Multi-Protocol Transport Services

NetBIOS, NetBEUI, TCP/IP,
NetBIOS over TCP/IP

Introduction

- Overview of OSI Reference Model
- Local Area Networking Concepts
- NetBIOS/NetBEUI overview
- TCP/IP overview
- NetBIOS over TCP/IP
- OS/2 TCPBEUI
- Windows Clients
- Troubleshooting

The OSI Reference Model

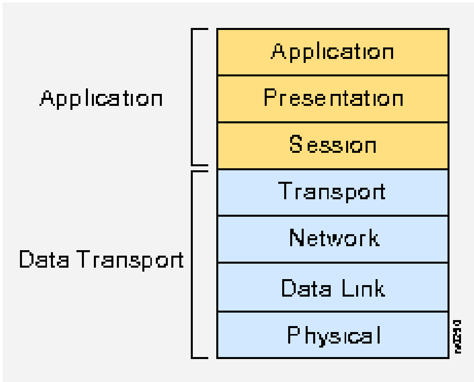
An Introduction

OSI reference model

- Developed by International Organization for Standardization (ISO) in 1984
- Conceptual model composed of seven layers
- Provides a conceptual framework for communication between computers
- Acutal communication is made possible by the usage of "communication protocols"
- Tasks assigned to each layer can be implemented independently

The seven layers of OSI model

- The seven layers of OSI model are shown below



The Physical layer

- Defines electrical, mechanical, procedural and functional specifications for activating, maintaining and deactivating physical link between systems
- Define characteristics such as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances and physical connectors used
- Examples: Ethernet, Token-Ring, FDDI, etc.

The Data Link layer

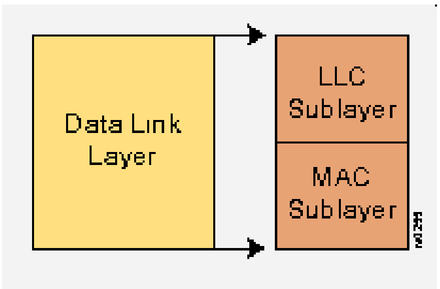
- Provides reliable transit of data across the network
- Data Link characteristics include:
  - Physical addressing: Defines how devices are addressed at the data link layer
  - Network topology: Specifies how devices are to be physically connected
  - Error notification: Alerting upper layers that a transmission error has occured

The Data Link layer (contd..)

  - Sequencing of frames: Reordering data frames that are transmitted out of sequence
  - Flow control: Moderates transmission of data so that the receiving device is not overwhelmed with more traffic than it can handle at one time

The Data Link layer (contd..)

- The Data Link layer is subdivided into
  - Logical Link Control (LLC) sublayer
  - Media Access Control (MAC) sublayer



The Logical Link Control sublayer

- Manages communication between devices over a single link of a network
- Defined in the IEEE 802.2 specification
- Supports both connectionless and connection-oriented services used by higher protocols
- Typically protocols that use the 802.2 LLC implementation are non-routable
- Example: LLC implementation in NetBEUI

The Media Access Control (MAC) sublayer

- Manages protocol access to the physical network medium
- MAC addresses are defined by IEEE MAC specification that allow multiple devices to uniquely identify one another in the data link layer
- Examples: Ethernet, Token-Ring, FDDI, Frame Relay, PPP, etc.

The Network layer

- Provides routing and related functions that allow multiple data links to be combined
- Supports both *connection-oriented* and *connectionless* service from higher layer protocols
- Examples: Internet Protocol (IP), NetBEUI, OSPF, etc.

The Transport layer

- Implements reliable data transport services transparent to the upper layers
- Transport layer functions include
  - Flow Control: Manages data transmission between devices so that the transmitting device does not send more data than the receiving device can process
  - Multiplexing: Allows data from several applications to be transmitted onto a single physical link

The Transport layer (contd..)

  - Virtual circuit management: VCs are established, maintained and terminated by the transport layer
  - Error checking and recovery: Takes care of detecting transmission errors. Error recovery (requesting data to be retransmitted) is handled by the transport layer
- Examples: TCP, NetBEUI

The Presentation layer

- Provides a variety of coding and conversion functions that are applied to application layer data
- Provides functions that ensure that information sent from the application layer will be readable by the application layer of another system.
- Not typically associated with a particular protocol stack

The Session layer

- Establishes, manages and terminates communication sessions between presentation layer entities.
- Communication sessions consist of service requests and service responses between applications.

The Application layer

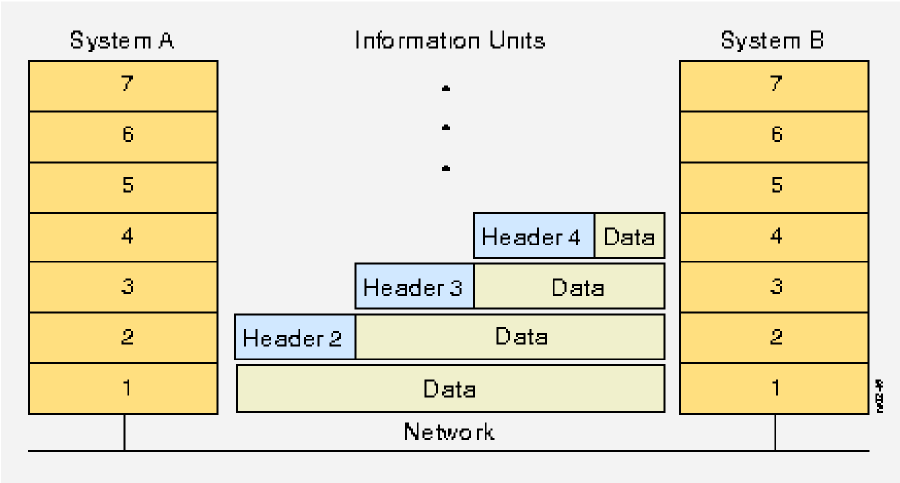
- Identifies communication partners
- Determines if sufficient network resources are available for communication
- Synchronization of communication is managed by the application layer
- Examples: File Transfer Protocol (FTP), Telnet etc.

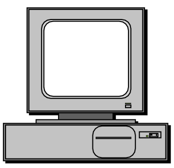
Basic elements involved in layered services

- Service user: The OSI layer requesting services from an adjacent OSI layer
- Service provider: The OSI layer that provides services to the service users
- Service Access Point (SAP): Conceptual location at which one OSI layer can request the services of another layer

Data encapsulation

- Encapsulation: Data unit at a given OSI layer potentially contains headers trailers and data from all the higher layers





Local Area Networking concepts

The workgroup and peer networks
The server and domain networks

The workgroup/peer network

- Group is dynamic
- All shares controlled by their users
- No master control/regulation over resources
- Highly flexible
- Low on security
- Example: IBM Peer for OS/2, Windows for Workgroups, Windows NT workstation

Advantages/disadvantages

- Advantages
  - Very flexible
  - Direct data exchange
  - Fast and simple to implement
  - All resources in the community can be used by everyone
- Disadvantages
  - Insufficient security
  - Resource sharing and access rights definitions complicated

The server/domain network

- All shared resources controlled by a server
- All access controls to critical resources centralized
- Administration is easy
- Does not have the flexibility of Peer networking
- Examples: OS/2 Warp Server network, Windows NT server network

Advantages/disadvantages

- Advantages
  - High secruity against unauthorized access
  - Management is easy
  - Centralized access control
- Disadvantages
  - Expensive than Peer networks
  - Skilled administrators required.

NetBIOS/NetBEUI overview

Network Basic Input/Output System (NetBIOS)

- Originally developed by IBM and Sytec Corporation
- NetBIOS is **not** a protocol but an Application Programming Interface
- NetBIOS needs a network protocol such as NetBEUI/TCPBEUI for its working
- It isolates the application program from the actual hardware used in the LAN
- Standardizes the interface for usage by application programs

Features of NetBIOS

- Computers using NetBIOS are known by their names
- These names must be unique in the network
- Communication is possible by
  - Establishing sessions (connection-oriented)
  - Using datagrams (connectionless)
- Defines standard format for commands called the Network Control Block (NCB)

Types of services provided by NetBIOS

- NetBIOS provides three types of services:
  - Name Service
  - Session Service
  - Datagram Service

NetBIOS Name Service

- Used to register/identify resources in NetBIOS
- NetBIOS names can be:
  - Unique: Owned by one and only one node
    - Example: a machine name
  - Group: Shared by many nodes that belong to a particular group
    - Example: a domain name
- Names registered by a node is maintained in the node's 'Name Table' until it is deregistered

NetBIOS Services

- Session Service
  - Provides a connection-oriented, reliable, full duplex message service
  - Destined for a specific node
- Datagram Service
  - Provides connectionless, unreliable, best effort service
  - Datagrams can be directed to a particular node or can be a broadcast

NetBIOS name characteristics

- All names are 16 bytes long
- The 16th byte specifies the functionality of the particular registered name
- A name can be *UNIQUE* name or *GROUP name* but not both
- Each node maintains a list of all names registered on it in a 'Name Table'
  - NBJDSTAT.EXE can be used on machines running NetBIOS over NetBEUI to dump the 'Name Table'

The NAME_NUMBER_1

- Every instance of NetBIOS configured on a machine has a NAME_NUMBER_1
- Consists of 10 bytes of ASCII '0' followed by the adapter's Universally Administered Address (UAA)
- This is a permanent name and cannot be deleted
- Used in 'STATUS_QUERY', 'STATUS_RESPONSE' and 'NAME_IN_CONFLICT' NetBEUI frames

The Network Control Block

- The NCB structure is shown below

Offset/# of bytes	Parameter Name
0/1	NCB_COMMAND
1/1	NCB_RETCODE
2/1	NCB_LSN
3/1	NCB_NUM
4/4	NCB_BUFFER@
8/2	NCB_LENGTH
10/16	NCB_CALLNAME
16/16	NCB_NAME
42/1	NCB_RTO
43/1	NCB_STO
44/2	NCB_POST@
46/2	NCB_DDID
48/1	NCB_ADAPTER_NUM
49/1	NCB_CMD_CMPL
50/14	NCB_RESERVE

NCB commands

- NCB.ADD.GROUP.NAME
- NCB.ADD.NAME
- NCB.CALL
- NCB.CANCEL
- NCB.CHAIN.SEND
- NCB.CHAIN.SEND.NO.ACK
- NCB.DELETE.NAME
- NCB.FIND.NAME

NCB commands *(continued)*

- NCB.HANG.UP
- NCB.LAN.STATUS.ALERT
- NCB.LISTEN
- NCB.RECEIVE
- NCB.RECEIVE.ANY
- NCB.RECEIVE.BROADCAST.DATAGRAM
- NCB.RECEIVE.DATAGRAM
- NCB.RESET

NCB commands *(continued)*

- NCB.SEND
- NCB.SEND.BROADCAST.DATAGRAM
- NCB.SEND.DATAGRAM
- NCB.SEND.NO.ACK
- NCB.SESSION.STATUS
- NCB.STATUS
- NCB.TRACE
- NCB.UNLINK

NetBIOS frames

- NetBIOS frames are of two types
  - Unnumbered Information (UI):
    - Used for Datagram communication
    - Frame contains calling and called names, hence frame size is 44 bytes
  - Information(I)
    - Used after a session is established
    - The LSN-RSN pair is used for identification of source and destination machines, hence frame size is 14 bytes

NetBIOS UI-frame format

- The NetBIOS UI frame format is shown below

0	2	4	5	6	8	12	28	44
NETBIOS Header Length	X'EFFF'	Command	Optional Data1	Optional Data2	Xmit/Resp Correlator	Dest Name	Source Name	

NetBIOS I-frame format

- NetBIOS I-frame format is shown below

0	2	4	5	6	8	12	13
NETBIOS Header Length	X'EFFF'	Command	Optional Data1	Optional Data2	Xmit/Resp Correlator	Dest Num	Source Num

14

NetBIOS functional address

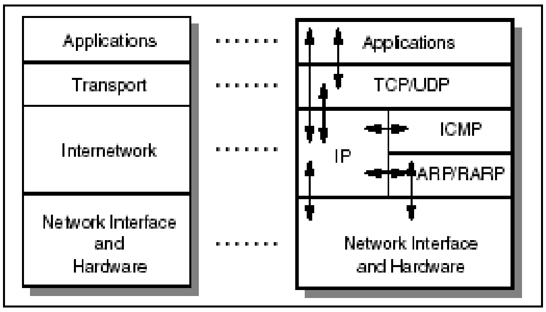
- The address C00000000080 is known as the NetBIOS functional address
- All machines in the 'NetBIOS Scope' must receive and process frames whose destination h/w address is the NetBIOS functional address
- Ethernet uses 030000000001 as the functional address

TCP/IP overview

Introduction to TCP/IP

- Layered protocol structure
- Named after two of its primary protocols
- Layered representation lead to the term *'protocol stack'*

TCP/IP architecture



The Application layer

- Provided for programs that use TCP/IP for communincation
- Interfaces with Transport layer using ports and sockets
- Examples: Telnet, FTP, DNS, etc.

The Transport layer

- Provides end-to-end data transfer
- Protocols in this layer provide
  - Connectionless service
    - eg., User Datagram Protocol (UDP)
  - Connection-oriented service
    - eg., Transmission Control Protocol (TCP)

The Internetwork layer

- Also called the internet/network layer
- Provides connectionless service
- Shields higher levels from network architecture below
- Provides the *routing* function
- Eg., Internet Protocol (IP), ICMP, IGMP, ARP, RARP

The Network Interface layer

- Also called the link/data link layer
- Is the actual interface to the network h/w
- TCP/IP does not specify any specific protocol at this level
- eg., IEEE 802.2, X.25, ATM, etc.

The client/server model

- Server
  - Application that offers service to internet users
  - Wait for requests on *well-known* ports
  - Receives request on the *well-known* ports
  - Performs the required service and sends back reply to clients

The client/server model (contd..)

- Client
  - Requester of a service
  - Uses an arbitrary port known as '*ephemeral port*' to contact the server
  - Sends service request to a *'well-known'* port on the server

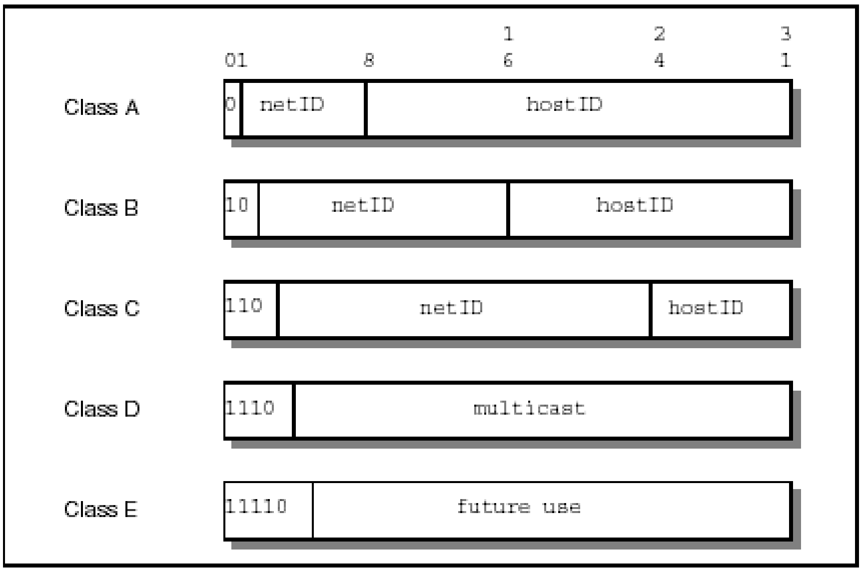
The Internet Protocol

- Provides unreliable, connectionless datagram delivery service
- Most protocols in the TCP/IP suite use IP datagrams for transmission
- Hides underlying details by creating a 'virtual network'

The IP address

- Used to uniquely identify a machine on the network
- Represented as an unsigned 32-bit value
- Represented in a dotted decimal format
- *IP address = <n/w number><host number>*

Class based IP addressing



Reserved IP addresses

- All bits 0
  - All bits 0 in host portion implies *'this'* host
  - All bits 0 in netwok portion implies *'this'* network
- All bits 1
  - Called directed broadcast
- Loopback
  - Is the class A n/w 127.0.0.0
  - Data to this n/w does not go on the wire

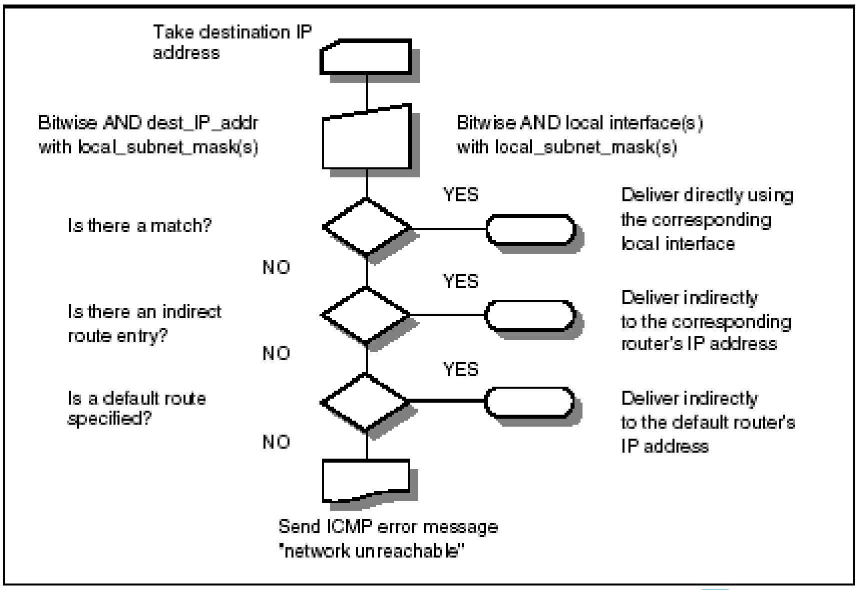
IP Subnets

- Created by dividing the host number
- Helps logically divide networks for easier administration
- Subnet mask
  - Specifies how many bits of the host number is used in the subnet
  - Made use of in routing

IP routing

- Provides mechanism to interconnect physical networks
- Routing function provided by a *'router'*
- Routing could be
  - Direct
    - *Destination on same physical n/w as source*
  - Indirect
    - *Destination and host on different physical networks*

The IP routing algorithm



Intranets

- Following ranges of addresses reserved for intranet use
  - *10.0.0.0*
    - *A Class A n/w*
  - *172.16.0.0 through 172.31.0.0*
    - *16 Class B n/ws*
  - *192.168.0.0 through 192.168.255.0*
    - *256 contiguous Class C n/ws*
- Routers should be configured *not* to forward packets from these addresses

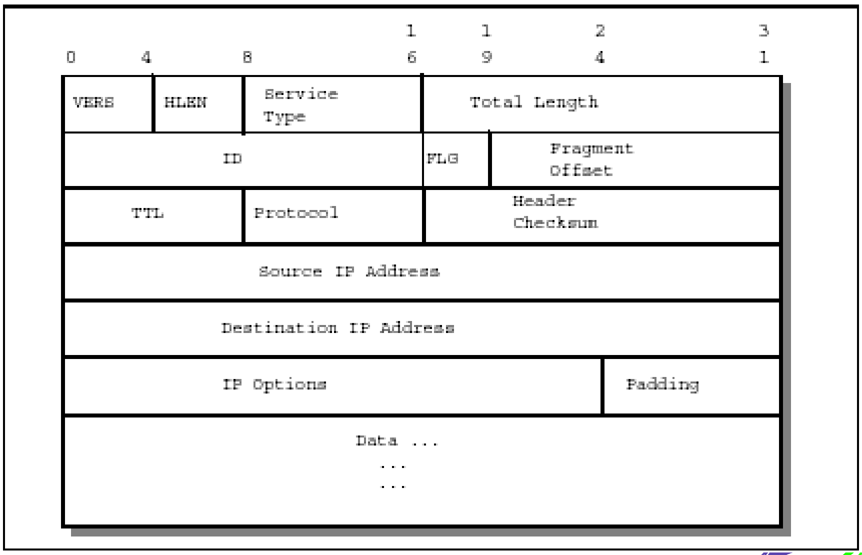
IP Broadcasting

- 3 primary types
  - Limited broadcast:
    - *255.255.255.255*
    - *Never forwarded by routers*
    - *Refers to all hosts on local subnet*
  - Net directed broadcast:
    - *<net id> <host id = all 1>*
    - *Used in unsubnetted environment*
    - *Routers must forward these messages*
    - *eg., 9.255.255.255*

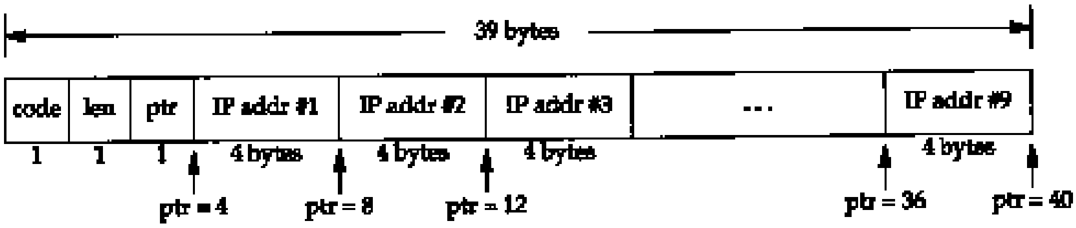
IP Broadcasting (contd..)

- Subnet directed broadcast:
  - *<net id><subnet id><host number = all 1s>*
  - *Refers to all hosts on the particular subnet*
  - *eg., 9.182.20.255 is the subnet directed broadcast address of a network with subnet mask 255.255.255.0*

The IP datagram



IP options



IP Fragmentation

- Needed when packet needs to traverse networks with different MTUs
- Steps to fragment a datagram
  - Check the fragmentation allowed flag
    - *If not allowed, send an ICMP error to originator*
  - If allowed, fragment datagram into 2 or more parts
  - Header is modified as follows
    - *More fragments bit set, except in the last*
    - *Offset field set in terms of 64bit values from 1st pkt*
    - *Total length = length of this fragment*

IP fragmentation (contd..)

- Fragments are reassembled only at the destination
- In case of loss of a fragment of a TCP segment, the entire segment is retransmitted

Transmission Control Protocol

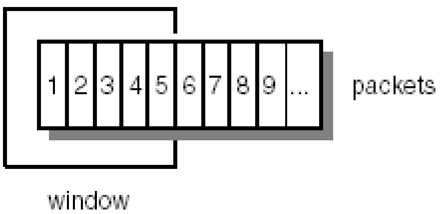
- Provides reliable logical circuit between pairs of processes
- TCP provides
  - Stream data transfer
    - *Forms groups data into segments*
  - Reliability
    - *TCP is an acknowledgement driven protocol*
  - Flow control
    - *Takes care that there are no buffer overruns*

TCP (contd..)

- Multiplexing
  - *Provided by the use of ports*
- Logical connections
  - *Status information for each stream is maintained*
  - *Each connection is identified uniquely by the pair of sockets*
- Full duplex
  - *TCP provides for concurrent data streams in both directions*

The Window principle

- For efficient utilization of network bandwidth, the window principle is used



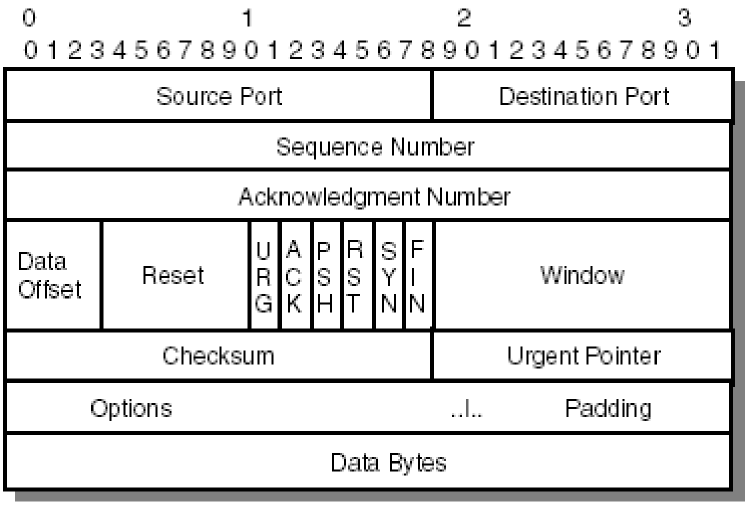
Advantages of Window principle

- Reliable transmission
- Better utilization of n/w bandwidth
- Flow control

Window principle applied to TCP

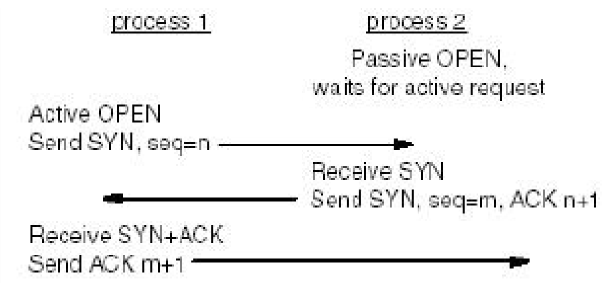
- Sequence numbers assigned at byte-level
- ACKs received will have byte-sequence numbers
- Window size is determined by the receiver
- Can vary dynamically during data transfer
  - *Each ACK will include the window size the receiver can deal with at that time*

TCP Segment format



TCP Connection establishment

- TCP connection is established by means of a *'3-way handshake'*



TCP Connection termination

- By means of a *'4 way handshake'*
- Both parties must explicitly close the connection
- The TCP half close
  - *Can occur when one end (A) is done sending data*
  - *The other end (B) can still send remaining data*
  - *Finally B has to send a FIN to close the session*

User Datagram Protocol (UDP)

- Is a datagram-oriented protocol
- Provides no reliability/flow control or error recovery
- Uses ports to multiplex/demultiplex data
- Normally UDP data is sent in a single IP datagram

UDP header

Source Port	Destination Port
Length	Checksum
Data...	

UDP applications

- TFTP
- DNS / NBNS
- SNMP
- NFS
- LDAP

NetBIOS over TCP/IP

NetBIOS over TCP/IP

- Implementation of NetBIOS to operate with TCP/IP as transport mechanism
- Formally standardized in RFCs 1001 and 1002
  - Provides for support services that make working of protocol more efficient
  - Categorizes nodes depending on type of operation

The RFCs 1001/1002

- RFC 1001
  - *Provides a general overview of the protocol*
  - *Gives emphasis on underlying ideas and techniques*
- RFC 1002
  - *Provides detailed specifications*
  - *Describes packet formats, protocol constants and variables*
  - *Provides pseudo-code for implementation*

Salient features

- Defines 3 classes of operation
  - *Broadcast node*
  - *Point-to-point node*
  - *Mixed node*
- Defines 2 support servers
  - *The NetBIOS Name Server (NBNS) node*
  - *The NetBIOS Datagram Distribution (NBDD) node*

The NetBIOS Name Server

- Essentially a *'bulletin board'* giving name-IP address mapping
- Used by P and M nodes
- Can work in secure/non-secure mode
- Is aligned with the Domain Name System (DNS), in addition provides
  - *Dynamic addition, updation and deletion of entries*
  - *Support for group names*

The NetBIOS Datagram Distribution Server

- Used by P and M nodes for datagram services
- End node may query NBDD if it will deliver datagram to a specified name
- WINS does not provide capabilities of a NBDD

RFC encoding of NetBIOS names

- Consists of two levels
  - 1st level
    - *Maps NetBIOS name into a Domain system name*
    - *Consists of NetBIOS name and scope id*
  - 2nd level
    - *Maps domain system name into 'compressed representation' required for DNS*

The NetBIOS Name service

- Name registration
- Name query
- Name release
- Name refresh
- Name challenge
- Name conflict
- NBNS WACK
- NBNS redirection
- Name caching

NetBIOS Session Service

- Session establishment
  - *Determine called name IP address*
  - *Establish a TCP connection*
  - *Send a Session Request*
- Steady state
- Session termination
  - *TCP connection termination*
  - *4 way handshake*

NetBIOS Datagram Service

- Used for services such as Browser announcements, Netlogon, etc.

OS/2 TCPBEUI

OS/2 TCPBEUI

- Implementation of RFCs 1001 and 1002 for the OS/2 operating system
- Provides support for multiple logical instances
- Provides support for multiple physical adapters (Software Choice release onwards)

Node types

- Broadcast/B-node
- Point-to-point/P-node
- Hybrid/H-node
  - Differs from the RFC M-node
    - Uses P-node operation for name registration and resoultion
    - If not successful, resorts to B-node style

Routing extensions in OS/2 TCPBEUI

- Three routing extensions are provided in OS/2 TCPBEUI for efficient working
  - *The Names file (RFCNAMES.LST)*
  - *The Broadcast file (RFCBCST.LST)*
  - *The Cache file (RFCCACHE.LST)*

RFCNAMES.LST

- Contains Name-IP address mapping
- Upto 2000 entries are supported
- Used before name lookup is done on the network
- *Important*
  - *All names MUST be entered in UPPERCASE*
  - *Should not be used in cases where name-ip mapping is dynamic*

RFCBCST.LST

- Can contain
  - IP addresses
  - Subnet broadcast addresses
  - Hostnames
- Upto 128 entries supported

RFCCACHE.LST

- Entries made by the stack in a format understood by it
- Cache lookup is done before sending a query packet on the network
- Should NOT be modified manually
- Given precedence over Names file and DNS lookup since it will be more *'current'*

Name registration

- Using BROADCASTS
  - Used by B-nodes
- Using the NBNS
  - Used by P/H-nodes

Name resolution

- Cache file lookup
- Names file lookup
- DNS lookup
  - *only if DOMAINSCOPE is configured*
- Query the network
  - B-nodes use broadcast method
  - P/H-nodes query the NBNS

DNS resolution

- Valid ONLY if DOMAINSCOPE configured
- Governed by ENABLEDNS parameter
  - If 0:
    - *Only encoded lookup is done*
  - If 1:
    - *Encoded first, if unsuccessful, unencoded lookup is done*
  - If 2:
    - *Unencoded first, if unsuccessful, encoded lookup is done*

TCPBEUI logical adapters

- Upto 4 logical instances is supported on 1 physical adapter
- CP levels provide support for TCPBEUI on multiple physical adapters with option for multiple logical instances on the physical adapters

Logical instances on same physical adapter

```
[NETBIOS]

DriverName = netbios$
ADAPTER0 = tcpbeui$,0
ADAPTER1 = tcpbeui$,1
ADAPTER2 = tcpbeui$,2
ADAPTER3 = tcpbeui$,3

[tcpbeui_nif]

DriverName = tcpbeui$
Bindings = IBMTOK_nif,IBMTOK_nif,IBMTOK_nif,IBMTOK_nif

[tcpip_nif]

DriverName = TCPIP$
Bindings = IBMTOK_nif
```

Multiple physical adapters

[NETBIOS]

DriverName = netbios$
ADAPTER0 = netbeui$,0
ADAPTER1 = tcpbeui$,1
ADAPTER2 = tcpbeui$,2
ADAPTER3 = tcpbeui$,3

[netbeui_nif]

DriverName = netbeui$
Bindings = IBMFEEO2_nif

Multiple physical adapters (contd..)

[tcpbeui_nif]

DriverName = tcpbeui$
Bindings = ,IBMFEEO2_nif,IBMFEEO2_nif,IBMTRP_nif
(lan0) (lan0) (lan1)

[tcpip_nif]

DriverName = TCPIP$
Bindings = IBMFEEO2_nif,IBMTRP_nif
(lan0) (lan1)

Multiple physical adapters (contd..)

- Number of TCP/IP LAN interfaces = Number of NIFs in the BINDINGS statement in tcpip_nif
- The BINDINGS statement in the [tcpbeui_nif] section is restricted to the set of NIFS that appear in the BINDINGS statement of the [tcpip_nif] section
- Number of TCPBEUI logical adapters = Number of NIFs in BINDINGS statement of tcpbeui_nif

Multiple physical adapters (contd..)

- Name query response from a multi-homed TCPBEUI system will contain IP addresses of all interfaces configured for TCPBEUI
- OS/2 systems from S/w Choice onwards can decode chained IP addresses in name query response
- Legacy systems of OS/2 can decode only 1st IP address from chain

Windows style of interaction

Windows clients

- Windows Domain Controllers register <domain_name> <1C> with WINS
- Windows clients query for this name while trying to logon
- OS/2 DCs do not register this name when server service is started
- Windows clients hence do not find the OS/2 DCs