Important applications built on top of RPC are NFS, the Network Filesystem, and NIS, the Network Information System.

Remote Procedure Call provides a different paradigm for accessing network services. Instead of accessing remote services by sending and receiving messages, a client invokes services by making a local procedure call. The local procedure hides the details of the network communication.

Purpose of RPC

Remote Procedure Call (RPC) is a client/server infrastructure that increases the interoperability, portability and flexibility of an application by allowing the application to be distributed over multiple heterogeneous platforms. It reduces the complexity of developing applications that span multiple operating systems and network protocols by insulating the application developer from the details of the various operating system and network interfaces.

Characteristics of RPC

The main characteristics of RPC are listed below:

- No reliability is implemented with RPC and reliability is left to the application.
- RPC does not rely on a specific transport protocol.
- RPC can run on any operating system.
- Fields for client and server identification and authorization are provided.

The main goal of RPC is to hide the existence of the network from a program. As a result, RPC doesn't quite fit into the OSI model:

1. The message-passing nature of network communication is hidden from the user. The user doesn't first open a connection, read and write data and then close the connection. Indeed, a client often does not even know they are using the network.
2. RPC often omits many of the protocol layers to improve performance. Even a small performance improvement is important because a program may invoke RPCs often.

RPC is especially well suited for client-server (e.g., query-response) interaction in which the flow of control alternates between the caller and callee. Conceptually, the client and server do not both execute at the same time. Instead, the thread of execution jumps from the caller to the callee and then back again. When making a remote procedure call:

1. The calling environment is suspended, procedure parameters are transferred across the network to the environment where the procedure is to execute and the procedure is executed there.
2. When the procedure finishes and produces its results, its results are transferred back to the calling environment, where execution resumes as if returning from a regular procedure call.

How RPC Works ?

An RPC is analogous to a function call. Like a function call, when an RPC is made, the calling arguments are passed to the remote procedure and the caller waits for a response to be returned from the remote procedure. The client makes a procedure call that sends a request to the server and waits. The thread is blocked from processing until either a reply is received, or it times out. When the request arrives, the server calls a dispatch routine that performs the requested service and sends the reply to the client. After the RPC call is completed, the client program continues. RPC specifically supports network applications.

A remote procedure is uniquely identified by the triple: (program number, version number, procedure number). The program number identifies a group of related remote procedures, each of which has a unique procedure number. A program may consist of one or more versions. Each version consists of a collection of procedures that are available to be called remotely. Version numbers enable multiple versions of an RPC protocol to be available simultaneously. Each version contains a number of procedures that can be called remotely. Each procedure has a procedure number.

The following steps take place during an RPC:

1. A client invokes a client stub procedure, passing parameters in the usual way. The client stub resides within the client's own address space.
2. The client stub marshalls the parameters into a message. Marshalling includes converting the representation of the parameters into a standard format and copying each parameter into the message.
3. The client stub passes the message to the transport layer, which sends it to the remote server machine.
4. On the server, the transport layer passes the message to a server stub, which demarshalls the parameters and calls the desired server routine using the regular procedure call mechanism.
5. When the server procedure completes, it returns to the server stub (e.g., via a normal procedure call return), which marshalls the return values into a message. The server stub then hands the message to the transport layer.
6. The transport layer sends the result message back to the client transport layer, which hands the message back to the client stub.
7. The client stub demarshalls the return parameters and execution returns to the caller.

The Remote Procedure Call (RPC) message protocol consists of two distinct structures: the call message and the reply message. The message flows are displayed as follows:

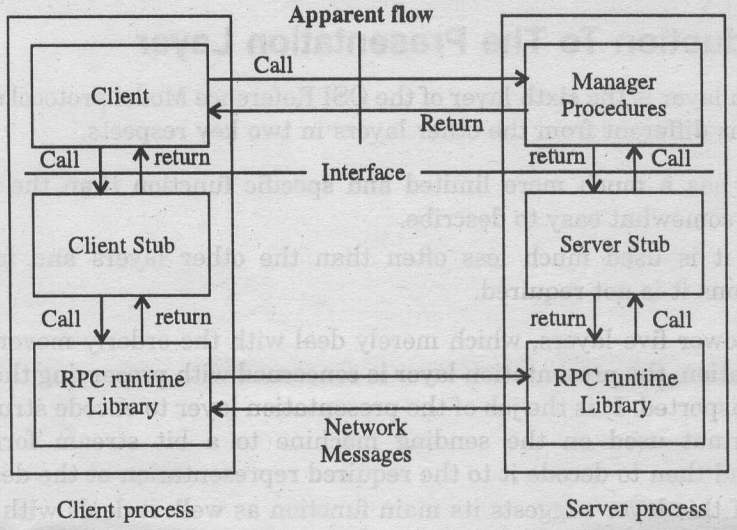


Fig. 15.3 Remote procedure call flow

RPC Issues

Some of the issues that must be addressed by RPC are listed below:

- **Marshalling :** Parameters must be marshalled into a standard representation. Parameters consist of simple types (e.g., integers) and compound types (e.g., C structures or Pascal records). Moreover, because each type has its own representation, the types of the various parameters must be known to the modules that actually do the conversion. For example, 4 bytes of characters would be uninterpreted, while a 4-byte integer may need to the order of its bytes reversed.
- **Semantics :** Call-by-reference not possible: the client and server don't share an address space. That is, addresses referenced by the server correspond to data residing in the client's address space. One approach is to simulate call-by-reference using copy-restore. In copy-restore, call-by-reference parameters are handled by sending a copy of the referenced data structure to the server, and on return replacing the client's copy with that modified by the server.
  However, copy-restore doesn't work in all cases. For instance, if the same argument is passed twice, two copies will be made and references through one parameter only changes one of the copies.
- **Binding :** How does the client know who to call, and where the service resides? The most flexible solution is to use dynamic binding and find the server at run time when the RPC is first made. The first time the client stub is invoked, it contacts a name server to determine the transport address at which the server resides.

15.3 Introduction To The Presentation Layer

The presentation layer is the sixth layer of the OSI Reference Model protocol stack, and second from the top. It is different from the other layers in two key respects.

- First, it has a much more limited and specific function than the other layers; it's actually somewhat easy to describe.
- Second, it is used much less often than the other layers and in many types of connections it is not required.

Unlike the lower five layers, which merely deal with the orderly movement of bits from source to destination, the presentation layer is concerned with preserving the 'meaning' of the information transported. It is the job of the presentation layer to encode structured data from the internal format used on the sending machine to a bit stream format suitable for transmission, and then to decode it to the required representation at the destination.

The name of this layer suggests its main function as well: it deals with the presentation of data. More specifically, the presentation layer is charged with taking care of any issues that might arise where data sent from one system needs to be viewed in a different way by the other system. It also takes care of any special processing that must be done to data from the time an application tries to send it until the time it is sent over the network.

The name "Presentation layer" has caused considerable confusion in the industry because some people mistakenly believe that this layer presents data to the user. However, the name has nothing to do with displaying data. Instead, this function is performed by applications running above the Application layer. The Presentation layer is so named because it presents a uniform data format to the Application layer. As a matter of fact, this layer is not commonly implemented because applications typically perform most Presentation layer functions.

The presentation layer serves as the data translator for the network. This layer on the sending computer translates the data sent by the application layer into a common format. At the receiving computer, the presentation layer translates the common format to a format known to the application layer.

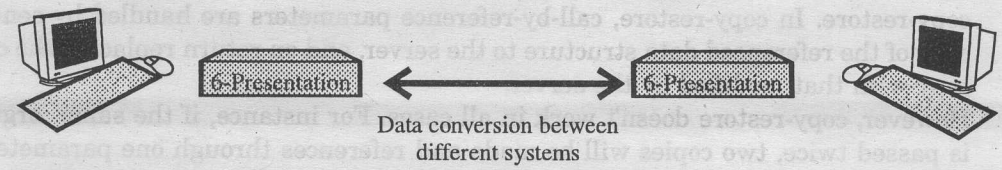


Fig. 15.4 Layer 6-Presentation Layer

15.4 Design Issues of Presentation Layer

The presentation layer is responsible for presenting data in a form that the receiving device can understand. The presentation layer serves as the translator for devices that need to communicate over a network. Layer 6 provides three main functions that are explained below:

- **Translation :** Networks can connect very different types of computers together : PCs, Macintoshes, UNIX systems, AS/400 servers and mainframes can all exist on the same network. These systems have many distinct characteristics and represent data in different ways; they may use different character sets for example. The presentation layer handles the job of hiding these differences between machines.
- **Compression :** Compression (and decompression) may be done at the presentation layer to improve the throughput of data. (More explanation in next section)
- **Encryption :** Some types of encryption (and decryption) are performed at the presentation layer. This ensures the security of the data as it travels down the protocol stack. For example, one of the most popular encryption schemes that is usually associated with the presentation layer is the Secure Sockets Layer (SSL) protocol. Not all encryption is done at layer 6, however; some encryption is often done at lower layers in the protocol stack.

After receiving data from the application layer, the presentation layer performs one, or all, of its functions on the data before it sends it to the session layer. At the receiving station, the presentation layer takes the data from the session layer and performs the required functions before passing it to the application layer.

On the receiving end, the Presentation layer converts the machine-independent data from the network into the format required for the local system. This conversion could include the following:

- **Bit-order translation :** When binary numbers are transmitted through a network, they are sent one bit at a time. The transmitting computer can start at either end of the number. Some computers start at the most-significant digit (MSD); others start at the least-significant digit (LSD).
- **Byte-order translation :** Complex values generally must be represented with more than one byte, but different computers use different conventions to determine which byte should be transmitted first. Intel microprocessors, for example, start with the least-significant byte and are called *little endian*. Motorola microprocessors, on the other hand, start with the most-significant byte and are called *big endian*. Byte-order translation might be needed to reconcile these differences when transferring data between a PC and a Macintosh.
- **Character code translation :** Different computers use different binary schemes for representing character sets. For instance: *ASCII*, the American Standard Code for Information Interchange, is used to represent English characters on all microcomputers and most minicomputers; *EBCDIC*, the Extended Binary Coded Decimal Interchange Code, is used to represent English characters on IBM mainframes; and *Shift-JIS* is used to represent Japanese characters.
- **File Syntax Translation :** File formats differ between computers. For instance, Macintosh files actually consist of two related files called a data fork and a resource fork. PC files, on the other hand, consist of a single file.

Presentation Layer Role in the OSI Model

The reason that the presentation layer is not always used in network communications is that the jobs mentioned above are simply not always needed. Compression and encryption are usually considered "optional" and translation features are also only needed in certain circumstances. Another reason why the presentation layer is sometimes not mentioned is that its functions may be performed as part of the application layer.

The fact that the translation job done by the presentation layer isn't always needed means that it is common for it to be "skipped" by actual protocol stack implementations. This means that protocols at layer seven may talk directly with those at layer five.

Now let us discuss the main functions of presentation layer in detail in next sections.

15.4.1 Data Formatting

To understand how data formatting works, imagine two dissimilar systems. The first system uses Extended Binary Coded Decimal Interchange Code (EBCDIC) to represent characters onscreen. The second system uses American Standard Code for Information Interchange (ASCII) for the same function. Layer 6 provides the translation between these two different types of codes.

Layer 6 standards also determine how graphic images are presented. Three of these standards are as follows:

- **PICT**—is a picture format used to transfer QuickDraw graphics between programs on the MAC operating system.
- **TIFF** (Tagged Image File Format)—a format for high-resolution, bit-mapped images.
- **JPEG** (Joint Photographic Experts Group)—graphic format used most often to compress still images of complex pictures and photographs.

Other Layer 6 standards guide the presentation of sound and movies. Included in these standards are the following:

- **MIDI** (Musical Instrument Digital Interface)—for digitized music.
- **MPEG** (Motion Picture Experts Group) is a standard for the compression and coding of motion video for CDs and digital storage.
- QuickTime is a standard that handles audio and video for programs on both MAC and PC operating system.

Conversions standards defined on the Presentation Layer for data conversion and formatting:

Category	Standards
Data Conversion	ASCII, EBCDIC, encryption
Audio/video conversion	MIDI, MPEG, QuickTime, AVI
Graphics conversion	GIF, JPEG, PICT, TIFF

15.4.2 Data Encryption

Encryption is scrambling the data so that only authorized participants can unscramble the messages of a conversation. Recall, that it's easy to "wiretap" transmission media such as Ethernets.

Layer 6 is also responsible for data encryption. Data encryption protects information during its transmission. Financial transactions (e.g., credit card information) use encryption to protect sensitive information as it traverses the Internet. An encryption key is used to encrypt the data at its source and then to decrypt the data at its destination.

The location of encryption in the OSI model has been so controversial that all mention of the subject was omitted from the initial standard. In theory, encryption can be done in any layer, but in practice three layers seem the most suitable: physical, transport, and presentation.

When encryption is done on the physical layer, an encryption unit is inserted between each computer and the physical medium. Every bit leaving the computer is encrypted and every bit entering a computer is decrypted. This scheme is called link encryption. It is simple, but relatively inflexible.

When encryption is done in the transport layer, the entire session is encrypted. A more sophisticated approach is to put it in the presentation layer, so that only those data structures or fields requiring encryption must suffer the overhead of it.

The messages to be encrypted, known as **plaintext** are transformed by a function that is parameterized by a key. The output of this is known as **ciphertext.** There are three main ways to achieve this in increasing complexity.

- **Substitution ciphers** use a direct mapping function and are open to cracking. For example if we are dealing with straight alphabetical characters we might employ the following substitution.
  - plaintext : abcdefghijklmnopqrstuvwxyz.
  - ciphertext : QWERTYUIOPASDFDFGHJKLZXCVBNM.

This general system is called a monoalphabetical substitution with the key being the 26-letter string corresponding to the full alphabet.

This type of encipherment is easily attacked by using the statistical properties of natural languages. i.e., we might note that *e* is the most common letter and decipher the ciphertext on this statistical basis.

- **Transposition ciphers** reorder the letters but do not disguise them. You use a key to reorder the original plaintext. However like a substitution cipher it is open to attack.
- The **Data Encryption Standard** is recognized as an industry standard currently. It is based on both substitution and transposition techniques. Operating on a 64-bit stream of plaintext a series of substitutions and permutations are performed using a 56-bit key. To make it even more effective the key may be changed over time by EX-ORing with the plaintext at either end. This leads to the key being dependent on the history of all previous transmissions of that session.

Despite all its complexity, DES is basically a monoalphabetic substitution cipher using a 64-bit alphabet. (More about encryption in chapter 17)

15.4.3 Data Compression

The presentation layer is also responsible for the compression of files. Data compression squeezes data so it requires less disk space for storage and less bandwidth on a data transmission channel. Communications equipment like modems, bridges and routers use compression schemes to improve throughput over standard phone lines or leased lines. Compression is also used to compress voice telephone calls transmitted over leased lines so that more calls can be placed on those lines. In addition, compression is essential for videoconferencing applications that run over data networks. Compression works by using algorithms (complex mathematical formulas) to shrink the size of the files. The algorithm searches each file for repeating bit patterns and then replaces them with a token. A token is a much shorter bit pattern that represents the long pattern.

The aim of data compression is to reduce redundancy in stored or communicated data, thus increasing effective data density. Data compression has important application in the areas of file storage and distributed systems. Most compression schemes take advantage of the fact that data contains a lot of repetition. For example, alphanumeric characters are normally represented by a 7-bit ASCII code, but a compression scheme can use a 3-bit code to represent the eight most common letters.

In addition, long stretches of "nothing" can be replaced by a value that indicates how much "nothing" there is. For example, silence in a compressed audio recording can be replaced by a value that indicates how long that silence is. White space in a compressed graphic image can be replaced by a value that indicates the amount of white space.

A simple characterization of data compression is that it involves transforming a string of characters in some representation (such as ASCII) into a new string (of bits, for example) which contains the same information but whose length is as small as possible. Data compression has important application in the areas of data transmission and data storage. Many data processing applications require storage of large volumes of data and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the proliferation of computer communication networks is resulting in massive transfer of data over communication links.

Compressing data to be stored or transmitted reduces storage and/or communication costs. When the amount of data to be transmitted is reduced, the effect is that of increasing the capacity of the communication channel. Similarly, compressing a file to half of its original size is equivalent to doubling the capacity of the storage medium. It may then become feasible to store the data at a higher, thus faster, level of the storage hierarchy and reduce the load on the input/output channels of the computer system.

Two important compression concepts are lossy and lossless compression:

- **Lossy compression :** With lossy compression, it is assumed that some loss of information is acceptable. The best example is a videoconference where there is an acceptable amount of frame loss in order to deliver the image in real time. People may

appear jerky in their movements, but you still have a grasp for what is happening on the other end of the conference. In the case of graphics files, some resolution may be lost in order to create a smaller file. The loss may be in the form of color depth or graphic detail. For example, high-resolution details can be lost if a picture is going to be displayed on a low-resolution device. Loss is also acceptable in voice and audio compression, depending on the desired quality. The techniques that we will discuss under this category are listed below:

- JPEG.
- MPEG.

- **Lossless compression :** With lossless compression, data is compressed without any loss of data. It assumes you want to get everything back that you put in. Critical financial data files are examples where lossless compression is required. The techniques that we will discuss under this category are listed below:

  - Run-Length Encoding
  - Huffman Encoding
  - Lempel-Ziv-Welch Encoding

The removal of information in the lossy technique is acceptable for images, because the loss of information is usually imperceptible to the human eye. While this trick works on humans, you may not be able to use lossy images in some situations, such as when scanners are used to locate details in images.

Lossy compression can provide compression ratios of 100 : 1 to 200 : 1, depending on the type of information being compressed. Lossless compression ratios usually only achieve a 2 : 1 compression ratio. Lossy compression techniques are often "tunable" in that you can turn the compression up to improve throughput, but at a loss in quality. Compression can also be turned downed to the point at which there is little loss of image, but throughput will be affected.

15.4.3.1 Basic compression techniques

Data compression is often referred to as coding, where coding is a very general term encompassing any special representation of data that satisfies a given need. Information theory is defined to be the study of efficient coding and its consequences, in the form of speed of transmission and probability of error. Data compression may be viewed as a branch of information theory in which the primary objective is to minimize the amount of data to be transmitted.

Because compression algorithms are software-based, overhead exists that can cause problems in real-time environments. Compression is processor intensive, so for real-time data transmissions like network links, you will need a system on both ends of the link that can compress and decompress data without causing appreciable delays. Some applications may not tolerate any delays, in which case you may need to tune the compression levels and/or boost the processing power to remove those delays.

Another consideration is that compression affects the portability of files. You will need to make sure that all recipients have the software needed to decompress files. Note that some

files are already compressed to begin with and don't benefit from any further external compression techniques. Some graphics file formats, such as TIFF (Tagged Image File Format), are automatically compressed when they are stored.

File compression is handled in several ways. Various utilities are available that let you compress files one at a time or as a group. Groups of files can be compressed into a single file that is much easier to send to another user. A decompression utility unpacks the files. Nearly universal file compression utilities are PKZip and WINZip from Niko Mak Computing. A file or a group of files can be compressed into a zip file (called an archive if multiple files are compressed into it) that can be decompressed by anybody else that has the utility, or you can create self-extracting files that can be opened by anyone who does not have the utility. Most operating systems, including DOS, NetWare, Windows NT, and others, now include disk compression software. Some file encryption utilities also include compression routines so you can make files private and compress files in the same steps.

There are three general approaches to data compression that are listed below:

- Finite Set of Symbols.
- Huffman Encoding.
- Context Dependent Encoding.

Each approach assumes that the data stream can be transformed into a more compact representation, which the receiver reconstructs back into the original data.

1. Finite Set of Symbols

Consider a library with many branch offices in which the previous days transactions are sent to every other branch after closing. Transactions consist of checked out and returned books. We could exchange information in the following ways:

1. We could send the name of the book, its author, the copy number etc. together with the type of transaction.
2. Alternatively, the library could maintain a sitewide table assigning a unique ID number to every book in every branch. Transactions could then refer to the book's ID number, rather than its title. Because book IDs are small (e.g., a few bytes), less data will be transmitted.

The above technique is used throughout programming. We frequently exchange pointers and array subscripts to avoid the cost of transferring large amounts of data between subroutines.

The previous approach assumes that all objects occur with equal frequency and that the set of objects (e.g., books) is finite. If we examine text, however, we immediately notice that some words appear more often than others. We could reduce the number of bits needed to represent a document by using a coding scheme that employs small code words to represent common words and longer code words to represent words the appear infrequently.

2. Adaptive Huffman Coding and Lempel-Ziv-Welch Algorithms

These compression techniques use a symbol dictionary to represent recurring patterns. The dictionary is dynamically updated during compression as new patterns occur. For data transmissions, the dictionary is passed to a receiving system so it knows how to decode the characters. For file storage, the dictionary is stored with the compressed file.

Huffman encoding is a technique used to encode symbols according to the frequency of their use. The algorithm is as follows:

1. Create a set of nodes, one node per symbol, with a node's value given by the probability of its occurrence in the data.
2. Find the two nodes having the smallest value, remove them from the set and create a new node having the two removed nodes as children and assign the new node a value that is the sum of its children's values. Add the new node back to the set of nodes.
3. Repeat step 2 until only one node remains. We now have a tree, whose probability value is one.
4. The encoding for each symbol is the path from the root to the symbol. Using a code of 0 for a left child, 1 for a right child, the length of each symbol's encoding is proportional to the relative probability of its occurrence.

One drawback with Huffman encoding, however, is that symbols have differing lengths, making it relatively expensive (computationally) to decode. Also, A single-bit error can wipe out the entire message.

In the **Lempel-Ziv data-compression algorithm**, all single character strings occupy the table. As new strings appear, a tree structure is created, similar to Figure 15.5, which shows the "*T*" branch of the tree. Note that a three-character word can be deduced by following any branch of the tree. Each branch of the tree is identified by a codeword and the codeword is sent in any transmissions. If a new string appears, nodes are added to an appropriate branch of the tree and a new codeword is generated to represent it.

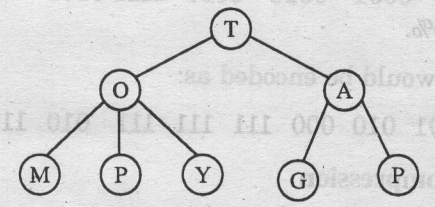


Fig. 15.5 Lempel-Ziv data compression

3. Context Dependent Encoding

The last technique, context dependent encoding, recognizes that the probability of a particular symbol occurring next depends on the previous symbol. For instance, the probability that a "T" directly follows a "*Q*" is about 4 times less than the probability of a "*U*" following a "*Q*".

The main disadvantage of conditional probability methods is the increase in table space.

Each symbol has its own table that gives the codes for those symbols immediately following it. For K symbols, the tables will contain K^2 entries. All K symbols have K entries for the symbols that follow them.

One variation to this technique is as follows:

1. Use 5-bit code to represent symbols.
2. Have four different "modes", where the current mode determines the meaning of symbols. Four symbols are reserved to denote a switch to another mode. For instance, different modes could represent upper-case characters, lower-case characters, numeric and special characters, and control characters.
3. The underlying assumption is that lower-case letters are likely to follow lower-case letters, numbers are likely to occur in groups, etc.
4. The occurrence of a mode symbol signifies a mode change.
5. Our 5-bit code now represents $4 \times 28 = 112$ different values.

One advantage that the above technique has over Huffman encoding is that symbols are all fixed length, making encoding and decoding using table lookups very efficient. In addition, it is more immune to transmission errors.

Run Length Encoding

Another alternative, run length encoding, is used to encode data containing repetitive symbols. Consider binary strings of 0s and 1s. One way to handle long runs of 0 is to use a k-bit symbol that tells how many 0 bits occurred between consecutive 1's. A code word of all 1's means that the true distance is $2^K - 1$ plus the value of the following symbol(s). For example:

000100001010011000000000000000000000000100000001 (47-bits)
consists of runs of length 3, 4, 1, 2, 0, 23 and 7.

Using 4-bit symbols, it would be encoded as:

0011 0100 0001 0010 0000 1111 0100 0111, for 32-bits

and a savings of 15/47 = 30%.

Using 3-bit symbols, it would be encoded as:

011 100 001 010 000 111 111 111 010 111 000 for 33-bits.

15.4.3.2 Graphics/video compression

There are number of compression techniques designed to store graphic images in a much smaller space. Image compression also improves throughput when images files are transmitted from one place to another. Several compression techniques are discussed below:

- **DCT (discrete cosine transform) :** DCT is a common compression technique in which data is represented as a series of cosine waves. In the case of video, this technique replaces continuous sampling with an equation that represents the data. In the case of a still image, 8×8 blocks of information are converted into a wave that describes the number of color shifts and the extent of change in those color shifts.

- **Fractal compression :** The basic idea is to break an image down into smaller and smaller tiles. The compression engine (a dedicated board) searches for matching patterns in the image using a mathematical transformation that manipulates tiles in various ways. Repetitive patterns are saved to reconstruct the original, and unmatched data that is considered unimportant is discarded.
- **Wavelet transform :** When using a wavelet transform to describe an image, an average of the coefficients-in this case, pixels-is taken. Then the detail coefficients are calculated. Another average is taken, and more detail coefficients are calculated. This process continues until the image is completely described or the level of detail necessary to represent the image is achieved. As more detail coefficients are described, the image becomes clearer and less blocky. Once the wavelet transform is complete, a picture can be displayed at any resolution by recursively adding and subtracting the detail coefficients from a lower-resolution version. This technique is used by Iterated Systems.

JPEG (Joint Photographic Experts Group) Compression

It is an ITU and ISO standardized method for compressing still images using DCT, which converts three-dimensional color and coordinate image information into a format that is more responsive to compression. Color information is also encoded and some is discarded, depending on the desired end-results resolution. Compression can be lossless or lossy. The resulting information is then compressed using RLE, a special technique that compresses similar regions.

MPEG (Motion Picture Experts Group)

MPEG is developing several video compression standards that define formatting, data rates and compression techniques for international use. MPEG uses compression methods like DCT. Note that JPEG and MPEG are not related, although they are developed by the same organization. JPEG is for still images while MPEG is specifically designed for motion video.

• • •

16

Application Layer

Introduction

In the Open Systems Interconnection (OSI) communications model, the Application layer provides services for an application program to ensure that effective communication with another application program in a network is possible. This layer provides the interface to the communications environment, which is used by the application process. It is responsible for communicating application process parameters.

The application layer contains a variety of protocols that are commonly needed. For example, there are hundreds of incompatible terminal types in the world. Consider the plight of a full screen editor that is supposed to work over a network with many different terminal types, each with different screen layouts, escape sequences for inserting and deleting text, moving the cursor, etc.

One way to solve this problem is to define an abstract network virtual terminal for which editors and other programs can be written to deal with. To handle each terminal type, a piece of software must be written to map the functions of the network virtual terminal onto the real terminal. For example, when the editor moves the virtual terminal's cursor to the upper left-hand corner of the screen, this software must issue the proper command sequence to the real terminal to get its cursor there too. All the virtual terminal software is in the application layer.

Another application layer function is file transfer. Different file systems have different file naming conventions, different ways of representing text lines, and so on. Transferring a file between two different systems requires handling these and other incompatibilities. This work, too, belongs to the application layer, as do electronic mail, remote job entry, directory lookup, and various other general-purpose and special-purpose facilities.

The Applications distributed over networks share common needs. They are:

1. Establish a context for communication between separated processes.
2. Execute instructions on a remote system.
3. Reliable transfer of larger data volumes.
4. Coordinate components of a distributed application to avoid inconsistencies of state in shared data.

The Application layer is NOT the application itself that is doing the communication. It is a service layer that provides these services:

- Information transfer.
- Identification of intended communication partner, by name, address or some other description.
- Establishment of the authority of the application process to use the communication services.
- Determination of the availability of intended communication partner.
- Agreement of privacy mechanisms required for communication.
- Authentication of intended communication partners.
- Resource costing.
- Determination of the adequacy of the resources available for intended communication.
- Determination of service quality.
- Synchronization between applications.
- Selection of dialogue discipline. Including initiation and release procedures.
- Agreement on who has responsibility for error recovery.
- Agreement on procedures ensuring data integrity.

It may be convenient to think of the Application layer as the high-level set-up services for the application program or an interactive user.

Application Requirements

- **Bandwidth** : How many bits/sec required? Smooth or bursty traffic stream?
- **Holding time** : How long does application run?
- **Data-level reliability** : Reliable (in-order, no loss) delivery needed?
- **Performance** : Constraints on maximum application-to-application delay, tail of delay distribution?
- **QoS** : Quality of service guarantees required?
- **Communication structure** : 1-1, 1-many, many-many?
- **Security** : Authentication, encryption required?

Internet applications

Concerning the Internet application layer, it consists of a set of protocols providing different services. Among these, the most popular are:

- SMTP, IMAP, POP3 for electronic mail applications;
- HTTP, FTP for data transfer;
- TELNET for terminal emulation;
- RTP, RTCP, SIP for multimedia applications.

Now we will discuss all these protocols in detail.

16.1 FTP : File Transfer Protocol

FTP (File Transfer Protocol) is the simplest and most secure way to exchange files over the Internet. The most common use for FTP is to download files from the Internet. In addition, the ability to transfer files back-and-forth makes FTP almost essential for anyone creating a Web page, amateurs and professionals alike.

When downloading a file from the Internet you're actually transferring the file to your computer from another computer over the Internet. This is why the T (transfer) is in FTP. You may not know where the computer is that the file is coming from but you most likely know it's URL or Internet address.

Most often, a computer with the FTP address is dedicated to receive FTP connection. Just as a computer that is setup to host Web pages is referred to as a Web server or Website, a computer dedicated to receiving an FTP connection is referred to as an FTP server or FTP site. An FTP site is like a large filing cabinet. With a traditional filing cabinet, the person who does the filing has the option to label and organize the files however they see fit. They also decide which files to keep locked and which remain public. It is the same with an FTP site.

FTP Basics

FTP exists primarily for the transfer of data between two end points. The two objectives of the protocol are to:

- Promote the sharing of files.
- Transfer data reliably and efficiently.

FTP differs from HTTP fundamentally as it is an application made up of two distinct TCP connections:

- **Control connection :** This TCP-based connection is used to provide a communications channel for the delivery of commands and replies. This is effectively the mechanism that enables the user to tell the server which file is being requested, which directory it is in, and so forth.
- **Data connection :** The second TCP-based connection is used for the actual transfer of user data. Once the Control connection has been used to exchange information on which file is required, the Data connection is used to transfer the file between the client and server.

Using these two communication connections, two distinct modes of operation determine in which direction the connections are established: Active mode and Passive mode.

How FTP Works in Active Mode ?

FTP creates both a control and a data connection in order to transfer files. Within an Active FTP session, the Control connection is established from the client to the server, with the Data connection established back from the server to the client.

The control connection is based on telnet and is used to negotiate the parameters for the data transfer. This is called an active FTP connection.

1. The client FTP application opens a control connection to the server on destination port 21, and specifies a source port as the source to which the FTP server should respond (using TCP).
2. The FTP server responds on port 21.
3. The FTP server and client negotiate the data transfer parameters.
4. The FTP server opens a second connection for data on port 20 to the original client.
5. The client responds on the data port, completing a TCP connection.
6. Data transfer begins.
7. The server indicates the end of the data transfer
8. Client closes the connection once the data is received.
9. The data connection is closed.
10. The FTP connection is closed.

How FTP Works in Passive Mode ?

Passive mode FTP works similarly to Active mode FTP with one major exception: both the Control and Data connections within a Passive mode FTP session are established from the client to the server. The passive mode behavior of an FTP server:

1. The client opens a connection to the server on TCP port 21 (command channel).
2. The server accepts the connection.
3. The server initiates a connection to the client using port 20 as the source port (for the data channel).
4. The client accepts the connection and acknowledges all data transfers on port 20.

The objectives of FTP are :

1. To promote sharing of files (computer programs and/or data).
2. To encourage indirect or implicit use of remote computers.
3. To shield a user from variations in file storage systems among different hosts.
4. To transfer data reliably and efficiently.

Disadvantages are :

1. Passwords and file contents are sent in clear text, which can be intercepted by eavesdroppers.
2. Multiple TCP/IP connections are used, one for the control connection, and one for each download, upload or directory listing. Firewall software needs additional logic to account for these connections.
3. It is hard to filter active mode FTP traffic on the client side by using a firewall, since the client must open an arbitrary port in order to receive the connection. This problem is largely resolved by using passive mode FTP.
4. It is possible to abuse the protocol's built-in proxy features to tell a server to send data to an arbitrary port of a third computer.

16.2 HTTP : Hyper Text Transfer Protocol

The Hyper Text Transfer Protocol, or HTTP, must be the most widely used Application layer protocol in the world today. It forms the basis of what most people understand the Internet to be-the World Wide Web. Its purpose is to provide a lightweight protocol for the retrieval of Hyper Text Markup Language (HTML) and other documents from Web sites throughout the Internet. Each time you open a Web browser to surf the Internet, you are using HTTP over TCP/IP.

It's the network protocol used to deliver virtually all files and other data (collectively called resources) on the World Wide Web, whether they're HTML files, image files, query results, or anything else. Usually, HTTP takes place through TCP/IP sockets.

A browser is an HTTP client because it sends requests to an HTTP server (Web server), which then sends responses back to the client. The standard (and default) port for HTTP servers to listen on is 80, though they can use any port.

HTTP is used to transmit resources, not just files. A resource is some chunk of information that can be identified by a URL (it's the R in URL). The most common kind of resource is a file, but a resource may also be a dynamically generated query result, the output of a CGI script, a document that is available in several languages, or something else.

Basic HTTP Page Retrieval

In this section we will see how a basic browser retrieves a Web page from a Web server. The first important point to note is that a Web page is typically made up of many dozens of objects, ranging from the HTML base through to the images that are present on the page. The HTML can be thought of as the template for the page overall, instructing the browser on the layout of the text, font sizes and colors, background color of the page, and which other images need to be retrieved to make up the page.

The processes take place in the following order:

1. Client sends a request for the required page to the Web server.
2. The server analyzes the request and sends back an acknowledgment to the client along with the HTML code required to make the page.
3. The client will begin interpreting the HTML and building the page.
4. The client, in subsequent requests, will retrieve any embedded objects, such as images or other multimedia sources.

Once all elements of the page have been retrieved, the client browser will display the completed Web page.

The HTTP URL (Uniform Resource Locator)

The URL is the most important piece of information that the client browser includes. The URL is defined as being a combination of the host where the site is located, the scheme used to retrieve the page, and the full path and filename. Optionally, the URL may include information such as the TCP port number to be used or a unique reference point within a larger page.

Cookies—The HTTP State Management Mechanism

The biggest challenges in HTTP environments, whether content switched or not, is maintaining some form of client-side state that enables Web servers and intermediary devices to recognize the client session and understand the current status of the user session. In HTTP, cookies take the form of a small piece of text information that is implanted into the user's browser either permanently or temporarily. The term **cookie** is commonly used in computing to describe an opaque piece of information held during a session and, unfortunately, seems to have no more interesting origin than that. Once the backend server has implanted the cookie into the user's browser, the information can be used for a number of different applications ranging from content personalization, user session persistence for online shopping, and the collection of demographic and statistical information on Web site usage.

16.3 DNS : Domain Name System

Domain naming, and its most visible component, the Domain Name System (DNS), is critical to the operation of the Internet. The average phone number, with area code, is 10 digits in length and encodes 10^10, or 10,000,000,000 possibilities. The Internet IP address, at 32 bits, encodes 2^32 or 4,294,967,296 possibilities. For human engineering purposes, how can we build an effective directory of these difficult large numbers?

The telephone company solves this problem with lots of large paper directories, and operators you call and ask about numbers not in your directory. Internet solves this problem with a hierarchy of simple, mnemonic names, called domain names. Instead of remembering 205.216.138.22, all I need to know is the host's domain name—quick.com. Some people think the dots in a domain name correspond to the dots in the numeric address. This is not the case.

There are always three periods in an IP address, separating its four constituent bytes. There are a variable number of periods in a domain name.

Domain Name Service is the service used to convert human readable names of hosts to IP addresses. Host names are not case sensitive and can contain alphabetic or numeric letters or the hyphen. Avoid the underscore. A fully qualified domain name (FQDN) consists of the host name plus domain name as in the following example:

computername.domain.com

Three main components of DNS are listed below:

1. Resolver
2. Name server
3. Database of resource records (RRs)

The part of the system sending the queries is called the **resolver** and is the client side of the configuration. The **name server** answers the queries. The main function of DNS is the mapping of IP addresses to human readable names.

The Domain Name System (DNS) is basically a large database that resides on various computers and it contains the names and IP addresses of various hosts on the Internet and various domains. The Domain Name System is used to provide information to the Domain Name Service to use when queries are made. The service is the act of querying the database, and the system is the data structure and data itself. The Domain Name System is similar to a file system in Unix or DOS starting with a root. Branches attach to the root to create a huge set of paths. Each branch in the DNS is called a label. Each label can be 63 characters long, but most are less. Each text word between the dots can be 63 characters in length, with the total domain name (all the labels) limited to 255 bytes in overall length. The domain name system database is divided into sections called **zones**.

DNS names are assigned through the Internet Registries by the Internet Assigned Number Authority (IANA). The domain name is a name assigned to an Internet domain. For example, mycollege.edu represents the domain name of an educational institution. The names microsoft.com and 3Com.com represent the domain names of those commercial companies. Naming hosts within the domain is up to individuals who administer their domain.

Access to the Domain name database is through a resolver that may be a program or part of an operating system that resides on users workstations. The resolver will send requests to the name servers to return information requested by the user. The requesting computer tries to connect to the name server using its IP address rather than the name.

DNS servers and their databases

For any group of computers partaking of the DNS naming scheme there is likely to be a single definitive list of DNS names and associated IP addresses. The group of computers included in this list is called a **zone**. A zone could be a top-level national domain or a university department. Within a zone DNS service for subsidiary zones may be **delegated** along with a subsidiary domain. The computer that maintains the master list for a zone is said to have **authority** for that zone and will be the primary name server for that zone, there will also be

secondaries for that zone. When any process needs to determine an IP address given a DNS address it calls upon the local host to resolve the address.

Name Servers

A name server is an Internet host running software capable of processing DNS requests. There are three types of name servers:

1. The primary master builds its database from files that were preconfigured on its hosts, called zone or database files. The name server reads these files and builds a database for the zone it is authoritative for.
2. Secondary masters can provide information to resolvers just like the primary masters, but they get their information from the primary. Any updates to the database are provided by the primary.
3. Caching name server—It gets all its answers to queries from other name servers and saves (caches) the answers. It is a non-authoritative server.

The caching name server generates no zone transfer traffic. A DNS Server that can communicate outside of the private network to resolve a DNS name query is referred to as forwarder.

Primary and Secondary Name Servers

Typically, a single name server will be configured as the primary name server for a domain. For backup purposes, a number of other name servers may be configured as secondary name servers. From the standpoint of DNS, there is no difference between primary and secondary name servers, since the resolving algorithm simply uses a domain's NS records in the order provided. Typically, the primary name server is listed first, followed by the secondaries, but this is not a requirement. In fact, if a group of domains is served by a set of name servers, the ordering of the name servers may be mixed among the domains, to facilitate load balancing.

DNS can refer either to the entire system, or to the protocol that makes it work.

Structure and message format

DNS is hierarchical in structure. A domain is a subtree of the domain name space.

- **3 letter codes**

The DNS was originally introduced in the United States of America and the final component of an address was intended to indicate the type of organization hosting the computer. Some of the three letter final labels (edu, gov, mil) are still only used by organizations based in the USA, others can be used anywhere in the world.

The three letter codes are:

Code	Meaning
com	Commercial. Now international.
edu	Educational.
gov	Government.
int	International Organization.
mil	Military.
net	Network related.
org	Miscellaneous Organization.

- **Two letter codes**

The final two letter codes indicate the country of origin, e.g., the two letter code us is used by some sites in the United States of America. In some countries there are sub-domains indicating the type of organization such as ac.uk, co.uk, sch.uk in the United Kingdom and edu.au and com.au in Australia. Most European countries have not adopted this useful practice.

The figure 16.1 shows a partial DNS hierarchy. At the top is what is called the root and it is the start of all other branches in the DNS tree. It is designated with a period. Each branch moves down from level to level. When referring to DNS addresses, they are referred to from the bottom up with the root designator (period) at the far right.

Example : "myhost.mygrp.mycorp.com".

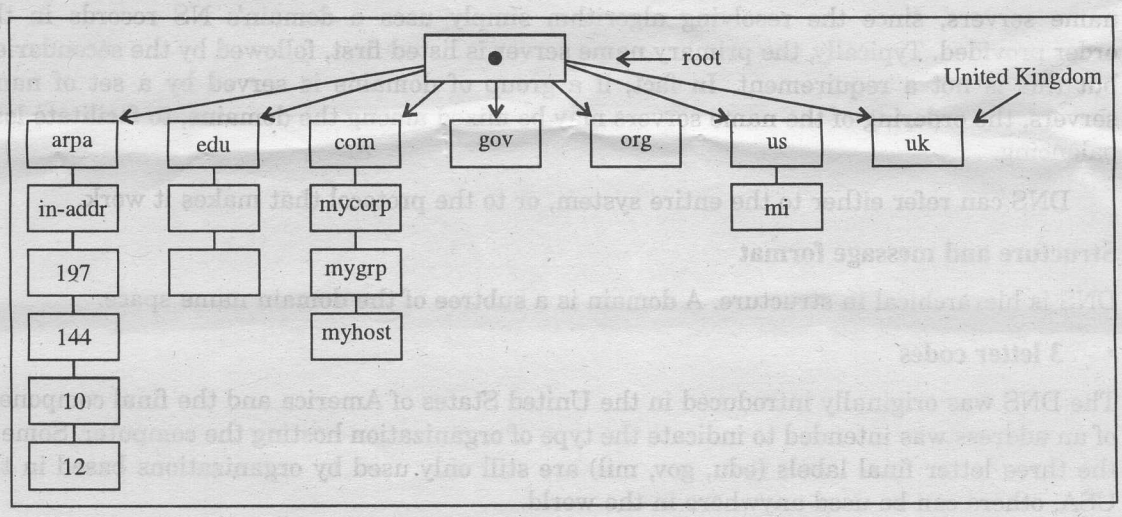


Fig. 16.1 Partial DNS hierarchy

Partial DNS hierarchy

To obtain a domain address it is necessary to identify the administrator of the required

domain and then all that is basically necessary is to send the administrator the required code and the associated IP address and they will, if they accept the request, include the details in their databases.

DNS Query Types

There are following types of queries issued:

1. **Recursive** queries received by a server forces that server to find the information requested or post a message back to the querier that the information cannot be found.
2. **Iterative** queries allow the server to search for the information and pass back the best information it knows about. This is the type that is used between servers. Clients used the recursive query.
3. **Reverse**—The client provides the IP address and asks for the name. In other queries the name is provided, and the IP address is returned to the client.

Generally (but not always), a server-to-server query is iterative and a client-resolver-to-server query is recursive. You should also note that a server can be queried or it can be the person placing a query. Therefore, a server contains both the server and client functions. A server can transmit either type of query. If it is handed a recursive query from a remote source, it must transmit other queries to find the specified name, or send a message back to the originator of the query that the name could not be found.

16.4 RTSP : Real Time Streaming Protocol

In the modern Internet, applications are required to deliver value. Application layer protocols such as RTSP enables the delivery of real-time video and audio in variable qualities. The other Application layer protocols we've looked at so far in this chapter work in a request/response manner, whereby the client asks for some piece of content, the content is delivered using TCP or UDP, and then the client application can display the content to the user. While these mechanisms are suitable for a large number of applications in the Internet, there also exists a requirement to deliver content, be it images, audio, video, or a combination of all three, in real time. Imagine if a user were to try to watch a full-screen video file of a one-hour movie using HTTP or FTP as the Application layer protocol. The movie file might be several hundred megabytes, if not several gigabytes, in size. Even with modern broadband services deliverable to the home, this type of large file size does not fit well in the "download then play" model we saw previously.

RTSP uses a combination of reliable transmission over TCP (used for control) and best-efforts delivery over UDP (used for content) to stream content to users. By this, we mean that the file delivery can start and the client-side application can begin displaying the audio and video content before the complete file has arrived. In terms of our one-hour movie example, this means that the client can request a movie file and watch a "live" feed similar to how one would watch a TV. Along with this "on demand" type service, RTSP also enables the delivery of live broadcast content that would not be possible with traditional download and play type mechanisms.

The Components of RTSP Delivery

RTSP is the control protocol for the delivery of multimedia content across IP networks. It is based typically on TCP for reliable delivery and has a very similar operation and syntax to HTTP. RTSP is used by the client application to communicate to the server information such as the media file being requested, the type of application the client is using, the mechanism of delivery of the file (unicast or multicast, UDP or TCP), and other important control information commands such as DESCRIBE, SETUP, and PLAY. The actual multimedia content is not typically delivered over the RTSP connection(s), although it can be interleaved if required. RTSP is analogous to the remote control of the streaming protocols.

16.4.1 Real Time Transport Protocol (RTP)

RTP is the Internet-standard protocol for the transport of real-time data, including audio and video. It can be used for media-on-demand as well as interactive services such as Internet telephony. RTP consists of a data and a control part. The latter is called RTCP.

The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and video), including timing reconstruction, loss detection, security and content identification.

RTP is the protocol used for the actual transport and delivery of the real-time audio and video data. As the delivery of the actual data for audio and video is typically delay sensitive, the lighter weight UDP protocol is used as the Layer 4 delivery mechanism, although TCP might also be used in environments that suffer higher packet loss. The RTP flow when delivering the content is unidirectional from the server to the client. One interesting part of the RTP operation is that the source port used by the server when sending the UDP data is always even-although it is dynamically assigned. The destination port (i.e., the UDP port on which the client is listening) is chosen by the client and communicated over the RTSP control connection.

16.4.2 Real Time Control Protocol (RTCP)

RTCP is a complimentary protocol to RTP and is a bidirectional UDP-based mechanism to allow the client to communicate stream-quality information back to the object server. The RTCP UDP communication always uses the next UDP source port up from that used by the RTP stream, and consequently is always odd.

RTCP provides support for real-time conferencing of groups of any size within an internet. This support includes source identification and support for gateways like audio and video bridges as well as multicast-to-unicast translators. It offers quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

16.5 SSL : Secure Sockets Layer

This protocol is neither a Layer 4 transport protocol nor an Application layer protocol, but one

that sits between these layers to provide security services to many modern Internet applications. Secure Sockets Layer, or SSL, has been one of the major forces in Internet security technology since its inception by Netscape Communications, and continues to be included in all major browsers. This has enabled Web application developers to deliver secure content and services using traditional HTTP servers with few changes required in terms of the setup of the basic server or restructuring of the HTML content. The other major advantage of the integration of SSL into all major browsers is its transparency to the user. SSL typically gets used without the knowledge of the client, other than the appearance of a small padlock in the corner of the browser window, thus meaning that no additional level of expertise is required to use Internet applications with this security.

The Need for Application Security

The need for security within Internet applications is clear—the Internet is still a public network with little or no security infrastructure designed to protect all users. Imagine using the online services of your favorite bank. Passing important data such as your bank account number, password, and balance across the Internet using only HTTP represents a huge personal security risk, as the data is potentially visible to any device sitting between your browser and the bank's Web site. SSL can be used very effectively to hide all of all the application data as it traverses the Internet to prevent anybody snooping the connection from reading personal data—a process referred to as encryption.

The second important feature provided by *SSL* for Internet application is authentication; in other words, the ability for the client to be able to distinguish the Web site as valid. Imagine in our previous bank example if another rogue site were to masquerade as the bank's Web site. This might allow the rogue site to intercept the personal and banking details of thousands of customers, not a welcome situation. SSL provides mechanisms to implement authentication as a way for each side to identify itself to the other.

The final security element that is provided by SSL is tamper detection. Imagine finally that someone were to sit between the client and the bank's Web site and change certain pieces of data as they pass back and forth. This would give the opportunity to alter key personal and banking data and potentially set up fraudulent transactions. SSL provides mechanisms for each side to ensure that the Application layer data being sent and received has not changed in any way as it traverses the Internet.

For the Internet to continue to grow, not only in size, but also as a credible medium for business and commerce, it must be able to provide mechanisms such as SSL as a way to guarantee security.

16.6 SIP : Session Initiation Protocol

The Session Initiation Protocol (SIP) is a signalling protocol used for establishing sessions in an IP network. A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session. The ability to establish these sessions means that a host of innovative services become possible, such as voice-enriched e-commerce, web page click-to-dial, Instant Messaging with buddy lists, and IP Centrex services.

SIP is a request-response protocol that closely resembles two other Internet protocols, HTTP and SMTP (the protocols that power the world wide web and email); consequently, SIP sits comfortably alongside Internet applications. Using SIP, telephony becomes another web application and integrates easily into other Internet services. SIP is a simple toolkit that service providers can use to build converged voice and multimedia services.

16.7 Internet e-mail Protocols

In this section first we will discuss meanings of Internet and E-mail and then we will discuss Internet E-mail protocols.

16.7.1 Introduction to Internet

The Internet is a computer network made up of thousands of networks worldwide. No one knows exactly how many computers are connected to the Internet. It is certain, however, that these number in the millions and are growing. No one is in charge of the Internet. There are organizations that develop technical aspects of this network and set standards for creating applications on it, but no governing body is in control. The Internet backbone, through which Internet traffic flows, is owned by private companies.

The Internet enables the individual user to reach other people and institutions all over the world and exchange or obtain information. Companies can provide via the Internet information about them, or obtain information about other companies, and offer their own products and services.

The Internet was created by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1960's, and was first known as the ARPANet. At that stage the Internet's first computers were at academic and government institutions. They were mainly used for accessing files and to send email. Since 1983 the Internet has accommodated a lot of changes and continues to keep developing. The last two decades has seen the Internet accommodate such things as network LANs and ATM and frame switched services. The Internet continues to evolve with it becoming available on mobile phones and pagers and possibly on televisions in the future.

"Internet" refers to the global information system that,

- Is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions.
- Is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions, and/or other IP-compatible protocols.
- Provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure.

Internet offers many opportunities for information and communication. Everyone who has access to the Internet can make use of the following services:

- Electronic mail (e-mail) : To send and receive post (mail).
- SMS (Short Message Service) : To send messages from GSM to GSM, or from PC to GSM.
- Telnet, or remote login : To log in to another computer from a distance.
- FTP (File Transfer Protocol) : To receive files from a computer at a distance and to examine and store them on your own computer.
- Gopher : A text-only, non-graphic method to receive Internet documents.
- Video-conferencing.
- World Wide Web.

All computers on the Internet communicate with one another using the Transmission Control Protocol/Internet Protocol suite, abbreviated to TCP/IP. Computers on the Internet use client/server architecture. This means that the remote server machine provides files and services to the user's local client machine. Software can be installed on a client computer to take advantage of the latest access technology.

Components of the Internet

- **World Wide Web**

The World Wide Web (Web or WWW) is a system of Internet servers that supports hypertext to access several Internet protocols on a single interface. Almost every protocol type available on the Internet is accessible on the Web. This includes e-mail, FTP, Telnet, and Usenet News. In addition to these, the World Wide Web has its own protocol: HyperText Transfer Protocol, or HTTP.

The World Wide Web provides a single interface for accessing all these protocols. This creates a convenient and user-friendly environment. It is not necessary to be conversant in these protocols within separate, command-level environments, as was typical in the early days. The Web gathers together these protocols into a single system. Because of this feature, and because of the Web's ability to work with multimedia and advanced programming languages, the Web is the fastest-growing component of the Internet.

The World Wide Web consists of files, called pages or home pages, containing links to documents and resources throughout the Internet.

- **e-Mail**

Electronic mail or e-mail allows computer users locally and worldwide to exchange messages. Each user of e-mail has a mailbox address to which messages are sent. Messages sent through e-mail can arrive within a matter of seconds. (Discussed in next section).

- **Telnet**

Telnet is a program that allows you to log into computers on the Internet and use online databases, library catalogs, chat services, and more. (Discussed later on)

- **FTP**

FTP stands for File Transfer Protocol. This is both a program and the method used to transfer files between computers.

- **Usenet News**

Usenet News is a global electronic bulletin board system in which millions of computer users exchange information on a vast range of topics. The major difference between Usenet News and e-mail discussion groups is the fact that Usenet messages are stored on central computers, and users must connect to these computers to read or download the messages posted to these groups. This is distinct from e-mail distribution, in which messages arrive in the electronic mailboxes of each list member.

Usenet itself is a set of machines that exchanges messages, or articles, from Usenet discussion forums, called newsgroups. Usenet administrators control their own sites, and decide which (if any) newsgroups to sponsor and which remote newsgroups to allow into the system.

- **Chat And Instant Messaging**

Chat programs allow users on the Internet to communicate with each other by typing in real time. They are sometimes included as a feature of a Web site, where users can log into the "chat room" to exchange comments and information about the topics addressed on the site. Chat may take other, more wide-ranging forms. For example, America Online is well known for sponsoring a number of topical chat rooms.

Internet Relay Chat (IRC) is a service through which participants can communicate to each other on hundreds of channels. These channels are usually based on specific topics. To access IRC, you must use an IRC software program.

16.7.2 Introduction to e-mail

Electronic mail (or e-mail) can be defined as "the exchange of computer-stored messages by telecommunications". These messages, usually in text form, are sent from one computer via telephone lines. When you send a message, it usually is stored on a remote computer until the receiver goes online and checks his or her mail. Electronic mail, or e-mail, is the most widely used Internet service. E-mail makes it possible for you to:

- Communicate with students.
- Send or post student assignments.
- Receive student papers and projects.
- Establish class discussions online.
- Communicate with friends who have e-mail accounts.
- Meet and interact with people all over the world.
- Participate in electronic conferences and discussions on an unlimited range of topics.
- Subscribe to electronic services.

- Get answers to technical questions.
- Take online workshops or classes.
- Mail any electronic text and graphics to anyone with an e-mail address.

Advantages of E-mail over other forms of Communication

There are three main advantages : it's cheap, fast and convenient.

Compared with the cost of a phone call, a fax or regular mail (snail mail), you will save money. Some e-mail software programs are even free, and you can even sign up with free e-mail service online. Next, you can't beat the speed of e-mail. Why wait days or weeks for your mail to arrive when you can send a message in seconds? Finally, you can write and send messages easily to one person or a hundred at one time. In addition, you can send graphics, pictures, files, and even movies via e-mail.

The parts of an e-mail address

e-mail addresses often have three parts: (1) the user name, (2) the host or domain name, and (3) the type of domain. Look at this example: **john@yahoo.com**

The first part, **john**, is the username which identifies the recipient. The next part, **yahoo**, is the host or domain name of the mail server where the recipient's mailbox is located. The final part, **.com**, identifies the type of domain (e.g., **.com** for commercial sites, **.edu** for educational institutions, **.org** for non-profit groups, etc.). In some cases, instead of the extension .com, you might see a two-letter country extension like .in for India, .jp for Japan, .fr for France, or .ru for Russia.

e-mail Program Classifications

In general, all email applications fall into at least one of three classifications. Each classification plays a specific role in the process of moving and managing email messages. While most users are only aware of the specific email program they use to receive and send messages, each one is important for ensuring that email arrives at the correct destination. The different types of Email programs are listed below:

- Mail Transfer Agent.
- Mail Delivery Agent.
- Mail User Agent.

Mail Transfer Agent

A Mail Transfer Agent (MTA) transfers email messages between hosts using SMTP. A message may involve several MTAs as it moves to its intended destination. Most users are totally unaware of the presence of MTAs, even though every email message is sent through at least one MTA.

While the delivery of messages between machines may seem rather straightforward, the entire process of deciding if a particular MTA can or should accept a message for delivery is quite complicated. In addition, due to problems from spam, use of a particular MTA is usually restricted by the MTA's configuration or by the lack of access to the MTA's network.

Many modern email client programs can act as an MUA when sending email. However, this action should not be confused with the role of a true MTA. The sole reason email client programs are capable of sending out email (like an MTA) is because the host running the application does not have its own MTA. This is particularly true for email client programs on non-Unix-based operating systems. However, these client programs only send outbound messages to an MTA they are authorized to use and do not directly deliver the message to the intended recipient's email server.

Mail Delivery Agent

A Mail Delivery Agent (MDA) is utilized by the MTA to deliver email to a particular user's mailbox. In many cases, an MDA is actually a Local Delivery Agent (LDA), such as /bin/mail or Procmail. However, Sendmail can also play the role of an MDA, such as when it accepts a message for a local user and appends it to their email spool file. Any program that actually handles a message for delivery to the point where it can be read by an MUA can be considered an MDA. Note that MDAs do not transport messages between systems or interface with the end user.

Many users do not directly utilize MDAs, because only MTAs and MUAs are necessary to send and receive email. However, some MDAs may be used to sort messages before they are read by a user, which is a big help if you receive a lot of email.

Mail User Agent

A Mail User Agent (MUA) is synonymous with an email client application. An MUA is a program that, at the very least, allows a user to read and compose email messages. Many MUAs are capable of retrieving messages via the POP or IMAP protocols, setting up mailboxes to store messages, and sending outbound messages to an MTA.

Protocols

Email, like other network services, uses a variety of protocols. These protocols allow different machines, often running different operating systems and utilizing different email programs, to communicate with one another and transfer mail so it arrives in the proper place. The following protocols are those most commonly used to transfer email from system to system:

- SMTP : Simple Mail Transfer Protocol.
- POP : Post Office Protocol.
- IMAP : Internet Message Access Protocol.
- MIME : Multipurpose Internet Mail Extensions.

Today, email is delivered using client/server architecture. An email message is created using mail client program. This program then sends the message to a server. The server then forwards the massage to the recipient's email server, where the message is then supplied to the recipient's email client.

Unlike the majority of Internet applications, the electronic mail use two different protocols, one for sending and another for getting messages.

For sending mails, SMTP is used. For retrieving messages from the server, POP3 or IMAP

are used. The following protocols discussed are the most commonly used in the transfer of e-mail.

16.7.3 SMTP : Simple Mail Transfer Protocol

The SMTP protocol is used for the transmission of e-mails. SMTP takes care of sending your email to another computer. Normally your email is sent to an email server (SMTP server), and then to another server or servers, and finally to its destination. SMTP can only transmit pure text. It cannot transmit binary data like pictures, sounds or movies. SMTP uses the MIME protocol to send binary data across TCP/IP networks. The MIME protocol converts binary data to pure text.

The Simple Mail Transfer Protocol is the most widely used protocol to send messages by Message Transfer Agents (MTA) on the Internet. SMTP defines the message format and the message transfer agent (MTA) stores and forwards the mail. SMTP was originally designed for only plain text (ASCII text), but MIME and other encoding methods enable executable programs and multimedia files to be attached to and transported with the e-mail message.

MTAs are client or server programs that perform email services, such as sending or receiving mail for a host computer. The protocol is designed to transfer mail independently of any specific transmission subsystem.

Sender and receiver MTAs send commands and replies in a structured, lock-step process. The sender MTA initiates the transaction steps by sending SMTP commands to the receiver. The receiver MTA replies to the sender with numeric reply codes, followed by a text string with additional information about the reply code.

One important point to make about the SMTP protocol is that it does not require authentication. This allows anyone on the Internet to send email to anyone else or even to large groups of people. It is this characteristic of SMTP that makes junk email or spam possible. Modern SMTP servers attempt to minimize this behavior by allowing only known hosts access to the SMTP server. Those servers that do not impose such restrictions are called open relay servers.

The Mail Transaction

The sender MTA initiates a two-way TCP communication channel between it and the receiver MTA, generally on port 25. Once the connection is open, the receiver MTA sends reply code 220 indicating that it is ready. The sender MTA then sends the HELO command with the client host as an argument. The HELO command identifies the sender MTA to the receiver MTA, and the receiver MTA will respond with a reply code 250. This tells the sender MTA that the connection is open and ready to go. This step in the transaction identifies and confirms host addresses for both the sender and receiver MTAs.

The reverse-path is the full reverse source route list, starting with the current client host and ending with the user mailbox. The reverse-path is modified by each MTA as the message travels toward its destination. Before transferring a message to the next relay host, the current host will remove its name from the beginning of the forward-path and adds its name to the beginning of the reverse path.

If the mailbox address is acceptable to the receiver MTA, it transmits a reply code 250 to

the sender MTA. If it cannot fulfill the request, it transmits a reply code of 550. One reason it may not be able to fulfill the request is because the mailbox is incorrect or non-existent.

The next step is for the sender MTA to issue the DATA command. There are no arguments to this command; it simply tells the receiver that the sender is ready to start sending the message. The receiver MTA will transmit a reply code 354 indicating that it is ready to receive the message. Once the sender MTA receives the correct reply code, it sends the mail data to the receiver.

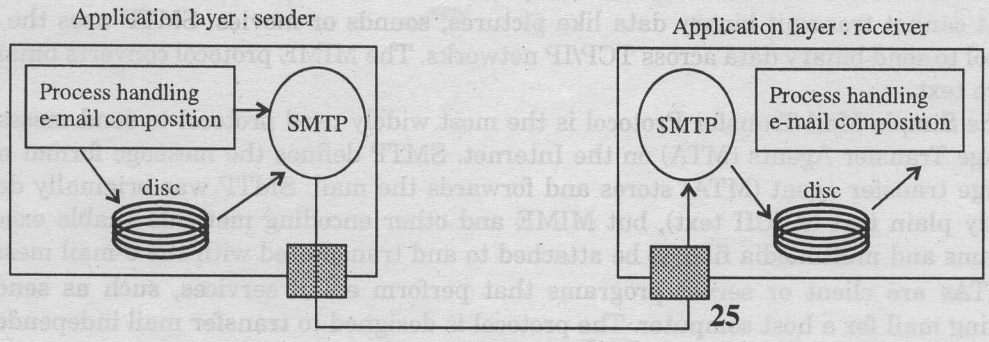


Fig. 16.2 Pictorial view of SMTP

16.7.4 POP : Post Office Protocol

There are two primary protocols used by email client applications to retrieve email from mail servers: the **Post Office Protocol (POP)** and the **Internet Message Access Protocol (IMAP)**. Unlike SMTP, both of these protocols require connecting clients to authenticate using a username and password. By default, passwords for both protocols are passed over the network unencrypted.

The Post Office Protocol (POP) allows email clients to pull off email from remote servers and save those messages on their local machine. Most POP email clients are automatically configured to delete the message on the email server after it has been successfully transferred to the client's system, though this can usually be changed.

The Post Office Protocol, version 3 (POP3) is the most commonly used protocol used for retrieving email messages on the Internet. It is designed for user-to-mailbox access. Facilities are provided for user authentication and mailbox manipulation. Authentication takes the form of a password transmitted as clear text, so POP3 should be used carefully if security is of concern.

POP is fully compatible with important Internet messaging standards, such as Multipurpose Internet Mail Extensions (MIME), which allow for email attachments.

POP works best for users who have one system on which to read email. It also works well for users who do not have a persistent connection to the Internet or the network containing the mail server. Unfortunately for those with slow network connections, POP requires client programs upon authentication to download the entire content of each message. This can take a long time if any messages have large attachments.

The POP protocol has been around for quite a few years and is supported by a large number of e-mail clients. POP was designed to be used as an "offline" protocol in which users connect to a server and download messages in their inbox to their local PC or MAC system. The user has the option of removing messages from the server or leaving them on the server (not recommended.) Leaving messages in the inbox on the server can create a great deal of extra load on the server system, since each time the user reconnects, the POP server must once again scan all of the messages in the inbox to determine which ones are new. Once the messages are downloaded, the user then processes them in offline mode; responding to messages, filing messages to various local folders, etc. The POP protocol does not allow you to create, manage, or manipulate folders or messages on the mail server.

Operation of POP

To connect to a POP server, the email client opens a TCP connection to port 110 on the server. At the time the connection is made, the POP server sends the POP client a greeting, after which the two machines send each other commands and responses specified in the protocol. As part of this communication, the POP client is asked to authenticate itself in what is called the **Authentication State**, where the user's username and password are sent to the POP server. If authentication is successful, then the POP client moves on to the **Transaction State**, where commands like LIST, RETR, and DELE can be used to list, download, and delete the messages from the server, respectively. Messages set to be deleted are not actually removed from the server until the POP client sends the QUIT command to end the session. At this point, the POP server enters the Update State, where it deletes the flagged messages and cleans up any resources remaining from this session.

POP is a much simpler protocol than IMAP, due to the fact that fewer commands can be sent between the client and the server. POP is also somewhat more popular, although most major email clients can use either protocol quite well.

Most POP users only have one system that they use to read email, and they download their messages to that machine for storage. POP also works well if you do not have a constant connection to the Internet or network containing your mail server, although IMAP can now be configured to store messages locally so that you can view them when disconnected from the network.

16.7.5 IMAP : Internet Message Access Protocol

Although the POP3 protocol is the most widely used mail retrieval protocol used today, there is an alternative that overcomes some of its limitations. The Internet Message Access Protocol (IMAP) was designed as a superset of POP3 and enhances both message retrieval and management.

The IMAP protocol is used by email programs (like Microsoft Outlook) just like the POP protocol. The main difference between the IMAP protocol and the POP protocol is that the IMAP protocol will not automatically download all your emails each time your email program connects to your email server. The IMAP protocol allows you to see through your email messages at the email server before you download them. With IMAP you can choose to download your messages or just delete them. This way IMAP is perfect if you need to connect

to your email server from different locations, but only want to download your messages when you are back in your office.

When using an IMAP mail server, email messages remain on the server where users can read or delete them. IMAP also allows client applications to create, rename or delete mail directories on the server to organize and store email.

IMAP is particularly useful for those who access their email using multiple machines. The protocol is also convenient for users connecting to the mail server via a slow connection, because only the email header information is downloaded for messages until opened, saving bandwidth. The user also has the ability to delete messages without viewing or downloading them.

For convenience, IMAP client applications are capable of caching copies of messages locally, so the user can browse previously read messages when not directly connected to the IMAP server. IMAP, like POP, is fully compatible with important Internet messaging standards, such as MIME, which allow for email attachments.

The IMAP protocol provides a superset of the features provided by POP, but also has much more advanced capabilities with regard to message and mailbox management on the mail server. For instance, you can delete messages, search for text in messages, store messages in different folders, or even create and delete folders on the server system. On most systems you have the option of leaving your messages on the server or moving to your local system. The advantage to leaving the messages on the server is that you can then access them from any PC with Internet access and which has an IMAP client loaded. Of course, it is assumed that you will actively manage the messages in your mailbox; move messages out of the inbox, delete what you don't need, etc.

16.7.6 MIME : Multipurpose Internet Mail Extensions

The Multipurpose Internet Mail Extension (MIME) protocol was developed to define a method of moving multimedia files through existing email gateways. Multipurpose Internet Mail Extensions (MIME), a standards-track Internet format defined by an Internet Engineering Task Force Working Group, offers a simple standardized way to represent and encode a wide variety of media types, including textual data in non-ASCII character sets, for transmission via Internet mail.

MIME (Multipurpose Internet Mail Extensions) is one of the Internet protocol standards defined by the Internet Engineering Task Force (IETF). Once associated primarily with electronic mail, MIME has evolved to become an important element supporting multimedia applications on the Net. In order to understand MIME and how it operates, it's helpful to step back and see how it got to where it is today.

MIME defines extensions to SMTP to support binary attachments of arbitrary format. The designers of MIME have learned a lot from the old SMTP protocol and its mailers

Two Main Functions of MIME

- MIME encodes binary data so that it can be passed over the Internet.

- Remember that the Internet is a 7-bit ASCII word.
- Even the 8-bit extensions don't work, because there are issues of line length and file formatting.

- MIME labels encoded data so that the "content" can be properly understood at the other end.
  - For example, "this is a Word document".

The original Internet mail message protocol was designed with text mail messages in mind. A mail message was defined as a block of plain text preceded by specially defined headers specifying routing or other information about the message (e.g., where the message was from, who it was to, whom copies were sent to, etc.). This specification said little about the format of the message *content.* At the time (which was not that long ago!), electronic mail messages were plain text files, so that concerns about the format of content were unwarranted.

Today there is enormous demand for electronic mail that can deliver messages containing components such as HTML text documents, image files, sound, and even video data, in addition to regular text. However, such messages can be widely communicated only if all mail-handling programs share a standard for constructing, encoding and transporting such complex, *multipurpose*, messages.

The MIME protocol provides this common standard. MIME provides an extensible format for including multimedia components within a mail message. MIME defines several document headers, placed inside the document, that specify such things as the nature of a message (multipart or single part), how the message parts are separated, the data content of each part, and the encoding scheme used to encode each part.

16.8 Telnet

The Internet implements the TCP/IP protocol to allow computers to converse with each other over networks. Telnet is an example of a program that makes use of this protocol. This program allows a person to log into and use a distant computer supporting TCP/IP protocols almost as if sitting physically at that computer. It is an application designed for reliable communication between two hosts to support user interaction with a remote system. It is one of the oldest IP protocols and from it several other protocols were developed.

It is best understood in the context of a user with a simple terminal using the local telnet program (known as the client program) to run a login session on a remote computer where his communications needs are handled by a telnet server program. It should be emphasized that the telnet server can pass on the data it has received from the client to many other types of process including a remote login server.

The purpose of the TELNET Protocol is to provide a fairly general, bi-directional, eight-bit byte oriented communications facility. Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other. It is envisioned that the protocol may also be used for terminal-terminal communication ("linking") and process-process communication (distributed computation).

A telnet server listens for connections on TCP port 23. When a connection is opened from

a telnet client to a server, the client attempts to connect to the remote machine using TCP on port 23 and uses a local port above 1023. The remote server will then provide services over that TCP connection. The client sends in ASCII text data and the server responds according to it's design. Telnet is the most basic of all TCP based protocols. When the client receives input from the user, it forwards that information to the telnet server.

Characteristics of Telnet

A TELNET connection is a Transmission Control Protocol (TCP) connection used to transmit data with interspersed TELNET control information. The TELNET Protocol is built upon three main ideas:

- First, the concept of a "Network Virtual Terminal".
- Second, the principle of negotiated options
- Third, a symmetric view of terminals and processes.

1. When a TELNET connection is first established, each end is assumed to originate and terminate at a "Network Virtual Terminal" or NVT. An NVT is an imaginary device that provides a standard, network-wide, intermediate representation of a canonical terminal. This eliminates the need for "server" and "user" hosts to keep information about the characteristics of each other's terminals and terminal handling conventions. All hosts, both user and server, map their local device characteristics and conventions so as to appear to be dealing with NVT over the network, and each can assume a similar mapping by the other party. The NVT is intended to strike a balance between being overly restricted (not providing hosts a rich enough vocabulary for mapping into their local character sets), and being overly inclusive (penalizing users with modest terminals). The "user" host is the host to which the physical terminal is normally attached, and the "server" host is the host that is normally providing some service.
2. The principle of negotiated options takes cognizance of the fact that many hosts will wish to provide additional services over and above those available within an NVT, and many users will have sophisticated terminals and would like to have elegant, rather than minimal, services. Independent of, but structured within the TELNET Protocol are various "options" that will be sanctioned and may be used with the "DO, DON'T, WILL, WON'T" structure to allow a user and server to agree to use a more elaborate (or perhaps just different) set of conventions for their TELNET connection. Such options could include changing the character set, the echo mode, etc. The basic strategy for setting up the use of options is to have either party (or both) initiate a request that some option take effect. The other party may then either accept or reject the request. If the request is accepted the option immediately takes effect; if it is rejected the associated aspect of the connection remains as specified for an NVT. Clearly, a party may always refuse a request to enable, and must never refuse a request to disable some option since all parties must be prepared to support the NVT.
3. The symmetry of the negotiation syntax can potentially lead to non-terminating acknowledgment loops—each party seeing the incoming commands not as acknow-

ledgments but as new requests which must be acknowledged. To prevent such loops, the following rules prevail:

- Parties may only request a change in option status; i.e., a party may not send out a "request" merely to announce what mode it is in.
- If a party receives what appears to be a request to enter some mode it is already in, the request should not be acknowledged. This non-response is essential to prevent endless loops in the negotiation. It is required that a response be sent to requests for a change of mode—even if the mode is not changed.
- Whenever one party sends an option command to a second party, whether as a request or an acknowledgment, and use of the option will have any effect on the processing of the data being sent from the first party to the second, then the command must be inserted in the data stream at the point where it is desired that it take effect.

For the average web user telnet is not so important as it was a few years ago, since many of the features it once made available are now otherwise accessible via browsers. People who have accounts on multi-user computers (particularly those using unix/linux operating systems), however, still find telnet to be an indispensable tool for accessing their work, from home, from the road or from around the world.

16.9 SNMP : Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) is an application-layer protocol designed to facilitate the exchange of management information between network devices. By using SNMP-transported data (such as packets per second and network error rates), network administrators can more easily manage network performance, find and solve network problems, and plan for network growth.

Simple Network Management Protocol (**SNMP**) is a widely used protocol for monitoring the health and welfare of network equipment (e.g., routers), computer equipment and even devices like UPSs.

SNMP is based on the manager/agent model. SNMP is referred to as "simple" because the agent requires minimal software. Most of the processing power and the data storage resides on the management system, while a complementary subset of those functions resides in the

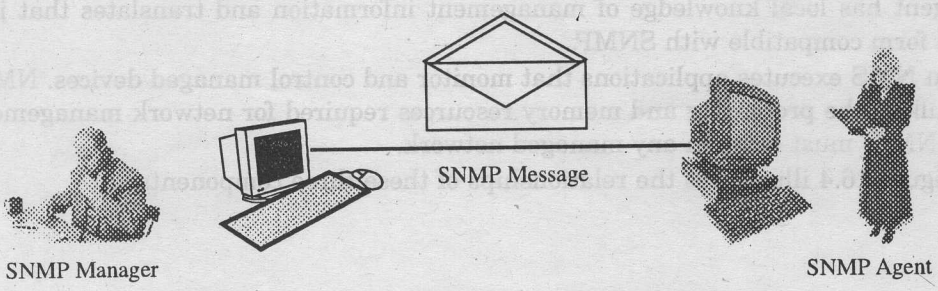


Fig. 16.3 Pictorial view of SNMP

managed system. SNMP is an asymmetric protocol, operating between a management station (smart) and an agent (dumb).

Like the Transmission Control Protocol (TCP), SNMP is an Internet protocol. Internet protocols are created by the Internet community, a group of individuals and organizations that developed and/or regularly use a large, diverse international network called the Internet. The Internet derived from the Advanced Research Projects Agency network (ARPANET), which was created by packet switching researchers in the early 1970s.

There are two versions of SNMP: Version 1 and Version 2. Most of the changes introduced in Version 2 increase SNMP's security capabilities. Other changes increase interoperability by more rigorously defining the specifications for SNMP implementation. SNMP's creators believe that after a relatively brief period of coexistence, SNMP Version 2 (SNMPv2) will largely replace SNMP Version 1 (SNMPv1). SNMP is part of a larger architecture called the Internet Network Management Framework (NMF).

Today, SNMP is the most popular protocol for managing diverse commercial internetworks as well as those used in universities and research organizations. SNMP-related standardization activity continues even as vendors develop and release state-of the-art, SNMP-based management applications. SNMP is a relatively simple protocol, yet its feature set is sufficiently powerful to handle the difficult problems presented in trying to manage today's heterogeneous networks.

SNMP Basic Components

An SNMP-managed network consists of three key components:

- Managed devices.
- Agents.
- Network-management systems (NMSs).

A **managed device** is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.

An **agent** is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP.

An **NMS** executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs must exist on any managed network.

Figure 16.4 illustrates the relationships of these three components.

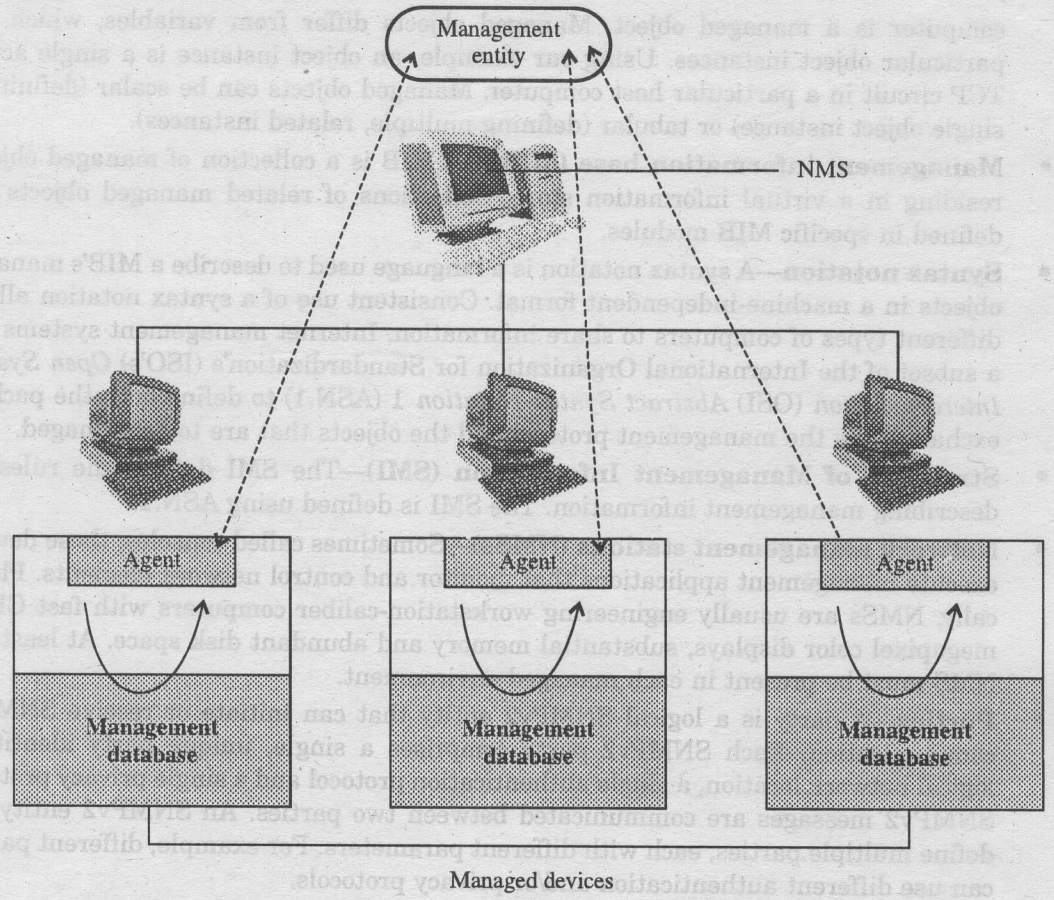


Fig. 16.4 SNMP-managed network consists of managed devices, agents, and NMSs

SNMP Technology

SNMP is part of the Internet network management architecture. This architecture is based on the interaction of many entities that are described in the following section.

The Internet Management Model

A network management system comprises:

- **Network elements**—Sometimes called managed devices, network elements are hardware devices such as computers, routers, and terminal servers that are connected to networks.
- **Agents**—Agents are software modules that reside in network elements. They collect and store management information such as the number of error packets received by a network element.
- **Managed object**—A managed object is a characteristic of something that can be managed. For example, a list of currently active TCP circuits in a particular host

computer is a managed object. Managed objects differ from variables, which are particular object instances. Using our example, an object instance is a single active TCP circuit in a particular host computer. Managed objects can be scalar (defining a single object instance) or tabular (defining multiple, related instances).

- **Management information base (MIB)**—A MIB is a collection of managed objects residing in a virtual information store. Collections of related managed objects are defined in specific MIB modules.
- **Syntax notation**—A syntax notation is a language used to describe a MIB's managed objects in a machine-independent format. Consistent use of a syntax notation allows different types of computers to share information. Internet management systems use a subset of the International Organization for Standardization's (ISO's) *Open System Interconnection* (OSI) *Abstract Syntax Notation* 1 (ASN.1) to define both the packets exchanged by the management protocol and the objects that are to be managed.
- **Structure of Management Information (SMI)**—The SMI defines the rules for describing management information. The SMI is defined using ASN.1.
- **Network management stations (NMSs)**—Sometimes called consoles, these devices execute management applications that monitor and control network elements. Physically, NMSs are usually engineering workstation-caliber computers with fast CPUs, megapixel color displays, substantial memory and abundant disk space. At least one NMS must be present in each managed environment.
- **Parties**—A party is a logical SNMPv2 entity that can initiate or receive SNMPv2 communication. Each SNMPv2 party comprises a single, unique party identity, a logical network location, a single authentication protocol and a single privacy protocol. SNMPv2 messages are communicated between two parties. An SNMPv2 entity can define multiple parties, each with different parameters. For example, different parties can use different authentication and/or privacy protocols.
- **Management protocol**—A management protocol is used to convey management information between agents and NMSs. SNMP is the Internet community's de facto standard management protocol.

SNMP Basic Commands

Managed devices are monitored and controlled using four basic SNMP commands: read, write, trap and traversal operations.

The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.

The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.

The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.

Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

The SNMPv2 specifications discuss two primary security protocols: one for authentication and one for privacy. These are the **Digest Authentication Protocol** and the **Symmetric Privacy Protocol**. The authentication protocol is designed to reliably identify the integrity of the originating SNMPv2 party. It consists of authentication information required to support the authentication protocol used. The privacy protocol is designed to protect information within the SNMPv2 message from disclosure. Only authenticated messages can be protected from disclosure. In other words, authentication is required for privacy.

The Digest Authentication Protocol verifies that the message received is the same one that was sent. Data integrity is protected using a 128-bit message digest calculated according to the Message Digest 5 (MD5) algorithm. The digest is calculated at the sender and enclosed with the SNMPv2 message. The receiver verifies the digest. A secret value, known only to the sender and the receiver, is prefixed to the message. After the digest is used to verify message integrity, the secret value is used to verify the message's origin.

To help ensure message privacy, the Symmetric Privacy Protocol uses a secret encryption key known only to the sender and the receiver. Before the message is authenticated, this protocol uses the Data Encryption Standard (DES) algorithm to effect privacy. DES is a documented National Institute of Standards and Technology (NIST) and American National Standards Institute (ANSI) standard. (DES discussed in chapter 17)

• • •

17

Network Security

Introduction To Network Security

Network security is a complicated subject, historically only tackled by well-trained and experienced experts. However, as more and more people become "wired", an increasing number of people need to understand the basics of security in a networked world.

Network security involves any and all countermeasures taken to protect a network from threats to its integrity. As modern networks have continued to grow and as more and more networks have been connected to the public Internet, the threats to the integrity and privacy of a company's networks have also grown. The attacks that are made on a network are increasingly more complex and pervasive, and the tools used for such purposes are easy to acquire. For example, anyone can log on to an Internet search engine and perform a search on hacking and be presented with an immense amount of sites that offer information and tools on hacking. Therefore, the need for network security is obvious. But what exactly is involved in a good network security policy?

A network security plan should be as comprehensive as possible. This not only includes physical aspects, such as locating your servers in a secure room, use of fault tolerance and power protection, but also includes all the steps taken to protect the data on your network. This would include setting up user accounts and passwords, setting permissions and access rights, the use of encryption for sensitive data, virus monitoring, and intrusion detection, to name just a few of the critical elements involved. And if your network is connected to any other networks, such as the Internet, then your network security plan should also include the implementation of a firewall. An effective network security strategy requires identifying threats and then choosing the most effective set of tools to combat them.

Definition

- *Security is a state of well-being of information and infrastructures in which the possibility of successful yet undetected theft, tampering, and disruption of information and services is kept low or tolerable.*
- *Security rests on confidentiality, authenticity, integrity, and availability.*

17.1 Need of Network Security

Information is a strategic resource. A significant portion of organizational budget is spent on managing information. There are many types of information and it has several security related objectives like:

- Confidentiality (secrecy)—protect information value.
- Integrity—protect information accuracy.
- Availability—ensure information delivery.

Network Security Goals

The various goals of network security are listed below:

- **Confidentiality :** Confidentiality is the concealment of information or resources. Information should not be disclosed or revealed to unauthorized persons; the same as privacy—keeping private documents private.
- **Integrity :** Integrity refers to the trustworthiness of data or resources in terms of preventing improper and unauthorized changes. Information should be safe from modification without the sender's or owner's consent.
- **Authentication :** Authenticity is the identification and assurance of the origin of information. The sender of the information really is who he/she claims to be.
- **Non-repudiation :** Principally, the sender cannot deny that he/she actually sent the message (non-repudiation of origin). In some cases also, the sender can prove the information was available to the recipient (non-repudiation of delivery) and the receiver cannot deny receipt of the message (non-repudiation of receipt).
- **Accessibility :** The information should be available to authorized users when required; and not available to unauthorized users.

These goals and their possible solutions are listed in the table below:

Security goal	Non-electronic Solutions	Electronic Solutions
Confidentiality	Sealed letter, opaque envelope	Encryption
Integrity	Hologram on credit card, indelible ink	Message digest
Authentication	Photo ID card, knowing personal details	Digital signature
Non-repudiation	Notarized signature, registered post	Digital signature
Accessibility	Locks, guards	Password, firewall

Network Security Tradeoffs

In network security, as in any form of information security, sometimes compromises have to be made between opposing forces.

Security versus Access : Protection of information involves a trade-off between security and accessibility. At one extreme information could be completely secure by storing it on a computer that no one ever uses. However this totally defeats the purpose for which the information was created. A more realistic approach to network security is to assess the possible threats to the data and then minimize the threat as much as possible without making it difficult for legitimate users to access the information.

Cost versus Benefits : Security economics requires that cost of security measures to protect information—both financial costs and user inconvenience—should not exceed the value of the information. Network managers must justify the cost of controls and safeguards in light of the expected frequency of attacks and the anticipated loss resulting from a successful attack. For example, an expensive firewall can be justified to protect a vital and vulnerable asset such as a company's Web site. However, the company's office equipment database would not require any projection other than normal back-up procedures.

17.1.1 Security Threats and Attacks

A **threat** is a potential violation of security. These are flaws in design, implementation, and operation. It is a circumstance, condition or event with the potential to cause harm to personnel and/or network resources in the form of destruction, disclosure, modification of data, denial of service, and/or fraud, waste, and abuse. Network security threats include impersonation, denial of service, packet replay, and packet modification.

An **attack** is any action that violates security. There are generally two types of attacks:

- Active attack.
- Passive attack.

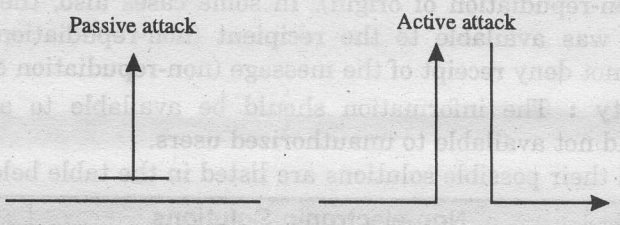


Fig. 17.1 Types of Attacks

Active attack : It is an attack on a computer network with the intent to insert, destroy, modify or divert data. Active attacks are difficult to prevent because preventive security reduces accessibility. Instead the primary goal is to detect active attacks and quickly recover from any disruption or delays caused by the attack. Active attack can actively modify communications or data.

Passive Attacks : It is an attack on a computer network with the intent to monitor network data transmission to read messages or analyze traffic. Passive attacks are difficult to detect because they do not involve alteration of the data. The emphasis is on prevention, rather than detection. Passive attack can only observe communications or data.

Types and Sources of Network Threats

1. Denial-of-Service

DoS (Denial-of-Service) attacks are probably the worst, and most difficult to address. These are the worst, because they're very easy to launch, difficult (sometimes impossible) to track, and it isn't easy to refuse the requests of the attacker, without also refusing legitimate requests for service.

Multi-user, multi-tasking operating systems are subject to denial of service attacks where one user can render the system unusable for legitimate users by hogging a resource or damaging or destroying resources so that they cannot be used. Denial of service attacks may be caused deliberately or accidentally. Taking precautions to prevent a system against unintentional denial of service attacks will help to prevent intentional denial of service attacks.

Three common forms of network denial of service attacks are service overloading, message flooding, and signal grounding. The premise of a DoS attack is simple: send more requests to the machine than it can handle. There are toolkits available in the underground community that make this a simple matter of running a program and telling it which host to blast with requests. The attacker's program simply makes a connection on some service port, perhaps forging the packet's header information that says where the packet came from, and then dropping the connection. If the host is able to answer 15 requests per second, and the attacker is sending 45 per second, obviously the host will be unable to service all of the attacker's requests, much less then any legitimate requests.

It is important for system administrators to protect against denial of service threats without denying access to legitimate users. In general, denial of service attacks are hard to prevent. Many denial of service attacks can be hindered by restricting access to critical accounts, resources, and files, and protecting them from unauthorized users. Some things that can be done to reduce the risk of being stung by a denial of service attack include:

- Not running your visible-to-the-world servers at a level too close to capacity.
- Using packet filtering to prevent obviously forged packets from entering into your network address space.
- Keeping up-to-date on security-related patches for your hosts' operating systems.

2. Eavesdropping

Eavesdropping allows a cracker to make a complete transcript of network activity. As a result, a cracker can obtain sensitive information, such as, passwords, data and procedures for performing functions. It is possible for a cracker to eavesdrop using wiretapping, eavesdropping by radio and eavesdropping via auxiliary ports on terminals. It is also possible to

eavesdrop using software that monitors packets sent over the network. In most cases, it is difficult to detect that a cracker is eavesdropping.

Many network programs, such as telnet and ftp are vulnerable to eavesdroppers obtaining passwords that are often sent across the network unencrypted. Network programs that involve file transfer are susceptible to eavesdroppers obtaining the contents of files. Encryption can be used to prevent eavesdroppers from obtaining data traveling over unsecured networks.

3. Unauthorized Access

Unauthorized access is a very high-level term that can refer to a number of different sorts of attacks. The goal of these attacks is to access some resource that your machine should not provide the attacker. For example, a host might be a web server, and should provide anyone with requested web pages. However, that host should not provide command shell access without being sure that the person making such a request is someone who should get it, such as a local administrator.

4. Executing Commands Illicitly

It's obviously undesirable for an unknown and untrusted person to be able to execute commands on your server machines. There are two main classifications of the severity of this problem:

- Normal user access
- Administrator access

A normal user can do a number of things on a system (such as read files, mail them to other people, etc.) that an attacker should not be able to do. This might, then, be all the access that an attacker needs. On the other hand, an attacker might wish to make configuration changes to a host (perhaps changing its IP address, putting a start-up script in place to cause the machine to shut down every time it's started, or something similar). In this case, the attacker will need to gain administrator privileges on the host.

5. Confidentiality Breaches

There is certain information that could be quite damaging if it fell into the hands of a competitor, an enemy, or the public. In these cases, it's possible that compromise of a normal user's account on the machine can be enough to cause damage (Like obtaining information that can be used against the company, etc.)

While many of the executors of these sorts of break-ins are merely thrill-seekers interested in nothing more than to see a shell prompt for your computer on their screen, there are those who are more malicious. It can also be possible that someone who is normally interested in nothing more than the thrill could be persuaded to do more: perhaps a dishonest competitor is willing to hire such a person to hurt you.

6. Packet Replay

Packet replay refers to the recording and re-transmission of message packets in the network. Packet replay is a significant threat for programs that require authentication sequences, because an intruder could replay legitimate authentication sequence messages to gain access

to a system. Packet replay is frequently undetectable, but can be prevented by using packet time-stamping and packet sequence counting.

7. Packet Modification

Packet modification is a significant integrity threat that involves one system intercepting and modifying a packet destined for another system. In many cases, packet information may not only be modified, but it may also be destroyed.

8. Destructive Behavior

Among the destructive sorts of break-ins and attacks, there are two major categories.

- **Data Diddling**
  The data diddler is likely the worst sort, since the fact of a break-in might not be immediately obvious. Perhaps he's toying with the numbers in your spreadsheets, or changing the dates in your projections and plans. Maybe he's changing the account numbers for the auto-deposit of certain paychecks. In any case, rare is the case when you'll come in to work one day, and simply know that something is wrong. An accounting procedure might turn up a discrepancy in the books three or four months after the fact. Trying to track the problem down will certainly be difficult, and once that problem is discovered, how can any of your numbers from that time period be trusted? How far back do you have to go before you think that your data is safe?
- **Data Destruction**
  Some of those perpetrate attacks are simply twisted jerks who like to delete things. In these cases, the impact on your computing capability—and consequently your business—can be nothing less than if a fire or other disaster caused your computing equipment to be completely destroyed.

Where do these threats come from?

An attacker can gain access to your equipment through any connection that you have to the outside world. This includes Internet connections, dial-up modems and even physical access. In order to be able to adequately address security, all possible avenues of entry must be identified and evaluated. The security of that entry point must be consistent with your stated policy on acceptable risk levels.

17.1.2 Security Management Techniques

Security management is the glue that holds together the other building blocks of a strong security solution. It involves facilities for coordinating users' service requirements and mechanism implementations throughout the enterprise network and across the Internet.

Security Policy and Mechanism

- **Policy :** A statement of what is, and is not allowed.
- **Mechanism :** A procedure, tool, or method of enforcing a policy.

Security mechanisms implement functions that help prevent, detect, and respond to

recovery from security attacks. Security functions are typically made available to users as a set of security services through APIs or integrated interfaces.

Three basic building blocks are used:

- Encryption is used to provide confidentiality. It can provide authentication and integrity protection.
- Digital signatures are used to provide authentication, integrity protection, and non-repudiation.
- Checksums/hash algorithms are used to provide integrity protection and provide authentication.

One or more security mechanisms are combined to provide a security service. We will be discussing these security mechanisms in detail in next sections.

A typical security protocol provides one or more services. Services are built from mechanisms. Mechanisms are implemented using algorithms.

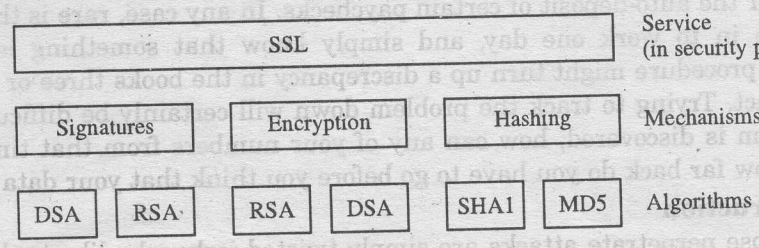


Fig. 17.2 Security Levels

Network Security Tools

The various network security tools are listed below:

Antivirus software packages : These packages counter most virus threats if regularly updated and correctly maintained.

Secure network infrastructure : Switches and routers have hardware and software features that support secure connectivity, perimeter security, intrusion protection, identity services, and security management. Dedicated network security hardware and software-tools such as firewalls and intrusion detection systems provide protection for all areas of the network and enable secure connections.

Virtual private networks : These networks provide access control and data encryption between two different computers on a network. This allows remote workers to connect to the network without the risk of a hacker or thief intercepting data.

Identity services : These services help to identify users and control their activities and transactions on the network. Services include passwords, digital certificates, and digital authentication keys.

Encryption : Encryption ensures that messages cannot be intercepted or read by anyone other than the authorized recipient.

None of these approaches alone will be sufficient to protect a network, but when they are

layered together, they can be highly effective in keeping a network safe from attacks and other threats to security.

17.2 Cryptography

Cryptography is the study of secret (crypto-) writing (-graphy). It is concerned with developing algorithms that may be used to :

- Conceal the context of some message from all except the sender and recipient (privacy or secrecy), and/or
- Verify the correctness of a message to the recipient **(authentication)**.

It forms the basis of many technological solutions to computer and communications security problems.

Cryptography, the encoding of messages to render them unreadable by anyone other than their intended recipient(s), is centuries old. Caesar cipher is one of the traditional cryptography techniques. The "Caesar Cipher" is so named because it was used by Julius Caesar. With the advent of modern computer technology, many of these older ciphers became trivially crackable using brute-force attacks. Modern cryptography, essential to the security of computer networks, is done with complex algorithms implemented on high-speed computer systems.

There are two kinds of cryptosystems: **symmetric** and **asymmetric**. Symmetric cryptosystems use the same key (the secret key) to encrypt and decrypt a message and asymmetric cryptosystems use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it. Asymmetric cryptosystems are also called public key cryptosystems.

Symmetric cryptosystems have a problem: how do you transport the secret key from the sender to the recipient securely and in a tamperproof fashion? If you could send the secret key securely, then, in theory, you wouldn't need the symmetric cryptosystem in the first place—because you would simply use that secure channel to send your message. Frequently, trusted couriers are used as a solution to this problem. Another, more efficient and reliable solution is a public key cryptosystem. A public-key cryptosystem has public encryption and private decryption.

17.2.1 Classical Cryptographic Techniques

Classical Cryptographic Techniques have two basic components of classical ciphers:

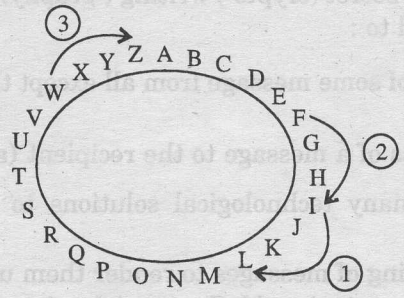
- **Substitution :** In substitution ciphers letters are replaced by other letters.
- **Transposition :** In transposition ciphers the letters are arranged in a different order. These ciphers may be:
  - *Monoalphabetic*—Only one substitution/ transposition is used.
  - *Polyalphabetic*—Where several substitutions/ transpositions are used.

Several such ciphers may be concatenated together to form a **product cipher**.

Caesar Cipher - A Monoalphabetic Cipher

Replace each letter of message by a letter a fixed distance away. E.g., use the 3rd letter on. It was reputedly used by Julius Caesar.

e.g., L FDPH L VDZ L FRQTXHUHG
I CAME I SAW I CONQUERED



i.e., mapping is
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

Fig. 17.3 A monoalphabetic cipher

Decryption consists of :

- Writing the message out in columns.
- Reading off the message by reordering columns.
- Uses read out keys.

Generally speaking, computer cryptographic tasks can be broken into two general categories:

- □ Encryption
- □ Authentication

17.2.2 Encryption

Encryption refers to the scrambling of information so that the original message cannot be determined by unauthorized recipients. An encryption algorithm is applied to the message, referred to as the plaintext, and a key to produce ciphertext, which ideally appears to be random bits. A decryption algorithm converts the ciphertext back into plaintext, but only if given the correct key. Conventional, or symmetric, algorithms use the same key for both encryption and decryption. Public key algorithms use paired keys, one for encryption and another for decryption.

Important terms to be used in this section are listed below:

Plaintext : The original intelligible message.
Ciphertext : The transformed message.