LESSON 36:
END-TO-END LAYERS AND PBX

Objective

To provide a detailed understanding of the concepts of end to end layers and PBX.

Introduction

End-to-end Layers

Many data networks were operational before the OSI model was designed. These networks worked well throughout upto the network layer. But little had been done about the transport layer and above. Consequently, the design of the bottom three layers of OSI was highly influenced by pre-OSI developments.

OSI design is expected to dominate the upper four layers in the years to come except for the influence of transmission control protocol (TCP) of ARPANET at the transport layer.

In this section we discuss following layers of OSI model:

(1) Transport Layer

(2) Session Layer

(3) Presentation Layer

(4) Application Layer.

Transport Layer

Transport Layer is the first end-to-end layer of the OSI model architecture. The function of this layer is to match the user message characteristics and service requirements with that of the network capabilities. In a packet switched network, the transport entity breaks up a long user message into packets to watch the network capabilities. These packets are reassembled at the destination transport subsystem to reconstruct the user message.

The transport entities may invoke sophisticated error control protocols to provide a reliable session service on an unreliable network. Similarly, a number of low rate user services may be multiplexed to use efficiently a single network connection, or a high data rate requirement from the user may be split into a number of network connections. Multiplexing and splitting that are performed by the transport layer, are transparent to session layer. Same as other layers, transport layer is also concerned with the establishment, control and release of the transport connections between peer entities in the source and destination systems.

End-to-end control and error recovery are also functions of the, transport layer. The need for flow control arises when the speed or the buffer space of the destination machine does not match with that of the source machine. Also, end-to-end error recovery becomes necessary when the source or destination system fails or when the network becomes disconnected due to link failures.

The transport layer buffers transport services regardless of the underlying subnetwork. The user makes his service requests to this layer by specifying certain 'Quality of Service' QOS parameter values.

Some QOS parameters that are of direct interest to the users are:

- Transit time delay
- Residual error rate
- Protection
- Transfer failure probability
- Priority
- Throughput

Transit time delay

A user may specify a desirable and maximum value for the transit time delay.

Residual error rate

It specifies the undetected errors or irrecoverable losses that a user may be willing to accept.

Protection

It deals with the data security.

Transfer failure probability

It is a measure of how well the transport layer is able to meet the user demands in terms of other parameters.

Priority

It enables the user to give more importance to some of the connections than others.

There are other QOS parameters like connection establishment and release delays, and the probability of a connection being terminated j abruptly due to some internal network problems.

These influence the confidence level of a user on the, network when a user makes a request to the transport layer for a connection, one of the following three things may happen:

- Connection may be established as requested.
- Options may be negotiated and a connection established with changed parameter values. This happens when network capabilities are limited and the network is unable to meet the user requirements.
- Connection is rejected as the network is unable to handle even the minimal requirements of the user.

Once a connection is granted, it is the responsibility of the transport layer to provide the necessary QOS demanded by user. To ensure the necessary QOS, different transport protocols are designed. This takes into account the quality of service available from the networks.

For this purpose, networks are classified in three categories:

(1) **Category-A Networks**

These Networks provide reliable error-free service

(2) **Category-B Networks**

These networks provide reliable packet delivery. But occasional network wide resets occur

(3) **Category-C Networks**

These networks provide unreliable service characterized by packet loss and duplication, and high residual error rates.

OSI transport protocol has 5 variants to deal with different categories of networks and to provide varying degrees of service sophistication to the user:

Class 0: Simple protocol with no error recovery or flow control.

Class 1: Basic error recovery protocol

Class 2: Protocol with multiplexing ability

Class 3: Protocol with error recovery and multiplexing

Class 4: Protocol with elaborate error detection and recov mechanisms.

Class 0 and Class 2 are suitable for category-A networks.

Class 1 and Class 3 are suitable for category-B networks

Class 4 are suitable for category-C networks.

Session Layer

The main function of the session layer is to organise different sessions between cooperating entities and perform all related functions like:

1. Synchronisation
2. Failure Management
3. Control, etc. for the successful execution of a session.

Online search of databases, remote job entry, remote login to a time sharing system and file transfer between two systems are all examples of sessions. Different sessions have different requirements. For example, a dialogue session may be two-way simultaneous or one-way alternate. A layer file transfer session may call for rollback points being established in order to recover from system crashes.

An online transaction processing session calls for semaphore management, and file, record and sometimes even item level lack mechanisms. A quarantine service is one which enables a specified number of presentation layer SDUs to be transported to the destination, system but not actually deliver them unless explicitly so requested by sender. Directory service is also offered by the session layer.

In a dialogue session, the interacting entities will have to communicate in an orderly fashion. Generally, it is a half duplex communication process with waits. For example, a user sends a query to a database system and waits for a response. A dialogue session in OSI environment is managed by using a data token. Only an entity which has a token can communicate.

After the communication is over, the data token is transferred to the other entity. At the time of session establishment, a decision is taken as to which of the two entities gets the token first.

Synchronisation

It is the operation in which rollback points (otherwise these are called synchronisation points) are set up in a session in order to recover from failures. Minor synchronisation points fall in between major synchronisation points. Minor and Major synchronisation points are shown in Fig.36.1.

A rollback can go beyond one or more minor synchronisation points. But, rollback cannot go beyond one or more major synchronisation points.

For example, a failure between 10 and 11 in Fig. 36.1 may roll back to any of the points 10, 9, 8, 7 or 6, but not beyond 6. Rolling back is called resynchronisation.

Session layer offers another facility that is called activity management. Here, a user is permitted to divide his total session job into a number of smaller, independent, and parallel asynchronous and perhaps event oriented activities. These activities are somewhat like tasks in a multitasking computer system.

Session layer functions are traditionally connection oriented functions. However, extensive research has been done in connectionless form of dialogue services in the context of distributed systems. This work is generally called Remote Procedure Call (RPC). It is now being made available in the OSI environment. RPC is often considered as an application layer function in the OSI model.

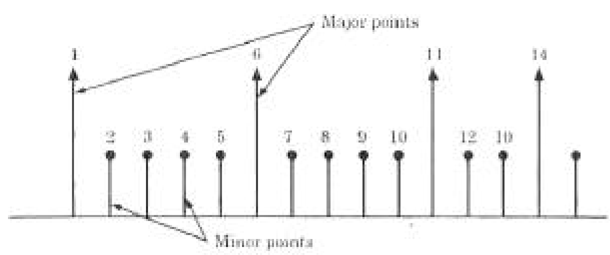

Fig.36.1 Illustration of major major and minor synhronisation points in session layer

Presentation Layer

The purpose of the presentation layer is to represent information to the communicating application entities in a way that preserves the meaning while resolving syntax differences. Syntax differences are resolved by encoding application data into a standard abstract notation. This notation is valid throughout the network. Thus, file format differences (e.g. IBM format or IDEC format), data representation differences (e.g. ASCII or EBCDIC), data structure differences are all resolved by using a standard notation. Data information and formatting may include data compression, encryption etc.

There are two aspects associated with network wide handling of a variety of data in standard form

(1) First, the representation of data in standard form

(2) Second, the transmission of the data as a bit stream across the network.

The standard notation used for representing information across the network is called Abstract Syntax Notation 1 (ASN.1).

The encoded bit stream of the ANS. 1 data objects for transmission across the network is known as the transfer Syntax. The use of ANS.1 as an intermediate representation format is of practical necessity. In the absence of such notation, a large number of conversion programs have to be written to match one representation to all others. If there are n different representations for a data object, n (n -1) conversion programs are required for direct conversion from each representation to every other representation. On the other hand, with the use of ASN.1, only 2n conversion programs are required.

The use of common communication format (CCF) in bibliographic databases and switching systems in telecommunication networks are the examples of applications of concept of using an intermediate standard notation.

ASN.1 data representation technique is somewhat similar to the ones used in programming languages like Pascal and C. There are a few predefined primitive data types and a set of constructors which can be used to build complex data types from the primitive ones. Summary of primitive data type and the constructors are given in Table 36.1. The first four data types are well known.

The primitive data type 'ANY' enables a user to defer type definition of a data object to a later instance like run time.

Initially, 'NULL' is a data item of no value.

Object Identifier

It is a name used to identify objects or subroutines or libraries associated with the processing of ASN.1 or transfer systems.

The meaning of the ASN.1 constructs are given in Table 36.1

Table 36.1 Primitive Data Types and constructs in ASN.1

ASN.l data types	Constructs	Meaning
Interger	Sequence of	Ordered list of single data type
Bollen	Sequence	Ordered list of a mix of data types
Bit string	Set of	Unordered collection of a data type
Octet string	Set	Unordered collection of a various data type
ANY	Choice	Anyone type taken from a list
NULL		
Object identifier		

Object identifier

The transfer syntax for ASN.1 consists of four fields. These fields are shown in Fig. 36.2

Identifier	Length	Data	Flag

Fig. 36.2 ASN.l transfer syntax.

ASN.1 has a flexible syntax notation. It permits a large number of optional parameters. If the optional parameters are not used, a l, NULL value may be transmitted in place of these fields. In such a case, this would result in excessive overhead in transmission and subsequent processing of the PDU. A better scheme is to omit the optional fields that are not used. In this case, a method of identifying the actual parameters that are transmitted is required. This is achieved by 'tagging' the parameters.

There are four types of tags that are used in ASN-1. These tags are:

1. Universal tags
2. Application tags
3. Private tags
4. Context specific tags.

Each tag under a category is specified by number. The identifier field of the transfer syntax has three subfields.

First Subfield

The first two bits specify the tag type, the next bit specifies whether the data object is pf primitive type or the constructed type, and the last five bits the tag number.

Second Subfield

The second subfield specifies the length of the data field.

Third Subfield

The third or the last subfield 'flag' indicates the end of the data in cases where the length is not specified. A provision exists for specifying tag numbers greater than 5-bit quantities and length greater than 8-bit quantities.

In the routing, data compression and encryption, techniques are not standardized in the ISO model. These are left as implementation dependent objects. Run length coding and Huffman coding are the widely used data compression techniques.

Application Layer

Application layer is the highest layer in the OSI reference model. It provides services to the users of OSI environment. This layer provides all services that are directly comprehensive by the users. These services are given as under:

1. Electronic mail or message handling services
2. Directory servicing
3. Cost Allocation
4. Determination of QOS
5. File transfer and management
6. Editors and terminal support services
7. Editors and terminal support services
8. Telematic services like videotext.

In general, every application requires its own software which in turn uses a number of generic supporting packages.

The generic packages and the application specific packages are part and parcel of the application layer. For example, file transfer or remote file access may be used by airlines reservation system, applications etc. Similarly, electronic mail may be used in order placement and processing systems. There is yet another level of software packages that are used by some of these generic applications packages. They include:

1. Connection establishment
2. Transaction processing
3. Database crash recoveries etc.

Two such packages are:

- Association Control Service Element (ACSE)
- Commitment, Concurrency and Recovery (CCR)

Now we discuss file transfer, access and management (FTAM) and Virtual Terminal Support Function (VTSFs). Directory in OS1 environment provide the presentation SAP addresses of the users. The access is based on attributes. These attributes may be the name, address, profession, organisational affiliation, telephone number etc.

File transfer and Remote file access are two of the most common applications in a network environment.

In file transfer, an entire file (Binary, textual, or graphical etc.) is copied from one system to another. Remote file access in to file transfer except that only parts of the files are read or written

The techniques used for remote file access and file transfer are similar.

FTAM involves three aspects:

1. File Design
2. Concurrency Control
3. File Replication.

File Design

File design deals with structure, attributes and operations file. OSI file design is based on the concept of a virtual file store. Virtual file store presents a standardised interface and provide a set of standardised operations. It enables a user to access and on files without having to know the internal machine structure etc.

Like other file systems, OSI virtual file store supports hierarchical file structures. Table 36.2 summarise the attributes of the OSI virtual file.

Table 36.2 Summary of the attributes of the OSI virtual file.

Attribute name	Data type
File name	String
Allowed operations	Bit Map
Access control	List
Account number	Integer
Date and time of the file creation	Time
Date and time of last file modification .	Time
Date and time of last file read	Time
Date and time of last attribute modification	Time
Owner	Userid
Identity of last file modifier	Userid
Identity of last attribute modifier	Userid
Identity of last reader	Userid
File availability	Boolen
Contents type	Objected
Encryption key	String
Size	Integer
Maximum future size	Integer
Legal qualifications	String
Private use	String

Most of the attributes names are self-explanatory.

The field, 'legal qualifications' is a string field starting some legal aspects regarding the information contained in the file, e.g. copyright.

The field 'private use' is again a descriptive field. It gives information that is not covered by the other attributes.

The standard operations of the OSI virtual file store are given in Table 36.3

Table 36.3 OSI Virtual File store Operations

File level	Attribute level	Record level
.Create.	.Read	.Locate
.Delete	.Modify	.Read
.Select		.Insert
.Deselect.		.Replace
.Open		.Extend (Append)
.Close		.Erase

The operations of OSI virtual file store are placed under three categories:

(1) File Level

(2) Attribute Level

(3) Record Level.

Concurrency Control

Concurrency control is handled using 'locks' of two kinds:

(1) Shared Locks

(2) Exclusive Locks.

Shared Locks are used for reading and Exclusive Locks are used for writing.

Locking may be provided at three levels:

(1) File level

(2) Sub-tree level of hierarchical structure

(3) Record level.

File Replication

It is important from the redundancy and failure points of view. A simple solution is to have a master file and treat all other copies as slaves. Updates are done only on the master file, which are then propagated to other slave files. This scheme fails if the master file is not available for some reason. A more sophisticated scheme is based on voting quorum. Here, a specified number of file servers are invoked simultaneously carrying out an operation like update or modify.

A major problem encountered in networks is the incompatibility between the user terminals and the computer systems. Unfortunately, attempts to standardise terminals have not met with success. OSI approach for solving the problem is to define virtual terminal. The virtual terminal presents a uniform interface network wide. The virtual terminal is essentially an abstract data structure that reflects the present display status which can be manipulated from the keyboard as well as the computer. The current contents of the data structure are displayed on the screen. Although there are a number of minor variations in terminal designs. Therefore, terminals are placed under three broad categories:

(1) Scroll Mode or Dumb Terminals

(2) Page Mode or Semi-Intelligent Terminals.

(3) Form mode or Intelligent Terminals.

Dumb Terminals

These are connected to the network via an interface unit known as Packet Assembler/ Deassembler (PAD). PAD provides a standard interface to the network on the one side and on the other side the RS-232-C serial interfaces to the terminals. Virtual terminal feature of OSI uses two models for interaction with intelligent terminals. These models are known as asynchronous and synchronous models. Asynchronous and synchronous models are shown in Fig.36.10 (a) and 36.10 (b), respectively.

Fig. 36.10 Virtual terminal data structure models.

Asynchronous Model

In this model, input and output can take place simultaneously

Synchronous Model

In this model, specific conventions are needed so that only one update can take place at a time; i.e., either the terminal processor or the computer can update the buffer but not both, as the buffer is common. The asynchronous' model is more popular. In both the models, replica of data structures exists at the terminal and computer ends.

Private Branch Exchanges (Pbx)

In the U.S.A. the term PBX refers generically to any switching system owned or leased by a business or organisation to provide both internal switching functions and access to the public network. Thus a PBX in U.S.A. may use either manual or automatic control. The term PABX, *i.e.* Private Automatic Branch Exchange is also used in the U.S.A., and particularly in other countries, to refer specifically to automatically controlled PBXs.

How a PBX works

A phone system in which you had a wire running between your office and that of everyone you would ever need to talk with by phone would be incredibly expensive and the redundancies in it would be insane.

Instead the phone system is made up of switches that provide you with a connection to another phone as needed. When you hang up, the phone lines you were using are available for others to use.

There is a lot of science that goes into figuring out how many lines are needed so that you, the customer, don't encounter the dreaded message, "All circuits are busy. Please try your call later."

At one level, this is what a Key System or Private Branch Exchange (PBX) is all about. Instead of having to have one phone line for everyone in your office, you might set it up so that you have three phone lines for eight phone extensions. Or you might have a four by sixteen configuration. A key system makes this possible because it can "switch" an available phone line to an extension wanting to make a call.

Pretty simple concept.

Toss an intelligent microprocessor, sufficient memory, some Digital Signal Processor (DSP) chips into the mix and you can do a lot more than just share phone lines, a minimum of at least 110 by last count of the marketing literature

The historical development of PBX systems has followed closely that of switches in the public networks. PBXs with computerized control became available in 1963 when AT&T's No.101 ESS was first installed. Since that time a large number of independent manufacturers have developed computer controlled PBXs. In fact the PBX market has long been one of the most competitive business in telecommunication.

The use of computer control for PBXs introduced numerous new features for users. Not only were customized calling features (e.g. abbreviated dialing) provided, but the numerous facilities for cost management also became available. Some of the more useful, commonplace features in a PBX are the following:

1. Accounting summaries by the individual employee or department
2. Multiple classes of service with priorities and access restrictions to area codes, WATS lines, and so on.
3. Least-cost routing to automatically select lie lines, foreign exchange circuits, WATS, DDD, and so forth.
4. Automatic call back when circuits are available.
5. Traffic monitoring and analysis to determine the utilisation of existing circuits or to ascertain blocking probabilities and network cost effectiveness.

Notes