, but they do not contain the requested subscriber's profile. They must forward requests to the home terminal to obtain information about the subscriber. Home terminals contain the requested subscriber's profile. A terminal may act in both roles for a single messaging session. That is, a caller may dial into a paging terminal that holds the profile of the requested subscriber. In this case, the terminal acts both as the MS-I and the MS-H.

In some specialized applications, external systems such as operator assisted paging (OAP) systems have implemented parts of WMtp in order to behave as an input terminal (MS-I). This permits these systems to appear to the paging system as another paging terminal that can exchange messages with nodes that hold the subscriber database.

Systems that support WMtp are set up as mesh networks. Every paging terminal is able to directly access every other paging terminal in this network. That is, the paging terminals do not act as intermediaries between one paging terminal and another. This is illustrated in the following figure.

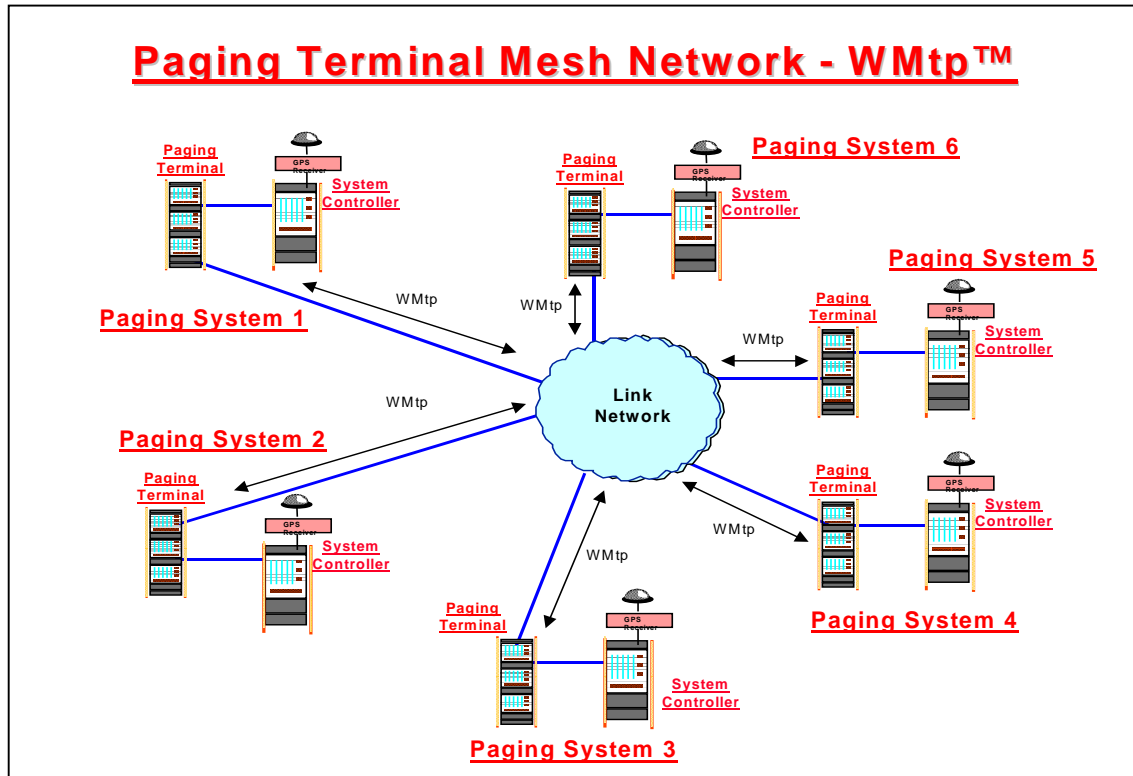


Figure 9. Paging Terminal Mesh Network for WMtp

Figure 9 shows six different paging systems interconnected via a link network. Each paging terminal contains a subset of the whole paging subscriber database. Callers may dial into any of the paging terminals to send messages to subscribers whose profile records may be in any of the terminals. The diagram shows each paging terminal connected to a single system controller. This is not a requirement of the protocol, but in fact, most existing paging systems that support WMtp are configured this way. The figure omits the transmitter and receiver components, which connect to the system controllers. Each system controller manages a set of transmitters and receivers.

Messaging through a paging system to subscriber devices using WMtp requires complex interactions between sub-systems that comprise the paging system. Sample flow diagrams are shown below which illustrate this complexity. Similar exchanges of information are necessary for other paging protocols.

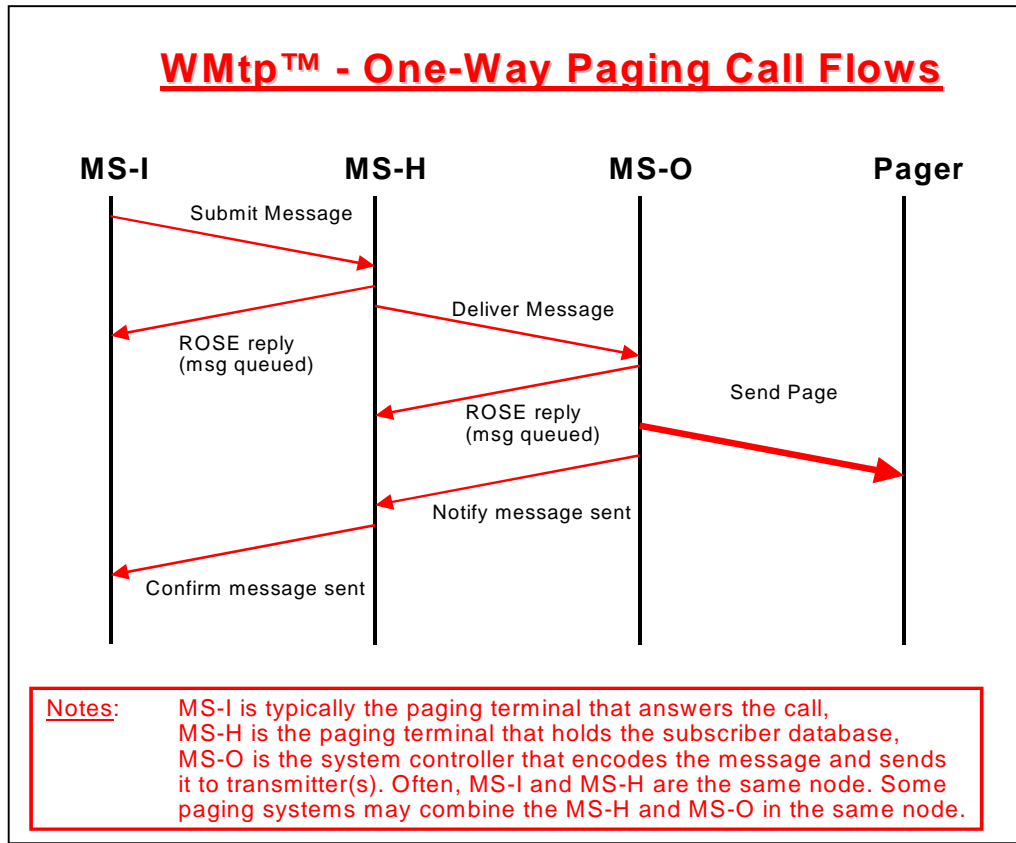


Figure 10. WMtp One-way Outbound Call Flow

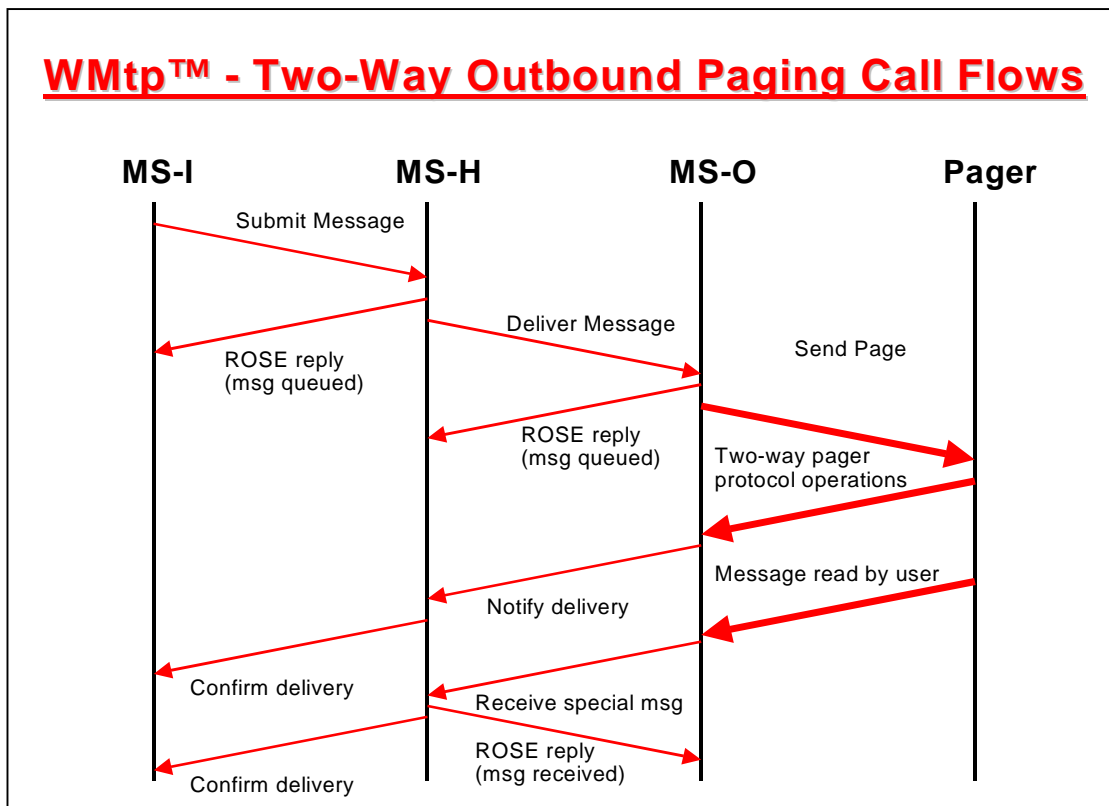


Figure 11. WMtp Two-way Outbound Call Flow

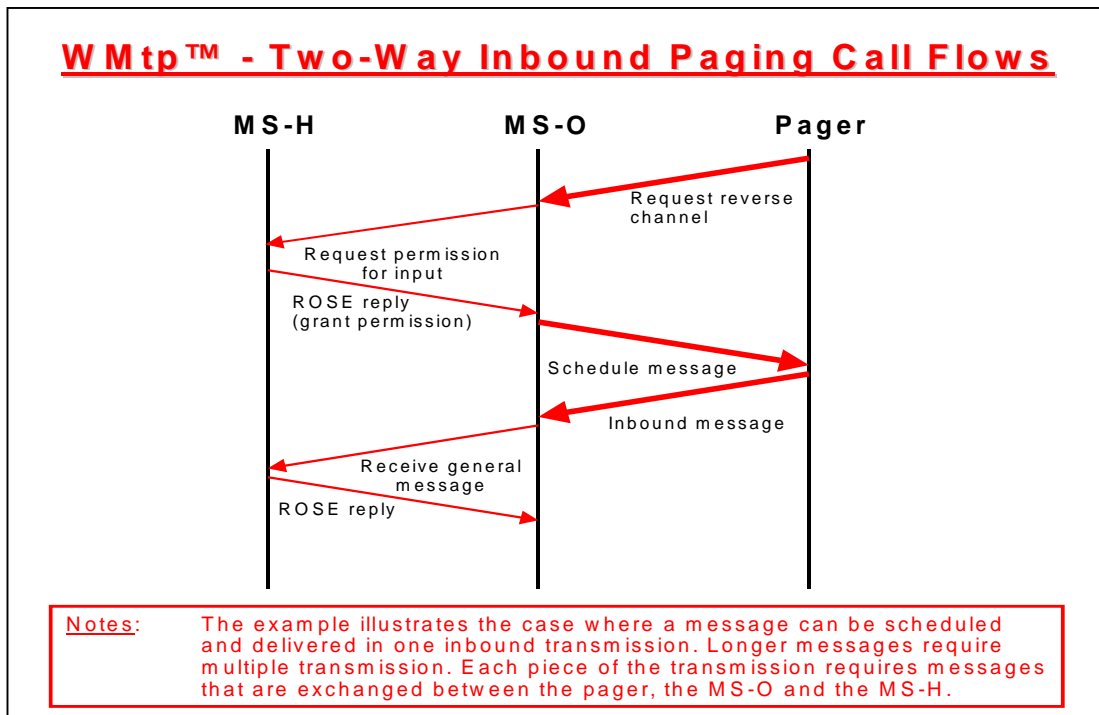


Figure 12. WMtp Two-way Inbound Call Flow

Tip: The point of the call flows is that considerable activity occurs within a paging system in support of a single inbound or outbound page. All of this increases message latency. Message exchanges should be minimized by combining short messages into longer ones and limiting total message size.

5.3.2 Terminal To System Control

5.2.1.3 TNPP (Telocator Network Paging Protocol)

TNPP, as described in section 5.2.1.1 and 5.3.1.1, can be used to submit messages to a paging system from an external system that has implemented the protocol, such as an Operator Assisted Paging (OAP) system. It can be also be used to route paging messages between terminals. Finally, it can be used to send CAP pages from paging terminals to paging system controllers.

5.2.1.3 WMtp (Wireless Messaging Transfer Protocol)

WMtp™, as described in section 5.3.1.2, is a terminal-to-terminal protocol. It also supports messaging between paging terminals and system controllers.

The system controller performs a role in the protocol as the output switch (MS-O). Some infrastructure suppliers combine the terminal and system controller roles in a single integrated system. Others separate these functions in different system components.

Call flows shown in Figures 10 through 12 illustrate the involvement of the system controller (MS-O) in the messaging flow in WMtp networks.

5.3.3 System Control to Transmitters

Protocols used between the system controller and transmitters are for the most part closed protocols, proprietary to particular infrastructure suppliers. Therefore, it is not possible to use one supplier's system controller with another supplier's transmitter controller, since one controller doesn't understand the other's protocol.

Some protocols are packet based using IP, others require synchronous data, circuit-switched networks. IP-based protocols usually use UDP at the transport layer, since it is a more efficient protocol than TDP in terms of distribution network overhead.

Protocols used between system controllers and transmitters must provide for more than just sending pages. Command and control information may be sent to the transmitter to change configuration settings, control the operation of the transmitter, initiate software downloads, etc. These command and control information packets will be defined by the protocol unless separate protocols such as SNMP are used to manage the transmitters.

The protocols must include provisions for selecting which transmitter or groups of transmitters should simulcast the message. This is necessary since some networks such as satellite based networks deliver a stream of information from the system controller to every transmitter in the network. Transmitters must be told whether they should accept or reject the message. The protocols also provide for selection of the frequency and the over the air protocol that should be used to simulcast the message.

Most protocols include error detection and correction capabilities. This overhead may result in transmitter link efficiencies of less than 100%. That is, the amount of information being sent across the distribution network from the system controller to the transmitter can be considerably more than the information simulcast from the transmitters to the pagers. This reduces the number of paging channels that can be supported over a particular distribution link. For example, a 19200 bps distribution link may only be able to support two 6400 bps paging channels due to this overhead. Some advanced protocols use compression techniques that minimize the overhead so that link efficiencies approach, and in some cases exceed 100%. These protocols allow a 19200 bps distribution link to carry three 6400 bps paging channels. The impact to carriers who deploy infrastructure that support these advanced protocols is reduced link distribution costs.

5.3.4 Receivers to System Control

Some standardization has occurred in protocols used between receivers and system control. The Inbound Paging Protocol (IPP) was jointly developed by Motorola and Glenayre. This standardization permits the mixing of receivers manufactured by various infrastructure suppliers.

The IPP protocol was designed with the following goals in mind:

- Reliable and efficient delivery of inbound paging messages
- Reliable and efficient delivery of alarm and maintenance messages
- Support for standard network topologies, concepts and protocols
- Biased for uni-directional data delivery.
- Lightweight, simple, and flexible protocol, suitable for implementation in embedded receiver controllers.
- Portable, expandable, and flexible network hierarchy
- Support for simplex and duplex network topologies

IPP is IP based. The current implementation defines UDP as the transport protocol. It uses ASN.1 and basic encoding rules (BER) to produce UDP packets.

The protocol defines the messages from receivers to system controllers and receiver concentrators. These messages have mandatory and optional fields. The protocol requires each receiver to send a Receive Data

APDU (application protocol data unit) for each frame (see FLEX and ReFLEX protocols for discussion of frames) even if no inbound page data is included.

Information included in the Receive Data APDU includes receiver ID, frequency of inbound channel, frame and cycle number, channel signaling speed, and paging data (optional). If paging data is included, it contains an indication of signal strength, the actual received data, and error detection information. Within the paging data field are the address of the subscriber unit, information bits, message sequence number, pager type (ReFLEX 25 or 50), and page data.

The IPP protocol supports the sending of APDUs to more than one system controller. This permits sharing of receivers by different paging systems. Most receiver suppliers have implemented this protocol.

5.4 Protocols Between Paging Infrastructure and Subscriber Devices

The following sections discuss over-the-air protocols used between the paging system and the subscriber devices. Subscriber devices generally support a single over-the-air paging protocol. Both one-way and two-way protocols are discussed.

5.4.1 One-way Over The Air Paging Protocols

This section discusses one-way protocols used in paging systems. Programmable devices used on paging systems are all two-way devices. Information on one-way protocols is nevertheless useful because the two-way paging protocols are built on a foundation defined for one-way protocols. Most of the section deals with the FLEX™ protocol, Motorola's high-speed one-way paging protocol that has been adopted as a standard in most regions of the world. It is the protocol on which the predominant two-way protocols are based. However, we begin the discussion with other widely used one-way paging protocols.

5.4.1.1 POCSAG (Post Office Code Standardization Advisory Group)

POCSAG is a one-way protocol that supports data rates of 512, 1200 or 2400 bps. This protocol has been in use for over 10 years. It supports the mixing of different paging protocols or different POCSAG speeds on the same paging channel. The protocol is still used in older one-way numeric and alpha pagers. It does not support binary data types or roaming. It is supported on any available paging frequency with an outbound channel bandwidth of 25 kHz.

5.4.1.2 ERMES (European Radio Message System)

ERMES is a one-way protocol adopted most extensively in Europe. It supports numeric, alpha and binary data types. Outbound data rates are 6250 bps. ERMES operates in the 169.425 to 169.800 MHz frequency bands in 25 kHz channels. Roaming is supported between ERMES systems.

5.2.1.3 FLEX

FLEX™ is Motorola's high-speed one-way paging protocol. Its framing structure is used in the two-way paging protocols discussed below. FLEX supports numeric, alpha and binary data types. It can operate on any available paging frequency and uses a 25 kHz outbound channel. It supports data rates of 1600, 3200 and 6400 bps. It has a defined roaming capability. It also defines a group call capability that permits a single message to be efficiently transmitted to many recipients.

FLEX uses a frame structure where each frame is 1.875 seconds long. Exactly 128 frames make up a complete paging cycle. A cycle therefore lasts 4 minutes (1.875 sec/frame x 128 frames/cycle).

FLEX pagers can be programmed to listen for messages in specific frames. Their battery life depends on how frequently they "wake up" and listen for messages. They can listen as frequently as every frame (128 times in 4 minutes), or as infrequently as once a cycle (one time in 4 minutes). The frequency is controlled

by what is called a collapse value ©. The frames that the pager will look at are determined by the collapse value according to the expression 2^c . This means the pager may wake up every frame, every other frame, every 4th frame, every 8th frame, ... , or every 128th frame. Obviously, there is a direct tradeoff of battery life and message latency that is greatly influenced by the collapse value setting. Collapse value can be set in the pager but can be changed by the paging system.

The FLEX protocol defines different types of fields that contain different types of information: system configuration & control, synchronization, pager address, user data and others. A single FLEX frame contains combinations of these types of fields; the user data being only part of the information that is transmitted.

Data is encapsulated in units of codewords, each of which is 32 bits (4 octets) long. Within a codeword, only 21 bits are available for information, the other 11 bits are used for error detection. Depending upon the type of data being transmitted, parts of one or more codewords may contain encoding or other control information rather than actual user data. What this all means is that the effective user data rate for FLEX is less than 65% (21 bits / 32 bits) of the actual data rate. That is, a 6400 bps FLEX channel really carries user data at a rate that is less than 4160 bps. The rest of the data rate is used for overhead.

User data in FLEX is packed into codewords without regard to codeword boundaries. That is, information is simply stuffed into the 21 available information bits in a codeword until the codeword is filled. Stuffing then continues into the next codeword until it is filled. This pattern repeats across codewords until the user data payload is completely packed into codewords.

Numeric data is encoded using 4 bits per digit. Alpha data is encoded using 7 bits per character. Hex / binary data is stored as individual bits.

Tip: The packing and encoding rules have implications to the developer. It is generally more efficient to send numeric data as binary than as either numeric or alpha. For example, to send the number 65535 ($2^{16} - 1$) as a binary encoded number requires 16 bits which will fit in a single 32 bit codeword (actually within the 21 information bits). Sending it as a numeric message requires 20 information bits (5 digits x 4 bits / digit), and sending it as an alpha string requires 35 bits (5 digits x 7 bits / numeric character).

The maximum numeric message that can be sent in a frame is 8 codewords containing up to 41 digits. Numeric messages cannot be continued across frames. The maximum alpha message that can be sent in a frame is 84 codewords or 252 characters. The maximum hex / binary message that can be sent in a frame is 84 codewords or 1764 bits. Alpha and hex / binary messages can be continued in subsequent frames with no specified maximum length. However, carriers generally enforce message size limits of a few hundred alpha characters. The following table summarizes the information presented above.

Table 1. Maximum Data Limits in FLEX

	Numeric Message	Alpha Message	Hex / Binary Message
Max. codewords per frame	8	83 in 1 st frame, 84 in subsequent frames	83 in 1 st frame, 84 in subsequent frames
Max. user data per frame	41 <u>digits</u>	249 <u>characters</u> in 1 st frame, 252 in subsequent frames	1743 <u>bits</u> in 1 st frame, 1764 in subsequent frames
Continued across frames?	No	Yes (no frame limit)	Yes (no frame limit)
Maximum user data imposed by protocol	41 digits	No limit	No limit

Tip: The significance of the above information to the developer is that message latency increases and battery life decreases as more frames are needed to transmit data. Therefore, developers should attempt to use the minimal number of frames to transmit data.

5.4.2 Two-way Over The Air Paging Protocols

The ReFLEX and InFLEXion protocols are built on the FLEX protocol foundation. These are the protocols supported in the current two-way pagers. The following sections will describe each of these protocols at a high level.

5.4.2.1 ReFLEX 50

ReFLEX 50 is a two-way protocol intended for data applications. The first versions of the specification were released in February 1994. The stated purpose of the protocol is to “provide an asymmetrical high capacity two-way signaling system that uses a paging based protocol on the forward channel”. The asymmetrical nature of the protocol is reflected by the different data rates supported on the inbound vs. outbound channels. The phrase high capacity must be understood in the context of the paging environment when the protocol was developed. The data rates supported by the protocol were a definite improvement over what was supported by earlier protocols. When compared to data rates available today in wired networks and anticipated for future 3G wireless networks, the term high capacity is misleading.

ReFLEX 50 systems operate in North America using the 930-931 MHz and 940-941 MHz outbound bands, and 901-902 MHz inbound bands. Outbound channels are 50 kHz; inbound channels are 12.5 kHz.

Features of the protocol include:

- From one to four forward 4-FSK subchannels each running at 6400 bps, for a total “raw” throughput of up to 25.6 kbps in a 50 kHz band.
- Support for up to seven FDD reverse channels and one TDD reverse channel, each running at 9600 bps. The TDD channel can operate within the same 50 kHz band as the forward channels.

Tip: As mentioned earlier for FLEX, a distinction must be made between raw data rates and effective data rates. Raw data rates are reduced due to overhead defined by the protocol. A rule of thumb for wireless environments is that effective data rates are roughly 50% of raw data rates.

The protocol is a synchronous time slot protocol tied to an accurate time reference such as GPS. The characteristics of the forward or outbound data channels are similar to those described for FLEX, and will not be repeated here.

The protocol defines personal, information services and global addresses. Personal addresses are used to deliver messages to a particular subscriber. Information services addresses permit the same information to be efficiently delivered to a group of subscribers; one message is sent to one address and is received and displayed by all pagers that have the information service address enabled. A single global address is defined that permits all pagers to receive a message, such as a system wide alert.

The reverse or inbound channel is synchronized to the forward channel. The TDD channel may operate in the same 50 kHz band as the forward channels. The FDD channels operate in a different band. The protocol supports both scheduled and unscheduled ALOHA random access inbound messages. Unscheduled messages are used for requests for scheduled transmissions, for pager status messages, etc. The protocol permits two-way pagers to request from the paging system a time slot for sending inbound messages. This approach is preferable to random inbound messages because it reduces message collisions, and improves inbound channel efficiency. The reverse channel protocol defines error detection and correction encoding.

The boundary between scheduled and unscheduled ALOHA transmissions on the inbound channel is configurable. This means that the amount of time allocated on the inbound channel for scheduled versus unscheduled transmissions can be changed.

Tip: The main point to developers concerning the configurability of the boundary between scheduled and unscheduled messages is that not all of the available inbound bandwidth can be used to send messages. Some of it is reserved for ALOHA control type messages. The split is generally determined by the carrier based on needs of the inbound channel.

The maximum message length that can be scheduled on the inbound channel is 2048 bytes when using the standard “Reservation Request”, or 18424 bytes when using a “Long Reservation Request”.

Tip: A developer should try to keep inbound messages shorter than 2048 bytes since carriers may not support message lengths longer than this.

The protocol defines many different message types on the outbound channel. Some of these include:

- Secure message used for Over The Air Programming (OTAP) of various features in the pager.
- Short message or tone only
- Numeric message
- Alphanumeric message
- Hex / binary message
- Various command and control messages

The protocol includes provisions for error detection and correction encoding, and for requesting erroneous data packets. It uses an ARQ sliding window scheme.

5.4.2.2 ReFLEX 25

The ReFLEX 25 is a two-way data protocol, first released in June 1995. It is similar to ReFLEX 50.

ReFLEX 25 systems operate in North America using the 929-932 MHz and 940-941 MHz outbound bands, and 896-902 MHz inbound bands. Outbound channels may be either 25 or 50 kHz; inbound channels are 12.5 kHz.

ReFLEX 25 features include the following:

- A synchronous frame structure similar to and compatible with FLEX
- Support for one to eight channels on outbound (system to pager) control path, each of which can be modulated independently
- Support of outbound data rates of 1600, 3200 and 6400 bps.
- Supports up to three subchannels of 12.5 kHz in each 50 kHz band.
- Inbound (pager to system) channel packet structure synchronized to outbound frame structure.
- Inbound channel support for up to eight 12.5 kHz channels.
- Inbound data rates for ReFLEX 25 of 800, 1600, 6400 or 9600 bps.

Much of the discussion of ReFLEX 50 applies to ReFLEX 25, and won't be repeated. Some differences in the two protocols include:

- ReFLEX 25 defines personal and information services addresses, but not a global address.
- ReFLEX 25 defines inbound data rates of 800, 1600, 6400 and 9600 bps, whereas ReFLEX 50 uses 9600 bps.
- The slower inbound data rates (800 and 1600 bps) allowed for ReFLEX 25 permit a smaller number of bits in a data packet (128 bits) than can be supported with higher data rates (6400 and 9600 bps support 154 bits per data packet).

The amount of data that can be packed into a single inbound message depends on several system settings. The protocol defines a single inbound packet as comprising either 128 or 154 bits depending on the inbound data rate. The actual number of user data information bits varies based on the number of data packets that are requested in a single data unit (from one to eight data packets make up a Data Unit). The number of packets in a Data Unit is specified by the system controller prior to the start of the inbound

transmission. The protocol supports sending multiple Data Units per message, but all transmissions must fit within a single frame.

The following table lists the total number of bits and number of user data bits available in a single Data Unit for each inbound data rate and each combination of data packets per data unit. The efficiency is shown as a percentage in parentheses, and is simply the number of user data bits divided by total data bits. Remember, more than one Data Unit can be scheduled and transmitted to handle messages longer than the number of user bits shown in the table. In fact, from 1 to 32 Data Units may be requested in a single inbound transmission.

Table 2. Breakdown of User Data Bits vs. Total Data Bits For Different Data Rates and Packets / DU

Inbound data rate ->	800 or 1600 bps	6400 or 9600 bps
<u>1 packet / Data Unit</u>		
Total bits : User Bits (% of total)	128 total : 68 user (53.1%)	154 total : 94 user (61.0%)
<u>2 packets / Data Unit</u>		
Total bits : User Bits (% of total)	256 total : 156 user (60.9%)	308 total : 208 user (67.5%)
<u>3 packets / Data Unit</u>		
Total bits : User Bits (% of total)	384 total : 244 user (63.5%)	462 total : 322 user (69.7%)
<u>4 packets / Data Unit</u>		
Total bits : User Bits (% of total)	512 total : 332 user (64.8%)	616 total : 436 user (70.8%)
<u>5 packets / Data Unit</u>		
Total bits : User Bits (% of total)	640 total : 420 user (65.6%)	770 total : 550 user (71.4%)
<u>6 packets / Data Unit</u>		
Total bits : User Bits (% of total)	768 total : 508 user (66.1%)	924 total : 664 user (71.9%)
<u>7 packets / Data Unit</u>		
Total bits : User Bits (% of total)	896 total : 596 user (66.5%)	1078 total : 778 user (72.2%)
<u>8 packets / Data Unit</u>		
Total bits : User Bits (% of total)	1024 total : 684 user (66.8%)	1232 total : 892 user (72.4%)

Tip: What Table 2 illustrates is that systems that operate inbound channels at higher data rates can achieve higher channel utilization in terms of percentage of user data versus total data. Furthermore, a carrier can achieve higher efficiencies with larger Data Units (more packets per DU), **but only if the Data Units are full**. ReFLEX 25 requires partially empty packets to be filled with a defined fill pattern rather than truncated. Because of the fill rather than truncate requirement, schedulers may specify smaller Data Units (fewer packets per DU) if most of the inbound traffic message lengths are short. This results in fewer unused bits, and can result in better overall channel utilization. Carriers may be able to provide guidance on selecting the optimal size for inbound messages on their networks.

5.4.2.3 InFLEXion

InFLEXion is a two way protocol designed for voice paging. Voice paging has not proven to be very successful in the paging market; demand did not meet industry expectations. Therefore, very little will be said about the InFLEXion protocol.

InFLEXion systems operate in North America using the 930-931 MHz and 940-941 MHz outbound bands, and 896-902 MHz inbound bands. Outbound channels are 50 kHz; inbound channels are 12.5 kHz.

A 50 kHz outbound channel can be subdivided into 7 InFLEXion subchannels, each 6.25 kHz wide. The inbound channel is used for message delivery and read acknowledgments, and various control messages.

The frame structure used for InFLEXion is very similar to that used for ReFLEX 25. InFLEXion uses linear AM-SSB modulation during voice transmissions to minimize bandwidth and maximize system capacity. Voice information is compressed before transmission.

6.0 Messaging Options

Paging systems are store and forward messaging systems. They support several forms of messaging, each of which can be useful in different ways. This section describes these forms of messaging.

6.1 Broadcast Messaging

Many information services use broadcast messaging to deliver content to many subscribers with a single message. The information is sent to a broadcast address. Any pager that has been pre-programmed with this broadcast address will receive the message. This is a very efficient form of messaging. This technique can be used to form groups of subscribers, sometimes called radio groups. A single message sent to a radio group address will be received by every pager that has this address programmed in its broadcast address list. This form of group messaging does not support confirmation of delivery. That is, a broadcast message is not acknowledged by the device.

An alternate form of group messaging is also possible. This form, sometimes called terminal group messaging is not really broadcast messaging, although it may appear that way to the user. Terminal group messaging simply involves setting up a list of subscribers in a paging terminal, and assigning this list of subscribers a unique subscriber number. The paging terminal knows that any message sent to this group subscriber should be sent to each subscriber in the list as a personal address message. A single inbound message request received by a paging terminal therefore results in multiple messages being sent to the system controller and transmitters. Since the device sees this message as a personal address message, it will acknowledge its receipt.

Broadcast messaging is regularly used within the paging infrastructure to send a single message, either a personal or a group message, to many transmitters. In the case of satellite distribution networks, all transmitters in the system will “hear” everything that is sent over the satellite link. The protocols used in these networks include transmitter addresses in the data stream which tell the transmitters whether the messages are for them. Some paging systems which incorporate IP packet switched distribution networks use broadcast IP to deliver messages or operations and maintenance commands to many transmitter sites.

6.2 Peer to Peer Messaging

This is the most common form of messaging used within the paging system. It is used for all personal messages. It requires the message to be addressed to an individual subscriber’s personal address. The unique address or capcode ensures that a single paging subscriber receives the message.

Peer to peer messaging is supported by two-way messaging protocols to send messages to a specific transmitter which is closest to a subscriber. Since two-way messaging devices have transmitters, they can tell the paging system where they are. They do this by identifying which transmitter’s signal is the strongest. This theoretically permits the paging system to target delivery to a single transmitter site. While two-way messaging protocols may support this targeted delivery, the paging infrastructure equipment used in some two-way systems may not support the capability.

6.3 Multicast Messaging

Multicast messaging is an efficient form of group messaging that is supported in some IP based distribution systems. This form of messaging permits a carrier to group transmitters so that a single message sent to this group address is distributed to all transmitters that belong to the group. This capability generally requires that routers in the distribution network support the message replication and distribution. This form of

messaging improves network link utilization, but is for the most part transparent to the subscriber and application developer.

Multicasting is not supported in the over the air paging protocols used for two-way messaging.

7.0 Challenges of the Wireless World

Writing applications for a wireless environment presents numerous challenges. While many of these challenges may exist in the wired world, they may be more pronounced in the wireless world. Application developers must be prepared to deal with these issues if they want their applications to behave reliably in a wireless environment.

7.1 Transmission Errors

Messages sent over wireless links are exposed to many forms of interference that can alter the content received by the target device or system. In fact, it is possible that messages are completely blocked due to some physical obstruction. The application developer must be prepared to handle both of these situations.

Over the air protocols contain sophisticated encoding rules to support error detection and correction. Two-way protocols typically provide support for retries when messages are sent but not successfully acknowledged by the receiving system or device.

Transmission problems can occur at any point in the distribution path. An application server may be offline, distribution links could be down, problems may exist in the paging system, interference may exist between the paging system and the messaging device, the subscriber's device may be turned off or its battery may be dead. Failures can occur at any point in the wireless transaction, before a request is initiated, in the middle of a request or after one is completed. The application developer must keep these possibilities in mind when designing error handling strategies.

7.2 Message Ordering

Message order is not guaranteed in all paging systems. Many paging systems implement sophisticated scheduling algorithms to maximize distribution link and over the air RF channel utilization. This goal requires the system controller to pack messages in the outbound queue as efficiently as possible. To do this, the controller may order message transmission based on message length as well as time in queue. The result is that messages may be re-ordered by the system controller. If message order is important, the application developer must handle it.

7.3 Coverage Problems

RF coverage is not universal in paging systems. Operators build out their paging systems by markets. They add additional markets and increase coverage in existing markets according to business plans which are driven both by revenue opportunities and availability of capital. Coverage is extended to include primary transportation routes connecting major markets before secondary routes are covered.

Some carriers have expanded one-way paging systems to support two-way messaging by adding receivers and upgrading paging system software. This may mean that coverage is more complete for outbound (paging system to subscriber device) messaging than for inbound. The affect is that a subscriber may be able to receive messages, but may not be able to send replies. Of course the opposite is also possible, depending on the location of transmitters and receivers.

7.4 Message Latency

Message latency is the time it takes to deliver a message. Many factors affect message latency. These include the queuing time in systems that handle the message, data rates across distribution networks, and processing time needed in various nodes along the path from point of origin to destination.

In paging systems, the queuing time in systems that handle the messages is usually the predominate source of message latency. Paging systems are store and forward systems. Paging protocols provide features that help maximize the use of available RF spectrum, an asset that is the most precious commodity to paging system operators. The protocols also tend to support features that help extend battery life in the subscriber devices. Unfortunately, it is not possible to achieve maximum RF utilization, maximum battery life and minimum message latency at same time. Tradeoffs must be made. Sophisticated message scheduling algorithms are used in the paging system to achieve an acceptable balance between these performance and utilization goals.

Paging system operators control the system configuration parameters that affect message latency, battery life, network and RF link utilization. The configuration parameters are set to optimize response and performance for the most typical application which is peer to peer messaging. Applications developed for paging systems must operate within performance and response constraints set by paging system operators.

Message latency is influenced by the specific paging protocols used. Some one-way, low speed paging protocols may actually achieve lower latency since the messages are not required to be sent in particular frames. Higher speed paging protocols such as those used for two-way messaging impose requirements on the paging system to send messages in the frames the subscriber device expects them. If sent at the wrong time, the subscriber device will be “asleep” and will miss the message. The amount of the paging cycle that a subscriber device is “awake” listening for messages is configurable. Devices that are programmed to wake up more often can send and receive messages more frequently. This reduces message latency and increases overall throughput, but shortens battery life.

A two-way paging system may be configured to send messages to subscriber devices as frequently as once a frame, or as infrequently as once a cycle. For the ReFLEX paging protocol, this represents a variation of between a few seconds to several minutes. Most paging systems try to keep this delay to less than 30 seconds.

Another less common source of message latency must be considered by application developers. It is possible that a message cannot be delivered to a subscriber at the time it is sent. This may be due to transmission problems, coverage problems or the fact that a subscriber’s device is turned off or has a dead battery. Two-way paging systems usually store non-delivered messages for quite some time waiting for the subscriber device to register with the system. Some systems simply retry transmission at various times until either the subscriber device acknowledges receipt, or some maximum retry value is reached. This possibility means that a device and hence an application running on it may receive “stale” information. Depending on the application, this could be as undesirable as not receiving the information at all. Consider the confusion caused by receiving a stock quote that is 2 days old, especially if the user is not told that it is 2 days old!

7.5 Network Capacity and Data Rate Limitations

Paging carriers will avoid hosting applications that result in excessive use of limited network resources. The most precious resource is usually available RF spectrum. Operators will attempt to block sources of data that place too much of a burden on their networks. Alternatively, paging carriers may price their service such that data intensive applications will be cost prohibitive to subscribers. What is “too much” may vary from carrier to carrier, or may vary within a carrier’s markets.

Paging protocols and RF channels support outbound data rates that range from less than 1200 to 6400 bps. Inbound data rates range from 1600 to 9600 bps. Effective data rates are lower than these numbers due to overhead needed for routing, packetizing, error detection and correction, etc.

Tip: A general rule of thumb is that effective data rates are around 50% of raw data rates in wireless systems. Another observation is that data rates in wired environments have always remained at least an order of magnitude greater than those in wireless environments.

The process of sending a message through a paging system is more complicated than just sending a single packet of information from source to destination. It involves a “call flow” where several messages are exchanged between various sub-systems in the paging system to control the handoff and delivery of the information. The complexity of sending messages coupled with the overhead incurred in sending a message means that it is sometimes more efficient to combine many small pieces of information into a single larger message than to send many small messages. The optimal message size may vary by carrier system.

7.6 Network Dependencies

The behaviors and features supported by a paging system may vary from one paging system to another, both between carriers and within a single carrier’s system. These differences are caused by several factors.

First, systems employ equipment provided by different paging infrastructure suppliers. Some paging carriers have developed some of the paging system components on their own. Different equipment supports different sets of features, capabilities and interfaces. Some of these have implications on how applications must be written to access information or route messages through the system. For example, an application may have to use one technique to access information from an application server when using one carrier’s network and use a different technique when using a different carrier’s network.

Most of the features supported in a paging system are provided through the software that runs in the various nodes that make up the paging system. Software version differences certainly exist across paging systems. This means that two paging systems built with the same hardware supplied by a single vendor may behave differently because the software releases loaded into the sub-system components are not the same.

A paging system is highly configurable to permit operators to manage performance, utilization, battery life, etc. Even if all paging carriers used exactly the same hardware and software from a single infrastructure vendor, their systems may be configured differently, since they may need to support different classes of subscribers or give preference to different applications.

All these network dependencies mean the performance and suitability of an application can vary from system to system. The application developer should be aware of this. Note that while it is possible an application could be suitable for some paging systems but not for others, it is much more likely that a poorly designed application would not be suitable for any paging system.

7.7 Protocol Capability vs. Network and Device Capability

As we discussed in the section on protocols, it is very likely that some components that makes up a paging system do not support the full range of capabilities allowed by the protocols used in the system. Some application developers may have a comprehensive understanding of some or all the protocols used within a paging system. They may assume that because a protocol says it supports a certain feature, the paging system must support it. Most of the time this is a valid assumption. However, it is wise to check with a carrier to make sure this is the case, especially for features that have crucial application design implications.

7.8 Security Issues

Any information that is transmitted in any form is subject to interception and alteration. For some applications, this is a critical concern.

The issue of security is tackled in various ways by paging system operators, protocol developers, paging infrastructure suppliers and subscriber device suppliers. The solution needed depends to some degree on the

physical configuration of the network, and the processes running on the network which support the application. The sensitivity of the information being transmitted will determine the level of security needed. Complete end to end security in an application almost certainly will require the application developer to implement it both at the server and the client (subscriber device) side.

Parts of a system can be made secure by physically or logically limiting access. For example, a database server may be placed behind a corporate firewall. Access to this server may be limited to a particular gateway. In this case, the information between the gateway and the database server need not be encrypted. Encryption would be implemented at the gateway and used from that point to the subscriber device.

Some paging devices, such as Motorola's PageWriter, support encryption. The encryption method used in the PageWriter is 128-bit RC4. The application developer must provide similar encryption / decryption support for the server side of the system. One solution to this is to use Motorola's FLEX Messaging Server (FMS) product.

Before implementing a system using encryption in a paging system environment, the developer must take care of a few details:

- Encryption uses the ReFLEX binary messaging (as opposed to the alphanumeric messaging that's usually used to send text). In order to use binary messaging some carriers may have to enable it, which means you may have to request that they do so.
- You may need to use a direct connection to the carrier's network to get binary data. To do this you need an account on the carrier's network. The direct connection is supported through Motorola's FLEX Messaging Server. If you do not want to use the FLEX Messaging Server, you need to contact the carrier for details on connecting directly to their network.
- Software is not currently available to support encrypted peer-to-peer messaging between all client devices. That is, you cannot send an encrypted message from your client device to a paging system and on to another client's device, unless you write a custom application that manages the encryption keys between units. (This is the current situation with respect to Motorola paging devices).
- You need software on the server side, of course, to decrypt messages. Again, Motorola's FLEX Messaging Server is capable of doing this.

8.0 General Recommendations and Guidelines

The previous section raises many issues that an application developer should be aware of. The intent is not to frighten, but to enlighten. Any type of application development presents a set of unique challenges. Those presented in the previous section must be considered when writing applications that are hosted on paging systems. The astute reader will recognize that many of them are common to any wireless environment.

This section gives general recommendations that a wireless application developer can use to best ensure his application will be successful.

8.1 Up-front Investigation

By far the most important recommendation is to do adequate up-front investigation before starting to design your application. You must understand the needs of your potential users and the requirements imposed by the networks and systems your application will rely on.

You should have a sound business plan that realistically considers potential number of users, the pricing model you intend to use, the cost for carrier services, marketing and sales support, software distribution provisions, customer care, etc. You should discuss the plan with your business partners, which should include the carriers, information service providers, potential investors, etc. on which you will depend. Up-front dialogue is important to ensure that the carriers and other partners are willing to support your application.

Important information may be found on potential business partner's web pages, bulletin boards, etc. Motorola maintains a developer support web page that may be helpful. Several useful sources of information are presented at the end of this document.

8.2 Application Architecture

The application architecture is perhaps the most important determinant of its suitability for a particular environment. All the optimization techniques in the world will not make up for a poorly architected solution.

Design goals for developing wireless applications should always include:

- Minimizing the total amount of data that is transmitted over the wireless link.
- Where possible, spreading the data traffic over time rather than incurring high burst rates.
- Anticipating and intelligently handling error conditions.
- Designing the user interface and application to deal with message latency.

The application developer has choices when designing an end to end solution. These include deciding on how functionality is partitioned between the server and client (device). For wireless applications, the decision must be driven in part on the design goals listed above.

Where possible, it is much better for the client to generate information than to have it transmitted over the air. For example, if your requirement is to obtain current loan rates and payment schedules for a potential home buyer, there are a couple of alternatives. One way is for the client application simply to request the current loan rate and 30 year payment schedule from a mortgage lender's website, and then display the results. This approach requires the payment schedule to be calculated by the server, and transmitted to the client. A far better approach for a wireless application is for the client to get current loan rates from the

mortgage lender, have the user select a loan term (say 30 years) and then have the application calculate and display payment schedule.

If a developer is only implementing a client application and intends to make use of existing data sources, the design choices are more limited. These may include limiting the amount of data requested. This is commonly done by web clipping applications, where data that is delivered to a wireless device is a limited subset of what is delivered to a wired PC or workstation.

8.3 Message Size and Data Packing

In the earlier discussions of the various paging protocols, it was pointed out that sending information over a paging system requires many complex message exchanges between the subscriber device and the paging system, and between the paging system and external servers. The packetizing and framing requirements of the paging protocols generally favor sending a single large message over sending many short messages. Therefore, it is generally better if the client collects all the user supplied data and packages it for transmission in a single message rather than sending each piece of information as a separate message.

Multiple messages may still be required to adhere to message size limits imposed by paging systems. However, the developer should try to stuff the maximum amount of information permitted into a single message. For example, let's assume that a wireless client application collects the following pieces of information from a user:

- User name (max of 48 characters)
- Address line 1 (max 48 characters)
- Address line 2 (max 48 characters)
- City (max 32 characters)
- State code (max 2 characters)
- Zip code (max 9 characters)
- Free form input field 1 (250 bytes)
- Free form input field 2 (250 bytes)

Furthermore, let's assume the paging system limits the maximum size of inbound messages to 500 characters of user data in a single message. The inferior solution is to send 8 separate messages, one for each piece of information. The preferred solution is to package the information, so that at most 2 messages are sent. For example, put name and address information in one message, and the two free form text fields in another.

Tip: When considering optimal packaging, you should find out from the carrier whether maximum message limits imposed by the paging system count just user data, or include message overhead as well. Your packaging scheme should not exceed user data limits, which may be less than maximum message length limits if overhead is counted.

The optimal message packaging may depend on the particular paging systems that you hope will host your application. The packaging should be done so that it will be acceptable on all intended paging systems. For example, if one system limits single inbound messages to 500 characters and another limits them to 2000 characters, and if the developer wants to offer the application to users on both systems, then the developer obviously needs to make sure that a single package of information is not larger than 500 characters. It would be terrific if the application could ask the paging system for its maximum message size and adjust the packing accordingly, but paging systems do not support this configuration query capability.

8.4 Data Representation

Data can often be represented in different forms. Numeric data can be encoded as hexadecimal numbers, as BCD digits or as alphanumeric characters. Alpha text may require different character sets for different languages.

In general, it requires fewer bits of information to send binary numeric data than either BCD or alpha characters. The following table shows some examples.

Table 3. Bits Needed Using Various Data Representations of Numeric Information

Number	Hex Encoded	BCD	Alpha string
1	0x1 (1 bit)	0001 (4 bits)	'1' (1 x 7 bits, 1 ASCII char)
255	0xFF (8 bits)	0010, 0101, 0101 (12 bits)	'255' (3 x 7 = 21 bits)
65,535	0xFFFF (16 bits)	0110, 0101, 0101, 0011, 0101 (20 bits)	'65535' (5 x 7 = 35 bits)
2,147,483,647	0xFFFFFFFF (32 bits)	0010, 0001, 0100, 0111, 0100, 1000, 0011, 0110, 0100, 0111 (40 bits)	'2147483647' (10 x 7 = 70 bits)

Similarly, time and date may be represented as hexadecimal numbers (UTC codes or just hex encoded numbers), numeric or alpha data.

The application developer should consider alternatives forms of data representation, and, where possible, pick the one that requires fewer bits.

Tip: The data representations that can be considered for an application very likely depend on the paging system, paging devices and / or server systems that make up the total solution. This is especially true for hex / binary data. Up front investigation of supported alternatives is highly recommended.

8.5 Data Compression and Encoding Techniques

Many data compression and encoding techniques are possible. We've already seen how selection of a particular data representation permits us to compress numeric data. A developer should look for all opportunities to encode information rather than send free form text over the air.

One common compression technique involves sending codes instead of text. We are all familiar with using social security number or employee ID numbers rather than name to identify a person. Similarly, states and provinces can be identified by 2 character codes.

Another technique that has been used in data processing systems over the years is to suppress repeated characters by using some encoding scheme. For example, rather than send 'AAAAAA' over the air, you could encode it to send the repeated character along with a repeat count. In the previous example, you could send 0x2, 0x6, 0x41. The 0x2 (sent as 8 bits) is a compression flag indicating the next byte is a repetition count. The 0x6 (sent as 8 bits) is the count. And, the 0x41 (sent as 8 bits) is the hexadecimal representation of the letter A. So, the alpha string of six repeated letters which would normally require 42 bits minimum (6 characters x 7 bits per ASCII character), can be compressed to just 24 bits.

Ideally, you should never send static information or information that can be derived over the air. You could for example send variable data as comma delimited fields, or fixed length, position dependent data streams.

The field number or position of data in a array of bits or characters (offset and length) determines what the field means. For example, a stock query application could query a stock trading system via some server and present the following information to a user on his wireless device:

<p>Stock: IBM – International Business Machines</p> <p>Trade date: November 10, 1999</p> <p>Trading range: 87-3/8 to 88-1/4</p> <p>Last trade: 87 7/8</p>

Figure 13. Sample Client Display Screen

Information shown in **bold font** in the above figure represents variable information; the rest is static. The server could send the complete screen image to the client, or it could simply return the variable information. Using the former approach, the server would send around 150 ASCII characters or 1050 bits (150 x 7) over the air. If the server just returns the variable data, using all available encoding techniques (e.g., represent date as 32 bit UTC, send numeric data as hexadecimal numbers, etc.), you could reduce the returned data to around 400 bits. In the second case, the client application supplies all the static information.

8.6 Managing Latency

Message latency is a fact of life in paging systems, given their store and forward nature. To make your application appealing to users, you must effectively deal with it.

We have already described the benefits of packing small pieces of information into a small number of larger messages as a way to maximize network efficiencies. This also has a benefit of reducing overall latency. If each request / response message through a paging system takes 30 seconds, it's far more acceptable to a user to suffer the wait once rather than for every piece of information that he supplies.

It is vitally important from a user friendliness standpoint to keep the user informed of the application's status when long running processes are underway. This is certainly true when a message transmission is in progress. It also applies when an operation is in progress on the device that is known to take more than a couple of seconds. In these cases, you should tell the user in some fashion that a wait is expected. For example, indicate on a status line that the request has been sent to a server, and that a reply is pending.

In some applications, it may be possible to continue working while a message request / response is in progress. These non-blocking type implementations give the user the impression that the latency is much shorter than it actually is. An example of how this can be used is in an application where a user requests entries from a database in some predictable fashion, for example by sequentially reading the records. The user interface screen could have an input field for the record key to access, and a form for the returned data. It also could have a **NEXT** and a **CANCEL** button. The client application obtains from the user the record key of the first record to read. It submits a database request for this record, and displays a message on the status line that a reply is pending. When the data is returned, the application displays the information to the user in the form. While the user is reading the returned information for the first record, the application quietly sends a request for the next record. Knowing that it takes some time for a user to read the returned information, and that a high probability exists that the user's next action is to request the next record, the application proactively makes the request. When the user presses the **NEXT** button, the new information may already be available. If not, the perceived delay should certainly be less than the actual delay. Of course, this technique generally means that one extra database access is requested than is needed. However, depending on the application and the amount of data returned, this may be an acceptable tradeoff.

Other non-blocking options are possible. The application could be designed to alert the user when the response is returned. The user submits the request, and then goes about his business. The client device alerts the user when the information is returned via an audible or vibrating alert mechanism. This approach frees the user from having to interact with the application while waiting on a reply.

8.7 Error Handling

The wireless world is much more prone to errors than the wired world. Interference from external sources is a greater problem. The likelihood that a power source fails, i.e., a device's battery is dead, is greater. More sub-systems and networks are usually involved in processing messages. Commercial applications deployed on wireless networks must anticipate and intelligently handle errors.

Paging systems tend to be highly reliable. However, situations can occur that prevent a message from being delivered in a timely fashion, or being delivered at all. These include congestion somewhere in the paging system, a failure in a non-redundant component of the paging system or network, a failure in the client device, or a problem with a server that supplies information or serves as a gateway to the paging system.

Two-way paging protocols provide mechanisms for reliable message delivery. They use ACKs / NAKs to indicate packets of information are delivered without error, or need to be re-transmitted. Systems will continue trying to deliver messages until system configured retry counters and timers expire.

The developer's approach to handling failed message receipt should be different in this environment than in one where receipt is not assured. In this environment, the "wrong" error handling technique is to send a failed request again. If it didn't get through the first time, the chances are very high it won't get through the second or third time either. This implies that the application should block a user from trying to send a request multiple times. Sending multiple requests can make a bad situation worse. A reply message may be held up in a system controller's message queue due to congestion in the system. Adding more messages is exactly the wrong thing to do! Blocking repeated requests can be done forcefully by disabling a **SEND** button, or passively by displaying a dialogue box explaining the problem, advising against re-sending the request, and then letting the user chose a course of action.

Although rare, it is possible that a message may arrive at a destination with extreme latency. This can happen because of a problem in the system that delays delivery. As mentioned, the paging system will periodically retry delivery, often for several days, until the message finally gets through, or the maximum retry interval expires. Applications should be prepared to deal with these stale messages. This can be done using timestamps indicating message origination time and date.

Another possibility that should be considered is that multiple messages sent through a paging system may arrive out of order. If in order delivery is required, the application must enforce it by using sequence numbers in the messages.

If the application implements a state machine, consideration should be given as to how to restore the state if something fails in the system. For example, the paging device could lock up and have to be reset. (After all, the paging device is controlled by embedded software, and this software, like any other, can break.). State may have to be restored in a server as well as the client.

9.0 Future Trends

This section discusses future trends. By its nature, much of this is speculative. However, the hope is that it provides useful forecasts of what the landscape might look like in the not too distant future.

9.1 New Protocols

Trials are underway for higher speed second generation data protocols such as GPRS on GSM systems. This protocol theoretically can operate at data rates which are an order of magnitude greater than current paging networks. However, achieving these higher data rates involves reserving significant numbers of GSM voice channels for data, a possibility that is not likely to happen initially. After all, voice is still the number one “killer app” for GSM. Nevertheless, initial deployments of GPRS should boost effective data rates by a factor of 2 or 3 over current paging systems. The success of systems that support GPRS will be determined in part by availability of wireless devices and applications.

Work is underway to define and finalize new protocols for 3rd generation wireless networks. CDMA2000, W-CDMA, etc. hold out the promise of much higher data rates. These efforts should help spur acceptance of wireless Internet opportunities. However, past experience with other new technology rollouts shows that it will take some time from the point the standards are finalized to when systems are deployed to fully support the standards. Then it could take several years for the technology to be fully accepted in the market place. Many hurdles must be overcome before commercial service is offered, not the least of which are billing and customer service issues.

Focus is also being paid to data oriented protocols and technologies that extend the Internet to wireless devices. Protocols such as WAP, WML, HDML, VoxML are already supported in various servers and wireless devices. This effort will continue and accelerate as more and more content providers offer WML or HDML versions of their HTML pages. At the same time, the hope is that customers in ever increasing numbers will be attracted to these new devices and services.

9.2 New Devices & Operating Systems

Device suppliers are busy designing products that are ubiquitous (can be used on all types of networks). Several recent product announcements, such as Motorola’s announcement of a new DSP that is capable of supporting all the new 3G protocols, bring this promise closer to reality. However, wireless device product development cycles are fairly long. It will take time to develop and market these devices.

Device manufacturers are working with major players in the software industry to develop operating systems that are suitable for small wireless devices. Future devices are much more likely to support these standard operating systems than proprietary ones. This makes it easier for developers to write applications that will work across different vendor’s products. However, migrating to new operating systems in devices will take some time.

9.3 Support for Standards

All the key players in the wireless Internet space recognize the need for standards. We are moving away from proprietary components to open standards that enable interoperability.

Standards will be developed and adopted for network protocols, for Internet content markup languages, and for operating systems. It is even feasible that standard SDKs will emerge which would simplify application development across wireless devices and server platforms.

Manufacturers will continue to move toward using common hardware components and platforms as a way to reduce costs and development times. This will help ensure applications will be portable across devices.

10.0 Conclusions

The future appears to be bright for developers who embrace the challenges of writing applications for the wireless Internet. Future directions all point to continued emphasis on wireless Internet opportunities. Success depends on bringing together all the necessary components: infrastructure, networks, devices, billing and customer care systems, protocols, applications and the people that develop and manage all of these.

Success for application developers in this market will come at a price. Developers must learn the hard lessons of working in the wireless world. They must forge new relationships with carriers, device suppliers, content providers and others. They must understand the needs of potential users. They must take risks. They must get started. The early adopters are the ones that will climb the learning curve first and be poised to adopt future technologies as they arrive.

This paper discusses issues faced by developers who are interested in writing applications for paging systems. These systems are available today. All the necessary components are in place to begin the move from the wired to the wireless world.

Hopefully, this paper has achieved its goals of increasing the reader's understanding of paging systems, developing an appreciation on the part of carriers, suppliers and developers for working together, and spurring interest by developers for writing applications for these systems.

11.0 Where to Find More Information

This paper barely scratches the surface of developing wireless applications. More in depth information is available from the Internet and from various books, periodicals and magazines. This section lists some useful sources of information. This is not a comprehensive list, but may serve as useful starting point.

11.1 Internet Sites

The following sites have information that may be useful to developers:

- Motorola's Developer Support site: <http://www.motorola.com/spin>
- Motorola's Enterprise Solutions (FLEX Messaging Server, etc.) site: http://www.mot.com/MIMS/PSD/products/enterprise_solutions.html
- SkyTel's Developer Relations site: <http://www.skytel.com/develop/index.html>
- Pagemart's Developer site: <http://www.pagemart.com/partnercorner/developers/index.html>
- Glenayre's Home page: <http://www.glenayre.com/main.asp>
- Personal Communications Industry Association site: <http://www.pcia.com>

11.2 Useful Text Books

The following textbooks contain information that may be useful to developers.

- *Voice and Data Communications Handbook, Signature Edition*, Regis J. "Bud" Bates and Donald Gregory, © 1998, McGraw-Hill.
- *Databases on the Web, Designing and Programming for Network Access*, Patricia Ju, © 1997 by M&T Books.

11.3 Email Address

Feedback on this paper is welcome. Please send comments to one of the following email addresses:

flr004@email.mot.com

SPIN4DEV@email.mot.com

12.0 Acronyms and Glossary of Terms

Acronym or Term	Meaning
3G >	Third generation. Radio link protocols such as W-CDMA, CDMA2000 and TDMA offer higher data rates than second generation digital.
ACK	Acknowledgment. Indication of successful completion of task.
ALOHA	A channel access method whereby a message is sent in a non-scheduled fashion. If a collision occurs, a back-off occurs and another attempt is made.
AM	Amplitude Modulation. A modulation technique that involves changes to a signal carrier's amplitude.
APDU >	Application Protocol Data Unit. A packet of information.
API	Application Program Interface. A collection of defined software functions, classes, methods, etc., generally packaged as software binary libraries, that are used by programmers to take advantage of various system features.
ARQ	
ASCII	American Standard Code for Information Interchange.
ASN.1 >	Abstract Syntax Notation One. An encoding technique that ensures different data types can be exchanged between systems having different data representations caused by different byte ordering, character sets, etc..
BER >	Basic Encoding Rules. Rules that determine how different data types are encoded.
BLOB >	Binary Large Object. A data type used by WMtp that may be any type of binary data, originally used for compressed voice files.
bps	Bits per second.
Caller	The person or machine that originates a messaging session with the paging system.
Capcode	A pager's internal address.
CDMA2000	A third generation wireless protocol which is an outgrowth of CDMA 1.
Code plug	The program and configuration image in a pager's memory.
CRC	Cyclic Redundancy Check. An error detection code.
CSU / DSU >	Channel Service Unit / Digital Service Unit. Devices that encapsulate information into the proper framing before distribution over a WAN.
DSP	Digital Signal Processor. A programmable device that is often used to provide embedded logic in wireless and other products.
DTMF >	Dual Tone Multi-Frequency. Tones generated in telephones used for in-band signaling.
DU	Data Unit. Packets of information.
ERMES >	European Radio Message System. A radio protocol developed by operators in Europe.

	in Europe.
FDD	Frequency Division Duplex.
FLEX >	The “high speed” one-way over the air paging protocol developed by Motorola.
FM >	Frequency Modulated. ???
FMS	FLEX Messaging Server. A product from Motorola that provides email and Internet access to paging systems.
Frame Relay	A simple connection oriented layer 2 protocol defined by the ITU-T used to transfer information between two end points.
FSK	Frequency Shift Keying. A modulation technique that involves changing the frequency of a signal carrier to convey digital information.
ftp >	File Transfer Protocol. A protocol used to reliably send files across a network.
GOTAP >	Generic Over The Air Programming. A set of generic commands used to modify specific attributes or configurations in pagers. These generic commands can be translated into pager specific commands needed to modify a pager’s code plug using OTAP.
GPRS >	General Packet Radio Service. A packet-based bearer that is being introduced on many GSM and TDMA mobile networks.
GPS	Global Positioning Satellite. A system of satellites maintained by the US government that provides location information and precise timing. Used by paging systems to synchronize simulcast.
GSM	Global System for Mobile Communications. A second-generation digital cellular radio standard developed in Europe but widely adopted around the world.
HDML	Handheld Markup Language. A markup language optimized for use in wireless hand held devices.
Hex	Hexadecimal.
HTML	Hypertext Markup Language. A markup language widely used to send information throughout the Internet. Suitable for wired environments.
HTTP >	Hypertext Transfer Protocol. A connectionless, stateless protocol well suited for browsing sites on the Internet.
Inbound	Message direction moving from a wireless device to the infrastructure.
InFLEXion >	An over the air RF protocol used to send compressed voice messages from transmitters to pagers that support the protocol.
IP >	Internet Protocol. The packet based network layer protocol used in many data networks. Predominant network layer protocol used to send data across the Internet.
IPP >	Inbound Paging Protocol. An open paging protocol used to send information from paging receivers to the system controller. It uses UDP/IP.
kbps	Kilobits per second. A data rate measure in units of a thousand bits per second.
kHz	Kilohertz. A frequency measure in units of a thousand cycles per second.

MCR >	Multiple Choice Response. The ability to provide a limited set of responses that a user can select from. In paging systems, the selection number rather than the text of the selection is sent across the paging system, resulting in more efficient use of the RF and network bandwidth.
MHz	Megahertz. A frequency measure in units of a million cycles per second.
MIME	Multipurpose Internet Mail Extensions. An RFC that defines methods of encoding various data types for use in sending them between systems on the Internet. Originally defined for mail enclosures, but extended to data types that can be sent using HTTP.
MS-H >	Messaging Switch – Home. Nomenclature used in WMtp to identify a node that contains the subscriber’s record that a caller wishes to page.
MS-I >	Messaging Switch – Input. Nomenclature used in WMtp to identify a node that handles the receipt of message request, typically from the PSTN. MS-I does not contain the subscriber record that the caller wishes to page.
MS-O >	Messaging Switch – Output. Nomenclature used in WMtp to identify a node that handles message scheduling and encoding prior to delivery to transmitters.
NAK	Negative Acknowledgment. Indication of failure in completion of task.
OAP >	Operator Assisted Paging. System involving human operators who receive and dispatch page requests, typically using display screens and systems that interface to a paging system..
OTA or OTAP >	Over The Air Programming. The ability to change information in a wireless device by sending information over the radio link.
Outbound	Message direction moving toward a wireless device from the infrastructure.
PC	Personal Computer.
PCS >	Personal Communications System. Cell based messaging systems, originally focused on enhanced voice services.
PDU >	Paging Data Unit. A packet of information containing paging data to be sent to a pager.
PIM	Personal Information Manager. A device that contains information organizers such as schedulers, TODO lists, etc.
PIN >	Personal Identification Number. A number used to identify a subscriber.
POCSAG >	Post Office Code Standardization Advisory Group. The name of a earlier generation one-way paging protocol featuring fairly slow data rates ranging from 512 to 2400 bps.
PPP >	Point-to-Point Protocol. A connection oriented, layer 2 protocol used with TCP/IP applications.
PSTN >	Public Switched Telephone Network. The circuit switches telephone system.
QWERTY	A standard typewriter keyboard.
Radio group	A group of subscribers who share a common broadcast address in their pagers. Messages sent to this common address are received by all subscribers in the group. A form of broadcast messaging.

ReFLEX 25	A second generation two-way paging over the air protocol developed by Motorola.
ReFLEX 50	A first generation two-way paging over the air protocol developed by Motorola.
RF >	Radio Frequency.
RFC	Request For Comment.
RIC >	Radio Identification Code. The capcode of a pager.
ROSE >	Remote Operation Service Element. An inter-node message exchange technique that supports a request / response paradigm.
RS-232	A specification for serial communications.
SDK >	Software Development Kit.
SLIP >	Serial Link Interface Protocol.
SMTP >	Simple Mail Transfer Protocol. A protocol used to transfer mail or other data between systems.
SNMP >	Simple Network Management Protocol. A simple light weight protocol used to manage devices, typically in a computer or communications system, including the networks.
SNPP	Simple Network Paging Protocol. A paging protocol used to send messages to a paging terminal.
SSB	Single Side Band. A modulation technique which suppresses one side band of an AM signal as a method to conserve bandwidth.
Subscriber	An individual who purchases service.
Subscriber ID	The identifier of a subscriber. In paging systems this may be a PIN or an actual telephone number.
Subscriber profile	Information maintained in a paging terminal describing the services available to a subscriber.
T1 / E1 >	Digital circuit switches telephone links operating at 1.544 Mbps and 2.048 Mbps, respectively.
TAP >	Telocator Alphanumeric Protocol. A paging protocol used to send alphanumeric messages to a paging terminal.
TCP >	Transmission Control Protocol. An packet protocol used at the transport layer, which is guaranteed to be reliable. It uses IP as the network protocol.
TDD	Time Division Duplex.
telnet	A terminal emulation protocol used to remotely access a computer.
Terminal group	A group of subscribers maintained as a list by a paging terminal so that sending a message to the group results in sending individual messages to each member of the group. One form of "broadcast" messaging.
TNPP >	Telocator Network Paging Protocol.
UDP >	User datagram protocol. An packet protocol used at the transport layer, which is not guaranteed to be reliable. It uses IP as the network protocol.
UTC	Universal Time Coordinated. Time is seconds from January 1, 1970. The time standard used in most UNIX systems.

	time standard used in most UNIX systems.
VoxML	Voice eXtensible Markup Language. A markup language that includes data types useful in voice applications.
WAN >	Wide Area Network. A network that generally involves long distance links.
WAP	Wireless Application Protocol. A specification for technology useful in developing wireless applications and services.
W-CDMA	Wideband CDMA. A third-generation, wireless protocol that is a migration path for GSM.
WML	Wireless Markup Language. A markup language optimized for use in wireless devices with limited capabilities.
WMtp	Wireless Messaging Transfer Protocol. A packet based paging protocol developed by Glenayre that is used to exchange messages between paging terminals and controllers in a paging system.
X.25	An ITU-T recommendation published in 1976 that defines a packet based interface protocol.
XML	Extensible Markup Language. A markup language that can be extended by defining new data types.