datagram. The total length field in the IP header is decremented by the size of the minimal forwarding header (8 or 12) and the header checksum field is recomputed.

12.2 WIRELESS APPLICATION PROTOCOL

The Wireless Application Protocol (WAP) is a universal, open standard developed by the WAP Forum to provide mobile users of wireless phones and other wireless terminals such as pagers and personal digital assistants (PDAs) access to telephony and information services, including the Internet and the Web. WAP is designed to work with all wireless network technologies (e.g., GSM, CDMA, and TDMA). WAP is based on existing Internet standards, such as IP, XML, HTML, and HTTP, as much as possible. It also includes security facilities. Ericsson, Motorola, Nokia, and Phone.com established the WAP Forum in 1997, which now has several hundred members. At the time of this writing, the current release of the WAP specification is version 2.0.

Strongly affecting the use of mobile phones and terminals for data services are the significant limitations of the devices and the networks that connect them. The devices have limited processors, memory, and battery life. The user interface is also limited, and the displays small. The wireless networks are characterized by relatively low bandwidth, high latency, and unpredictable availability and stability compared to wired connections. Moreover, all these features vary widely from terminal device to terminal device and from network to network. Finally, mobile, wireless users have different expectations and needs from other information systems users. For instance, mobile terminals must be extremely easy to use, much easier than workstations and personal computers. WAP is designed to deal with these challenges.

The WAP specification includes

- A programming model based on the WWW Programming Model
- A markup language, the Wireless Markup Language, adhering to XML
- A specification of a small browser suitable for a mobile, wireless terminal
- A lightweight communications protocol stack
- A framework for wireless telephony applications (WTAs)

The WAP specification consists of a number of different protocols and modules, whose relationship is depicted in Figure 12.8.

Architectural Overview

The WAP Programming Model is based on three elements: the client, the gateway, and the original server (Figure 12.9). HTTP is used between the gateway and the original server to transfer content. The gateway acts as a proxy server for the wireless domain. Its processor(s) provide services that offload the limited capabilities of the hand-held, mobile, wireless terminals. For example, the gateway provides DNS services, converts between WAP protocol stack and the WWW stack (HTTP and TCP/IP), encodes information from the Web into a more compact form that minimizes wireless communication, and, in the other direction, decodes the compacted form into standard Web communication conventions. The gateway also caches frequently requested information.

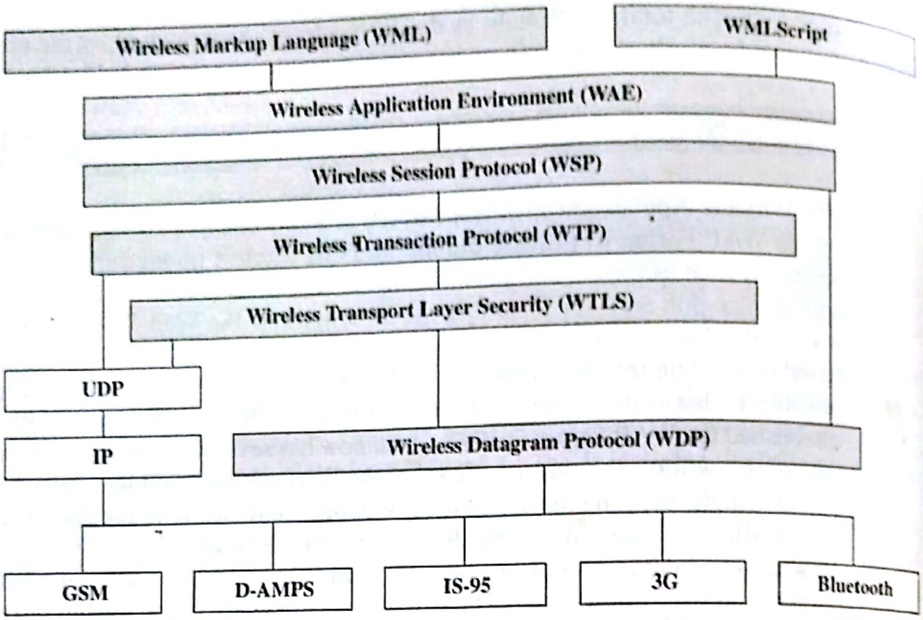

Figure 12.8 WAP Protocol Stack

Figure 12.10 illustrates key components in a WAP environment. Using WAP, a mobile user can browse Web content on an ordinary Web server. The Web server provides content in the form of HTML-coded pages that are transmitted using the standard Web protocol stack (HTTP/TCP/IP). The HTML content must go through an HTML filter, which may either be colocated with the WAP proxy or in a separate physical module. The filter translates the HTML content into WML content. If the filter is separate from the proxy, HTTP/TCP/IP is used to deliver the WML to the proxy. The proxy converts the WML to a more compact form known as binary WML and delivers it to the mobile user over a wireless network using the WAP protocol stack.

If the Web server is capable of directly generating WML content, then the WML is delivered using HTTP/TCP/IP to the proxy, which converts the WML to binary WML and then delivers it to the mobile node using WAP protocols.

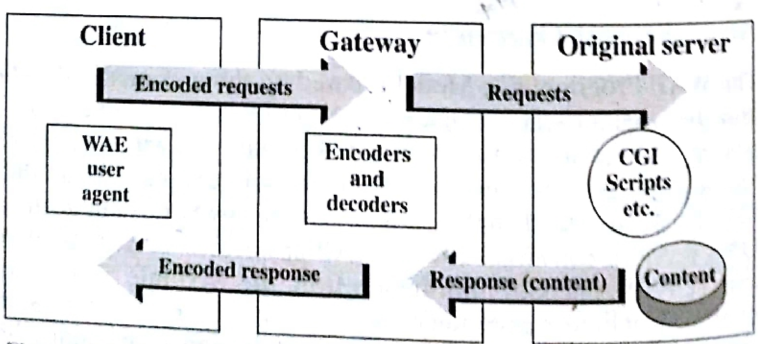

Figure 12.9 The WAP Programming Model

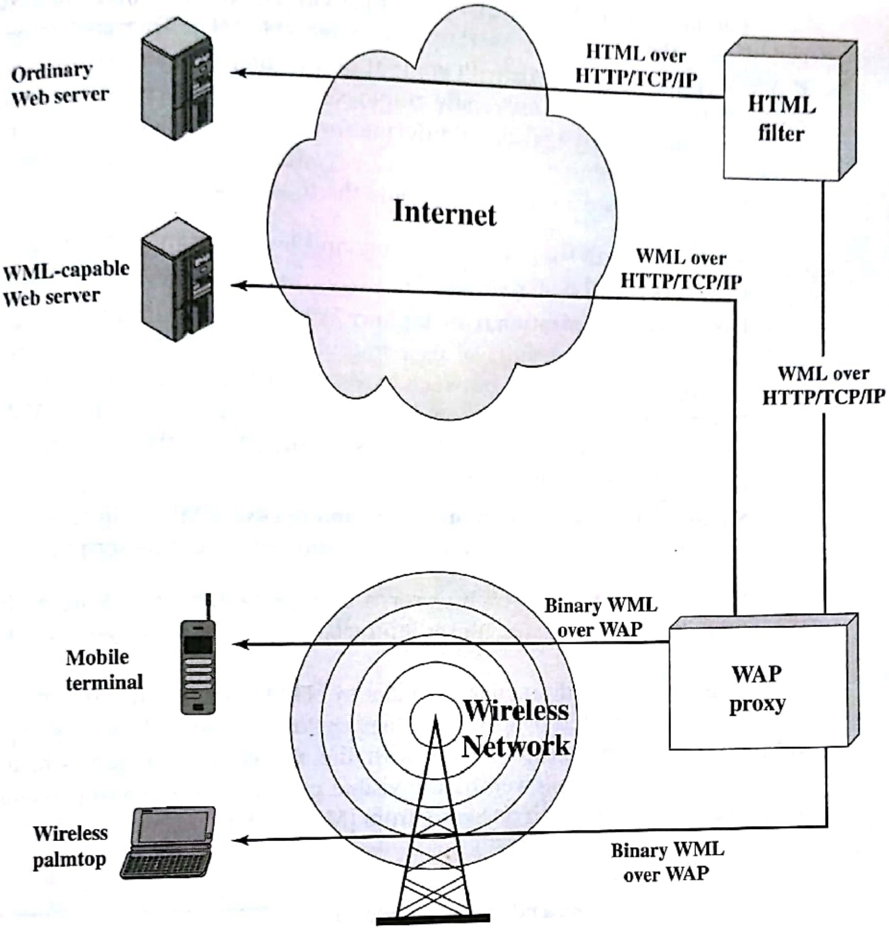

Figure 12.10 WAP Infrastructure

The WAP architecture is designed to cope with the two principal limitations of wireless Web access: the limitations of the mobile node (small screen size, limited input capability) and the low data rates of wireless digital networks. Even with the introduction of 3G wireless networks, which will provide broadband data rates, the small hand-held mobile nodes will continue to have limited input and display capabilities. Thus WAP or a similar capability will be needed for the indefinite future.

Wireless Markup Language

WML was designed to describe content and format for presenting data on devices with limited bandwidth, limited screen size, and limited user input capability. It is designed to work with telephone keypads, styluses, and other input devices common to mobile, wireless communication. WML permits the scaling of displays for use on two-line screens found in some small devices, as well as the larger screens found on smart phones.

For an ordinary PC, a Web browser provides content in the form of Web pages coded with the Hypertext Markup Language (HTML). To translate an HTML-coded Web page into WML with content and format suitable for wireless devices, much of the information, especially graphics and animation, must be stripped away. WML presents mainly text-based information that attempts to capture the essence of the Web page and that is organized for easy access for users of mobile devices.

Important features of WML include the following:

- **Text and image support:** Formatting and layout commands are provided for text and limited image capability.
- **Deck/card organizational metaphor:** WML documents are subdivided into small, well-defined units of user interaction called cards. Users navigate by moving back and forth between cards. A card specifies one or more units of interaction (a menu, a screen of text, or a text-entry field). A WML deck is similar to an HTML page in that it is identified by a Web address (URL) and is the unit of content transmission.
- **Support for navigation among cards and decks:** WML includes provisions for event handling, which is used for navigation or executing scripts.

In an HTML-based Web browser, a user navigates by clicking on links. At a WML-capable mobile device, a user interacts with cards, moving forward and back through the deck.

WML is a tagged language, similar to HTML, in which individual language elements are delineated by lowercase tags enclosed in angle brackets. Typically, the WML definition of a card begins with the nonvisible portion, which contains executable elements, followed by the visible content. As an example, consider the following simple deck with one card, from [MANN99]:

```
<wml>
    <card id='card1'>
    <p>
        Hello WAP World.
    <p>
    </card>
</wml>
```

The tags **<wml>**, **<card>**, and **<p>** enclose the deck, card, and paragraph, respectively. Like HMTL, most elements end with a terminating tag that is identical to the starting tag with the addition of the character **"\"**. When a wireless device receives this code, it will display the message "Hello WAP World" on the terminal's screen.

Table 12.3 lists the full set of WML tags, which are divided into eight functional groups.

WMLScript

WMLScript is a scripting language with similarities to JavaScript. It is designed for defining script-type programs in a user device with limited processing power and

Table 12.3 WML Tags

Tag	Description
Deck Structure	
<access>	Access control
<card>	Card definition
<head>	Deck-level information (meta, access, template)
<meta>	Meta information
<template>	Deck-level event bindings
<wml>	Deck definition
Content	
<img>	Image
<p>	Paragraph, visible content
<table>	Table
<td>	Table data
<tr>	Table row
Formatting	
<b>	Bold
<big>	Large font
 	Line break
<em>	Emphasis
<i>	Italic
<small>	Small font
<strong>	Strong font
<u>	Underline
User Input	
<fieldset>	Data entry items grouping
<input>	Data entry
<optgroup>	Subset of a choice list
<option>	Single choice in a list
<select>	Choice list
Variables	
<postfield>	Set an http request variable
<setvar>	Set a variable in a task
Timers	
<Timer>	Set a timer
Tasks	
<go>	Go to a URL
<noop>	No action
<prev>	Go to previous card
<refresh>	Screen redraw
Task/Event Bindings	
<a>	Abbreviated anchor
<anchor>	Anchor
<do>	Response to user button press
<onevent>	Intrinsic event binding

memory. Table 12.4 shows valid WMLScript statements. Important capabilities of WMLScript include the following:

- Check the validity of user input before it is sent to the content server.
- Access device facilities and peripherals.
- Interact with the user without introducing round trips to the origin server (e.g., display an error message).

Key WMLScript features include the following [WAPF98]:

- **JavaScript-based scripting language:** WMLScript is a subset of JavaScript, with some extensions.
- **Procedural logic:** WMLScript adds the power of procedural logic to the Wireless Application Environment (WAE), discussed subsequently.
- **Event based:** WMLScript may be invoked in response to certain user or environmental events.

Table 12.4 WMLScript Statements

Statement	Description
=	Assignment
break	Terminate the current loop
continue	Current loop iteration
for	Indexed loop
function	Function declaration
if..else	Conditional test
return	Exit the current function
var	Variable declaration
while	Boolean-controlled loop

- **Compiled implementation:** WMLScript can be compiled down to a more efficient byte code that is transported to the client.
- **Integrated into WAE:** WMLScript is fully integrated with the WML browser. This allows authors to construct their service using both WML and WMLScript.
- **Efficient extensible library support:** WMLScript can be used to expose and extend device functionality without changes to the device software.

Wireless Application Environment

The WAE specifies an application framework for wireless devices such as mobile telephones, pagers, and PDAs. In essence, the WAE consists of tools and formats that are intended to ease the task of developing applications and devices supported by WAP. The major elements of the WAE model are as follows (Figure 12.11):

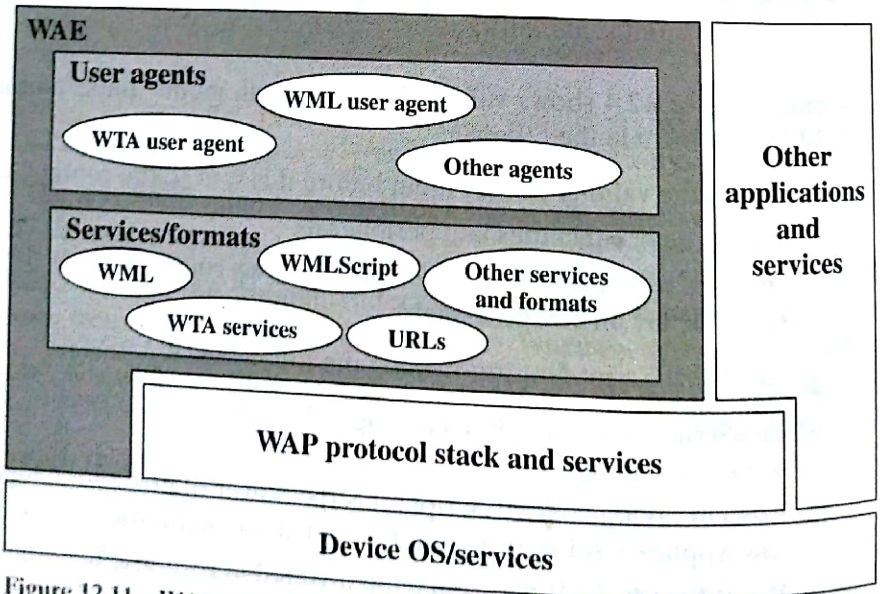

Figure 12.11 WAE Client Components [WAPF98]

- **WAE user agents:** Software that executes in the user's wireless device and that provides specific functionality (e.g., display content) to the end user.
- **Content generators:** Applications (or services) on origin servers (e.g., CGI scripts) that produce standard content formats in response to requests from user agents in the mobile terminal. WAE does not specify any standard content generators but expects that there will be a variety available running on typical HTTP origin servers commonly used in WWW today.
- **Standard content encoding:** Defined to allow a WAE user agent (e.g., a browser) to conveniently navigate Web content.
- **Wireless telephony applications (WTA):** A collection of telephony-specific extensions for call and feature control mechanisms that provide authors advanced mobile network services. Using WTA, applications developers can use the microbrowser to originate telephone calls and to respond to events from the telephone network.

Wireless Session Protocol

WSP provides applications with an interface for two session services. The connection-oriented session service operates above the reliable transport protocol WTP, and the connectionless session service operates above the unreliable transport protocol WDP. In essence, WSP is based on HTTP with some additions and modifications to optimize its use over wireless channels. The principal limitations addressed are low data rate and susceptibility to loss of connection due to poor coverage or cell overloading.

WSP is a transaction-oriented protocol based on the concept of a request and a reply. Each WSP protocol data unit (PDU) consists of a body, which may contain WML, WMLScript, or images, and a header, which contains information about the data in the body and about the transaction. WSP also defines a server Push operation, in which the server sends unrequested content to a client device. This may be used for broadcast messages or for services, such as news headlines or stock quotes, that may be tailored to each client device.

WSP Service In general, a connection-mode WSP provides the following services:

- Establish a reliable session from client to server and release that session in an orderly manner.
- Agree on a common level of protocol functionality using capability negotiation.
- Exchange content between client and server using compact encoding.
- Suspend and resume a session.
- Push content from server to client in an unsynchronized manner.

At the service level, WSP is defined in terms of a collection of service primitives, with associated parameters. These service primitives define the interface between WSP and users of WSP in the WAE.[1] At the protocol level, the WSP specification defines a PDU format used to exchange data between peer WSP entities.

[1] See Appendix 12C for a brief discussion of the concept of service primitives and parameters.

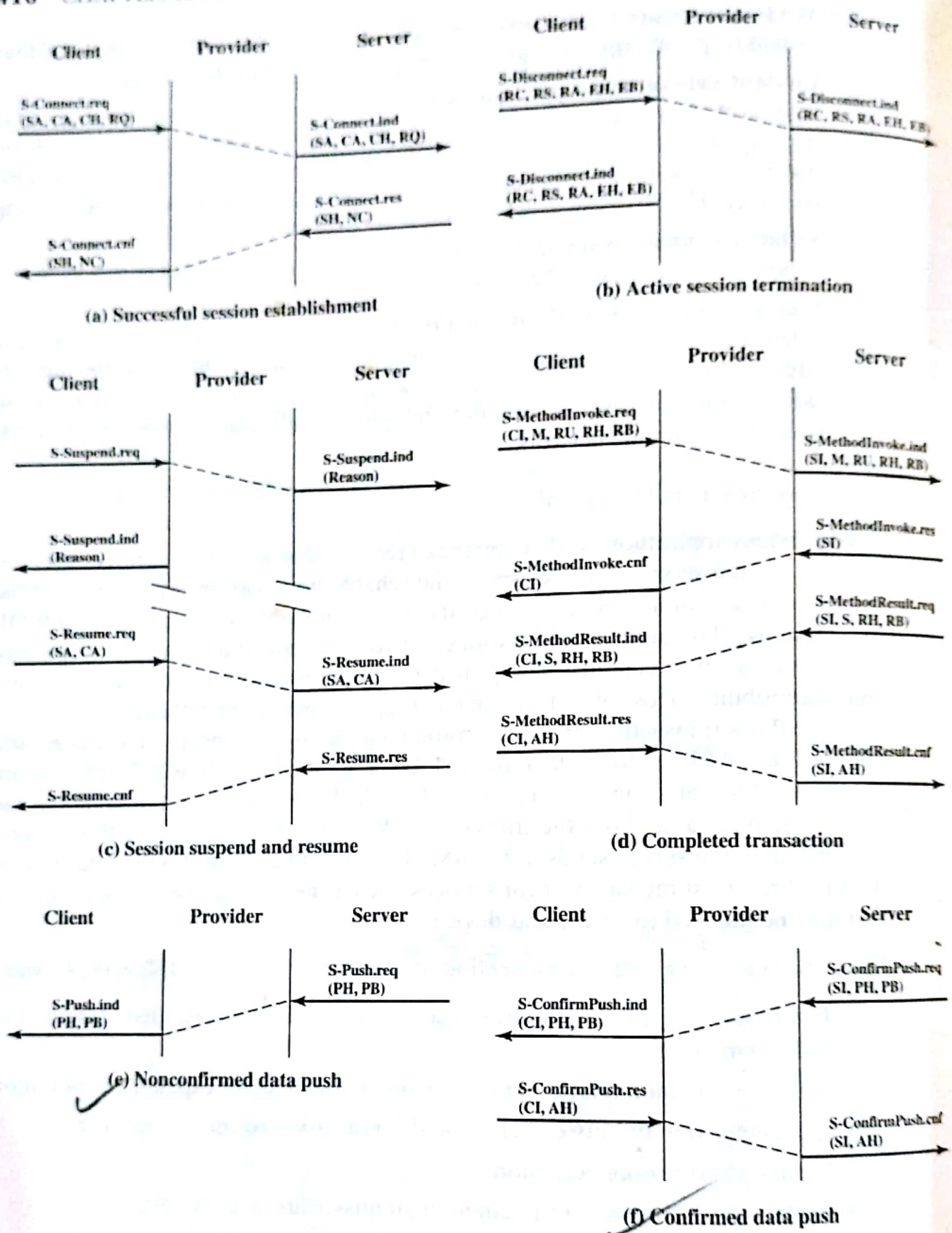

Figure 12.12 Wireless Session Protocol Primitives and Parameters

Figure 12.12 shows the key WSP transaction types in terms of the primitives and parameters that are exchanged. There are other transaction types, but these are sufficient to give a feel for the operation of WSP.

Session establishment involves the exchange of S-Connect primitives (Figure 12.12a). A WSP user acting as a client (mobile node side of the transaction) requests a session with a WSP user acting as a server (Web Server) on a remote system by issuing an S-Connect.req to WSP. Four parameters accompany the request:

- **Server address:** The peer with which the session is to be established.
- **Client address:** The originator of the session.
- **Client headers:** Contain attribute information that can be used for application-level parameters to be communicated to the peer. This information is passed without modification by WSP and is not processed by WSP.
- **Requested capabilities:** A set of capabilities for this session requested by the client; these are listed in Table 12.5.

The client's WSP then prepares a WSP PDU, containing these parameters, to convey the request to the peer WSP at the server. The server address, client address, and client headers are unchanged. However, the WSP service provider at the client or the WSP service provider at the server, or both, may modify the set of requested capabilities so that they do not imply a higher level of functionality than the provider can support. With this possible modification, an S-Connect.ind containing the same parameters as in the request is delivered to the WSP user at the server side. If the WSP user at the server accepts the session request, it responds by invoking WSP with an S-Connect.rsp containing server headers and negotiated capabilities. The negotiated

Table 12.5 Wireless Session Protocol Capabilities

Name	Class	Type	Description
Aliases	I	List of addresses	Indicates which alternative addresses the peer may use to access this session service user. Can be used to facilitate a switch to a news bearer when a session is resumed.
Client SDU size	N	Positive integer	The size of the largest transaction service data unit that may be sent to the client during the session.
Extended methods	N	Set of method names	The set of extended methods (beyond and HTTP/1.1) that are supported by both client server peers.
Header code pages	N	Set of code page names	The set of extension header code pages that are supported by both client and server peers.
Maximum outstanding method requests	N	Positive integer	The maximum number of method invocations that can be active at the same time during the session.
Maximum outstanding push requests	N	Positive integer	The maximum number of confirmed push the invocations that can be active at the same time during the session.
Protocol options	N	Set of facilities and features	May include Push, Confirmed Push, Session Resume, and Acknowledgment Headers.
Server SDU size	N	Positive integer	The size of the largest transaction SDU that may be sent to the server during the session.

transmit to the satellite, and a transmission from a satellite can be received by many stations.

Broadcast Radio

Physical Description The principal difference between broadcast radio and microwave is that the former is omnidirectional and the latter is directional. Thus broadcast radio does not require dish-shaped antennas, and the antennas need not be rigidly mounted to a precise alignment.

Applications *Radio* is a general term used to encompass frequencies in the range of 3 kHz to 300 GHz. We are using the informal term *broadcast radio* to cover the VHF and part of the UHF band: 30 MHz to 1 GHz. This range covers FM radio and UHF and VHF television. This range is also used for a number of data networking applications.

Transmission Characteristics The range 30 MHz to 1 GHz is an effective one for broadcast communications. Unlike the case for lower-frequency electromagnetic waves, the ionosphere is transparent to radio waves above 30 MHz. Thus transmission is limited to the line of sight, and distant transmitters will not interfere with each other due to reflection from the atmosphere. Unlike the higher frequencies of the microwave region, broadcast radio waves are less sensitive to attenuation from rainfall.

As with microwave, the amount of attenuation due to distance for radio obeys Equation (2.2), namely 10 log $(4\pi d/\lambda)^2$ dB. Because of the longer wavelength, radio wavessuffer relatively less attenuation.

A prime source of impairment for broadcast radio waves is multipath interference. Reflection from land, water, and natural or human-made objects can create multiple paths between antennas. This effect is frequently evident when TV reception displays multiple images as an airplane passes by.

Infrared

Infrared communications is achieved using transmitters/receivers (transceivers) that modulate noncoherent infrared light. Transceivers must be within the line of sight of each other either directly or via reflection from a light-colored surface such as the ceiling of a room.

One important difference between infrared and microwave transmission is that the former does not penetrate walls. Thus the security and interference problems encountered in microwave systems are not present. Furthermore, there is no frequency allocation issue with infrared, because no licensing is required.

2.5 MULTIPLEXING

In both local and wide area communications, it is almost always the case that the capacity of the transmission medium exceeds the capacity required for the transmission of a single signal. To make efficient use of the transmission system, it is desirable to carry multiple signals on a single medium. This is referred to as *multiplexing*.