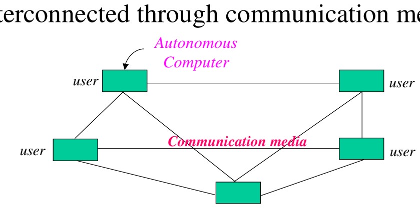
Computer Networks: Cheap, Fast, and Reliable Communication

Reference Books

- Andrew S. Tanenbaum, *"Computer Networks"*, 4th Edition, PHI
- Andrew S. Tanenbaum, *"Computer Networks"*, 3rd Edition, PHI
- W. Stallings, *"Data and Computer Communications"*, 8th Edition, PHI
- D. E. Comer, *"Internetworking with TCP/IP Vol. 1: Principles, Protocols, and Architecture"*, 5th Edition, PHI
- B. A. Forouzan, *"Data Communications and Networking"*, 4th Edition, Tata McGraw-Hill
- James F. Kurose, W. Ross, *"Computer Networking: A top-down approach featuring the Internet"*, 3rd Edition, Pearson
- M. Hassan, R. Jain, *"High Performance TCP/IP Networking"*, PHI
- L. Garcia, I. Widjaja, *"Communication Networks"*, 2nd Edition, TMH

Definition

- A collection of autonomous computers interconnected through communication media



- Autonomy: all computers are controlled by themselves
- Media: copper wire, fiber optics etc.

Why Networks

- Resource Sharing
  - *Software*
    - » Program, Database
  - *Hardware*
    - » Printer, Disk
- Load Balancing
  - ***Program in an overloaded computer transferred to and executed by other computer***

Why Networks

- High reliability
  - ***Effects of partial failure can be overcome by another computer***
- Location independence.
  - ***Users can access their files from anywhere in the network.***

Applications

- Railway Reservation Systems
- Digital Library
- Electronic Mail System
- Online Newspaper

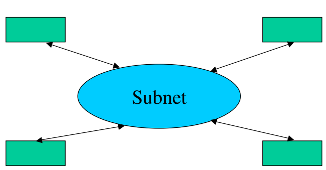
Before the Internet

- Postal network.
  - *Delivers different types of objects (letters, packages, etc.) world-wide.*
  - *Relatively high delay but relatively cheap.*
  - *Sender and receiver identified by their postal address (name, number, street, city, etc.).*
- Telephone network.
  - *Engineered to deliver real-time voice.*
  - *Also world-wide.*
  - *Low delay but more expensive.*
  - *Users identified but telephone number.*

Communication Subnet

- Connecting each pair of computers through dedicated communication media is expensive
- Individual computers bear the entire burden of managing communication with each of the other host
- Separate communication aspect of n/w from computation aspect of individual computers

Communication Subnet



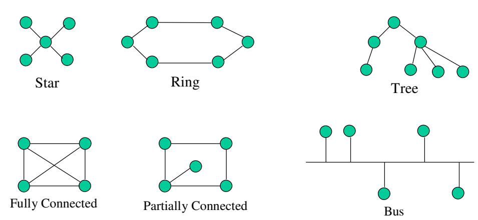
- Subnet consists of large number of communication links or channels and switching elements or nodes
- Nodes are properly interconnected

Function of nodes

- Interfacing with the hosts
- Some amount of processing on messages
- Switching messages from one link to another

Subnet Topology

1. ***Point-to-Point subnet***
2. ***Broadcast subnet***

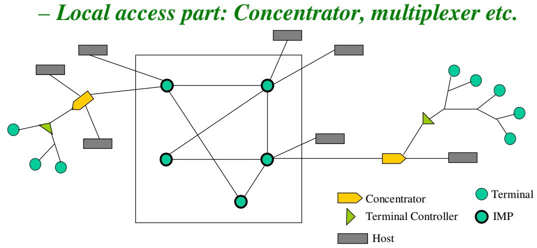


Subnet Topology

- Large number of hosts may be located far away from their nearest IMP
- Terminals, printers also need to be connected in the n/w
  - ***Do not have adequate speed or functional capabilities to directly access IMP efficiently***

Subnet Topology

- Design the subnet as two level hierarchy
  - *Backbone: consists of IMPS and high speed links*
  - *Local access part: Concentrator, multiplexer etc.*



Classification of networks

- Local Area Network (LAN)
  - **Small network, covers few kilometers, owned by a single organization**
- Metropolitan Area Network (MAN)
  - **Medium size network, covers entire city or part of it, owned by singe or multiple organization**
- Wide Area Network (WAN)
  - **Large network, covers number of countries, owned by multiple organization**

Traditional channel sharing techniques

» FDM

» TDM

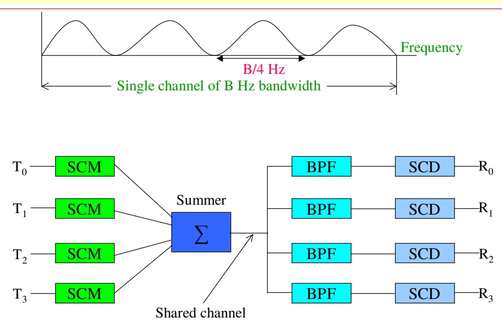
» Polling

» Concentrator

FDM

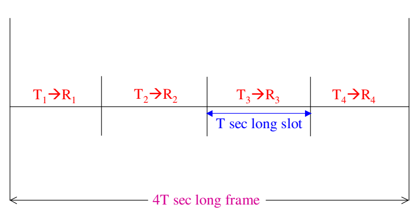
- Bandwidth of the channel is much higher than bandwidth of individual signal
- Partition the channel to create a number of logical channels
- Dedicate each logical channel to carry individual signal
- Example: Radio Broadcasting

FDM



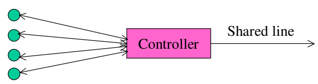
TDM

- Data rate is much higher than individual device
- Create a number of logical channels by time sharing the single available channel



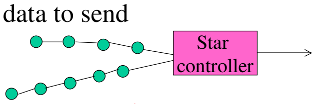
Polling

- A polling station sends a poll message to each station
- If the station has data to send, it starts sending data
- If not, sends poll reject message to controller

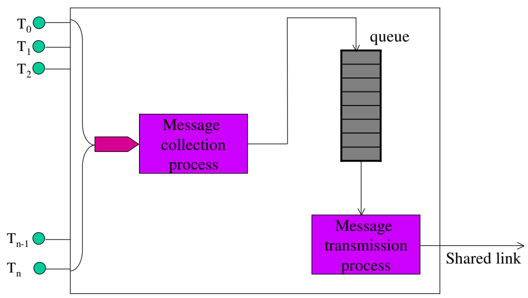


Hub Polling

- Stations participate in polling operation
- Controller polls the remote station
- If the station has data to send
  - **Sends data**
- If not
  - **Sends poll message to next station**
- Last station sends data or poll message back to the controller to terminate polling cycle



Concentrator or Statistical TDM



Concentrator or Statistical TDM

- Advantage over TDM
  - *High utilization of outgoing link*
  - *Supports more input devices than TDM*

- Disadvantage over TDM
  - *Increased design complexity*
  - *Occasional loss of messages owing to buffer overrun*
  - *Cost is increased for buffer*

Message transport across the subnet

- **Circuit Switching**
- **Message Switching**
- **Packet Switching**

Circuit Switching

- Traditional concept of circuits used in telephone system

- Three phases
  1. *Connection establishment*
     - **Connection request signal**
     - **Connection accepted signal**
     - **Path has been chosen (N.B. the chosen path is dedicated for that conversation)**
  2. *Data transfer*
  3. *Connection release: disconnection signal by any party*

Circuit Switching

- No store and forward delay at intermediate node
- Total delay=circuit setup + transmission time + propagation delay
- Example: Telephone system

Message Switching

- No path is established
- Host sends its message to the first IMP
- IMPs store the message, do some processing, and forward to the next IMP
- Message is transported across the subnet, one hop at a time

Message Switching

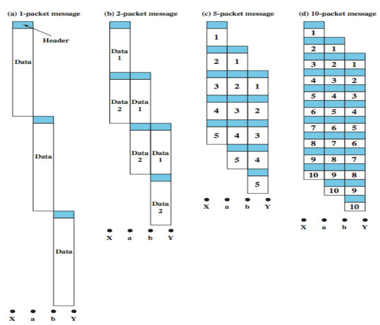
- Disadvantages:
  1. ***No limit on message size***
     - **Each IMP must have enough memory to store messages**
     - **Needs disk space**
     - **Cost increases**
  2. ***A single message may tie up a link for minutes***
- Example: Telegram

Packet Switching

- Works similar to message switching
- Places an upper bound on size of each block (called packet)
- Sender must break the message into several packets if size of message is larger than upper limit
- Send packets one by one
- Receiver must reassemble packets to generate original message
- Example: Fax

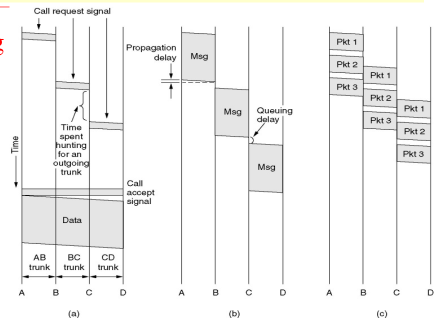
Packet Switching

- *Effect of packet size*
- Large packet → large delay
- Smaller packet → high overhead



Packet Switching

Event timing



(a) Circuit switching (b) Message switching (c) Packet switching

Packet Switching Vs Message Switching

- Advantages of packet switching:

1. ***Reduces store and forward delay at each IMP***
2. ***No disk space is needed at IMPs, stores packet in main memory itself***
3. ***Link is not tied up, making it suitable for interactive traffic***

Circuit Switching Vs Packet Switching

Circuit Switching	Packet Switching
Dedicated transmission path	No dedicated path
No store and forward delay	Store and forward delay at each IMP
Each message follow same route	Different route for different packet
Bandwidth wasted potentially	Unused bandwidth may be utilized by other packets from different sources

Layering

- Building complex systems is hard!
  - *Approach: "Divide and conquer"*
  - *Split job into smaller jobs, or layers*
- Analogy to other fields.
  - *Building a house: digging, foundation, framing, etc.*
  - *Car assembly line…*
- Basic idea: each step dependent on the previous step but does not need to be aware of how the previous step was done.

Analogy: Air Travel

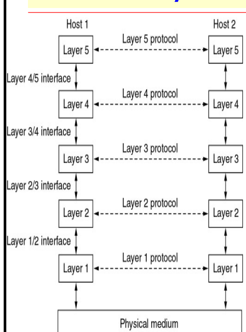
- Decomposed into series of steps:

Arrival at airport — Departure from airport
Check-in — Baggage claim
Boarding — Deplane
Takeoff — Landing
Traveling

Layered Architecture

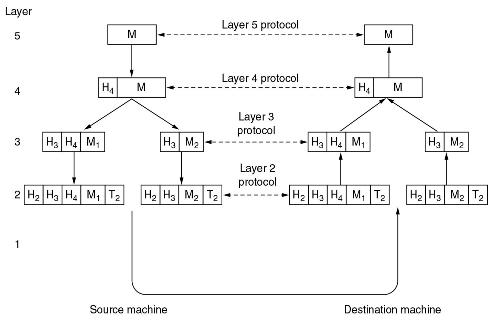
- **Layering model** is a solution to the problem of complexity in communication networks
- Each layer solves part of the communication problem
- Each layer is implemented independently
- Each layer provides a service to the layer above
  - *Relying on services provided by the layers below*

Layered Architecture



- Protocol: A set of rules and conventions governing the communication between two or more entities
- Interface: Defines a set of primitive operations through which each layer provides services to its immediate layer
- Peer process: Entities in a particular layer on different machine

Layered Architecture



The ISO OSI Model

- *ISO: International Standards Organization*
- *OSI: Open Systems Interconnection*

Application *Provides access to the OSI environment for users*
Presentation *Provides Independence from data representation*
Session *Establish, manage, terminate connection*
Transport *Connection oriented service*
Network *Routing, congestion control, connection oriented or connectionless*
Data Link *Framing, Synchronization, error control, flow control*
Physical *Transmission of raw bit stream over physical medium*

TCP/IP Reference Model

- TCP/IP: Transmission Control Protocol/Internet Protocol

	OSI	TCP/IP	
7	Application	Application	
6	Presentation		Not present in the model
5	Session		
4	Transport	Transport	
3	Network	Internet	
2	Data link	Host-to-network	
1	Physical		

Comparison of OSI and TCP/IP Model

OSI	TCP/IP
Stack of independent protocols	Stack of independent protocols
Layers above transport (inclusive) provides an end-to-end network independent service	Layers above transport (inclusive) provides an end-to-end network independent service
Seven layers	Four layers
Network layer may be connection-oriented or connectionless	Network layer is connectionless
Transport layer is connection-oriented	Transport layer may be connection-oriented or connectionless

Data Link Layer

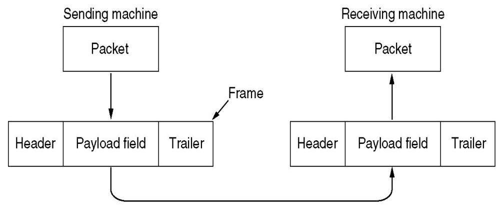
Functions

» Framing
» Error detection and correction
» Error control
» Flow control

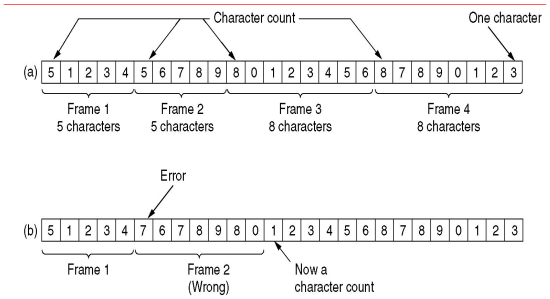
- Services provides to the network layer
  - ***Unacknowledged connectionless service***
  - ***Acknowledged connectionless service***
  - ***Acknowledged connection-oriented service***

Framing

- The process of encapsulating frames with special bit pattern, character pattern, encoding to let the receiver identify start and end of the frame

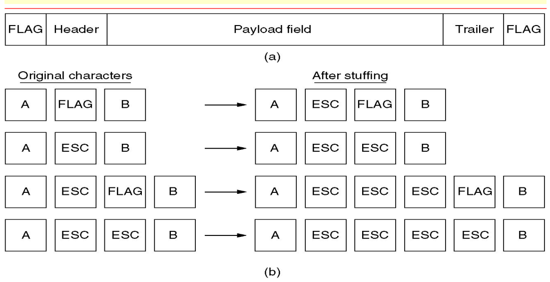


Character count



(a) Without errors (b) With one error

Character stuffing



(a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after stuffing.

Bit stuffing

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

- Bit stuffing (a) The original data (b) The data as they appear on the line (c) The data as they are stored in receiver's memory after destuffing.

Cyclic Redundancy Check (CRC)

- Bit strings are represented as polynomial with coefficient 0 and 1 only
  - **100101**
  - **$1.x^5+0.x^4+0.x^3+1.x^2+0.x^1+1.x^0$**
  - **X^5+x^2+x**
- Use of modulo-2 arithmetic
- Generator polynomial: Predetermined divisor
- Calculate CRC and append it to the end of the frame
- Resulting polynomial is divisible by Generator polynomial

Cyclic Redundancy Check (CRC)

- Algorithm:
  1. Let r be the degree of G(x). Append r zero to low-order-end of the frame ($x^rM(x)$), so it contains r+m bits
  2. Divide $x^rM(x)$ by G(x) using modulo-2 arithmetic
  3. Subtract the reminder from $x^rM(x)$ using modulo-2 subtraction
  4. Result is the checksumed frame to be transmitted (T(x))

Cyclic Redundancy Check (CRC)

- Example (Sender):
  - *M(x)=110011 (x^5+x^4+x+1)*
  - *G(x)=11001 (x^4+x^3+1)*
  - *Frame after 4 zero bits appended: 1100110000*

```
           100001
11001 | 1100110000
        11001
        -----
           10000
           11001
           -----
            1001
```

So, frame to be transmitted is 1100111001

Cyclic Redundancy Check (CRC)

- Example (Receiver):

```
             100001
11001 | 1100111001
        11001
        -----
             11001
             11001
             -----
             00000
```

So, no error.

Power of CRC

- Let the receiver receives T(x)+E(x)
  - *E(x)=error bit string*
  - *Each 1 bit in E(x) indicates a bit that has been inverted*
- Receiver performs (T(x)+E(x))/G(x)
  - *Gets E(x)/G(x)*
- If E(x)/G(x)=0
  - *Error not detected*
  - *Otherwise error will be detected*

Power of CRC

- Let $E(x)=x^i$, i indicates the position of the bit which is in error
  - *i.e. single bit error*
- If $G(x)$ contains two or more terms then $E(x)/G(x)\neq 0$
  - *i.e. all single bit errors will be detected*

Power of CRC

- Let there are two isolated single bit errors
  - *i,.e.* $E(x)=x^i+x^j$, $i>j$
    $=x^j(x^{i-j}+1)$
  - *Assume that x^j is not divisible by $G(x)$*
  - *Sufficient condition for all double bit errors to be detected => $G(x)$ does not divide $(x^{i-j}+1)$*

Power of CRC

- Now consider E(x) contains odd number of error bits
  - *E(x) contains odd number of terms* (x^6+x^2+1)
  - *No polynomial with odd number of terms has (x+1) as a factor*
  - *So, to detect odd number of errors G(x) must have a factor (x+1)*

Error Control

- Automatic Repeat Request (ARQ):
  - ***Detect errors at the receiving DLC layer and request the transmitting DLC layer to retransmit the erroneous frame***
    - **Purpose is to turn an unreliable link into an effective one**
    - **ARQ has four ingredients**
      - Error detection
      - Acknowledgement (ACK)
      - Negative Acknowledgement (NACK)
      - Timeout

Stop and Wait ARQ

- Transmitter transmits one frame at a time
- Wait for ACK/NACK from receiver
- If ACK, transmit next frame
- If NACK, retransmit the frame
- If no ACK/NACK within timeout period
  - ***Retransmit the frame***

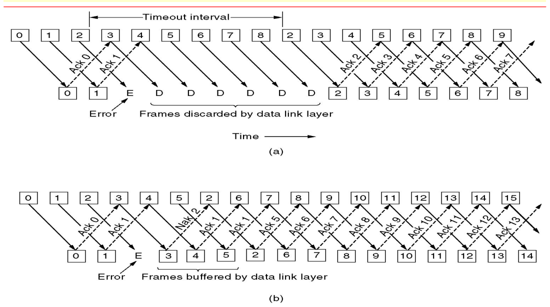
Go-Back-N ARQ

- Frames move continuously
- Transmitter maintains buffer storage
- If receiver detects error, it sends an NACK containing defective frame number N, discards subsequent frames
- Transmitter goes back to frame N and retransmits all frames starting from N
- Suitable when error rate is high

Selective Repeat ARQ

- Frames move continuously
- Transmitter maintains buffer storage
- If receiver detects an error, it sends an NACK containing defective frame number
- All subsequent frames are buffered by the receiver
- Transmitter transmits only the defective frame
- Suitable when error rate is low

Examples

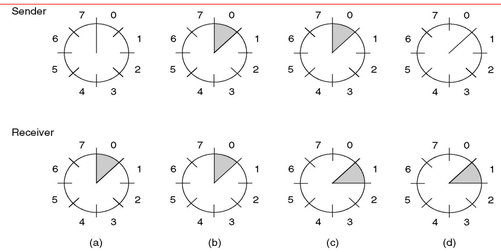


(a) Go-Back-n (b) Selective Repeat

Flow Control

- Sliding window protocol
  - ***Receiver has limited buffer space***
  - ***Number of frames the transmitter is allowed to transmit must be decided***
  - ***<u>Window size:</u> upper bound on the number of frames than can be transmitted by the transmitter***
  - ***To avoid frame loss due to buffer overflow***
    - **Number of outstanding frames <= window size**

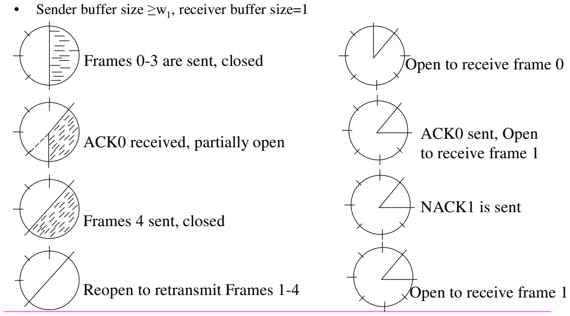
Sliding Window Protocol



A sliding window of size 1, with a 3-bit sequence number.
(a) Initially (b) After the first frame has been sent (c) After the first frame has been received (d) After the first acknowledgement has been received

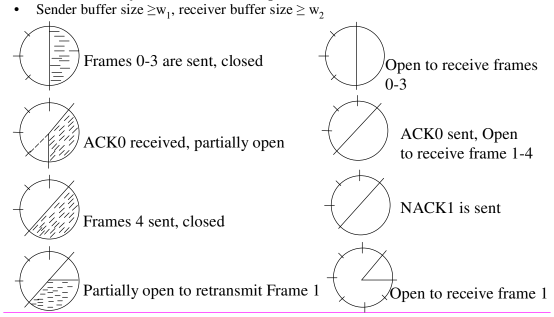
Sliding Window with Go-Back-N

- For sender $0 \le w_1 \le 2^n-1$, for receiver $w_2=1$
- Sender buffer size $\ge w_1$, receiver buffer size=1



Sliding Window with Selective Repeat

- For sender $0 \le w_1 \le 2^{n-1}$, for receiver $1 \le w_2 \le 2^{n-1}$
- Sender buffer size $\ge w_1$, receiver buffer size $\ge w_2$



Performance Analysis

- Efficiency is measured by link throughput
- Factor effecting efficiency of a protocol
  - ***Data transmission rate***
  - ***Frame size***
  - ***Frame overhead***
  - ***Link error rate***
  - ***Retransmission strategy***
  - ***Propagation delay***

Notations

- C: channel capacity
- D: number of data bits per frame
- H: number of header bits per frame
- F: total number of bits per frame=D+H
- A: number of bits in ACK/NACK
- I: propagation delay in seconds
- E: probability of a bit error
- P_1: Probability that a data frame is lost or damaged

Notations

- P_2: probability that an ACK frame is lost or damaged
- L: probability that a frame or its ACK is lost or damaged
- R: average number of retransmissions per data frame
- T: timeout interval in seconds
- W: window size
- U: channel utilization

Stop and Wait ARQ

- ***Case 1 (Ideal Channel)***
- Time required to transmit a frame=F/C Seconds
- Time required to transmit an ACK=A/C Seconds
- Total elapsed time between start of transmission of a frame and start of transmission of next frame
  t=(F/C)+I+(A/C)+I seconds
  =[(F/C)+(A/C)+2I] seconds

Stop and Wait ARQ

- ***Case 1 (Ideal Channel)***
- Number of data bits that could be transmitted in t seconds= C*t bits
  = (F+A+2CI) bits
- So, Channel Utilization U=(D/(F+A+2CI))
  =(D/(D+H+A+2CI))
  =(D/(D+O))

where O=H+A+2CI is the overhead

Stop and Wait ARQ

- ***Case 2 (Error prone channel):***
- Unsuccessful transmission uses (F+CT) bits of channel capacity
- So, total channel capacity used for one effective transmission B=R(F+CT)+(F+A+2CI) bits
- Probability of failure $L=1-(1-P_1)(1-P_2)$
- Probability that exactly K attempts are necessary for successful frame transmission $X=(1-L)L^{K-1}$

Stop and Wait ARQ

- ***Case 2 (Error prone channel):***
- Expectation of X=1/(1-L)
- So, expected number of retransmission per frame R=1/(1-L)-1=L/(1-L)
- Now B=(L/(1-L))(F+CT)+(F+A+2CI)
- Utilization U=D/B
  =D(1-L)/(LCT+F+(1-L)(A+2CI))

Stop and Wait ARQ

- ***Case 2 (Error prone channel):***
- Let T=2I+A/C then CT=A+2CI
- U=D(1-L)/(F+CT)
  =D(1-L)/(F(1+CT/F))
  =[D/D+H]*[(1-P1)(1-P2)]*[(1/(1+CT/F))]
  =U1*U2*U3

Sliding Window Protocol

- ***Step I (Error-free channel)***
- ***Case 1 <large window>:***
  - *One or more ACKs will come back before window is totally filled up*
  - *First ACK received after t=F/C+2I Sec*
  - *Number of frames that could be transmitted in t seconds=1+2CI/F*
  - *To ensure uninterrupted transmission W>(1+2CI/F)*
  - *So, U=D/(D+H), ignoring ACK*

Sliding Window Protocol

- *Case 2 <small window>*
  - *W<(1+2CI/F)*
  - *Sender receives first ACK after (F/C+2I) seconds*
  - *Number of data bits transmitted=W*D*
  - *So, U=(W*D)/(F+2CI)*

Sliding Window Protocol

- ***Step II (error prone channel)***
- *Case 1 <selective repeat>*
  - *Expected number of transmission per frame=1/(1-L)*
  - *For sliding window protocol=W/(1-L)*
  - *So, U=[D/(D+H)]*(1-L), large window*
    *=[(W*D)]/[(F+2CI)]*[(1-L)], small window*

Sliding Window Protocol

- Case 2 <Go-back-n>:
  - *Expectation of* $X = (1 - L)\sum_{i=1}^{\infty} f(i) L^{i-1}$
  - *Where f(i)=total number of frames transmitted if the original frame must be transmitted i times*
  - *f(i)=1+(i-1)*N=(1-N)+N*i*
  - *So, expectation of X=(1-L+NL)/(1-L)*
  - *For large window N=1+2CI/F*
  - *For small window N=W*
  - *U=[D*(1-L)]/[(D+H)*(1-L+NL)], large window*
    *=[(W*D)*(1-L)]/[(F+2CI)*(1-L+NL)], small window*

High-Level Data Link Control Protocol (HDLC)

- Uses bit stuffing for synchronization
- Uses sliding window with Go-back-N or selective repeat
- Defines three types of stations
  - *Primary station*
  - *Secondary station*
  - *Combined station*

High-Level Data Link Control Protocol (HDLC)

- Defines two link configuration
  - *Unbalanced configuration*
  - *Balanced configuration*
- Defines two data transfer modes
  - *Normal Response Mode (NRM)*
  - *Asynchronous Balanced Mode (ABM)*

High-Level Data Link Control Protocol (HDLC)

Bits	8	8	8	≥ 0	16	8
	0 1 1 1 1 1 1 0	Address	Control	Data	Checksum	0 1 1 1 1 1 1 0

- Three types of frame
  - *Information frame*
  - *Supervisory frame*
  - *Unnumbered frame*

High-Level Data Link Control Protocol (HDLC)

Bits	1	1	1	1	3
	1	3		1	3
(a)	0	Seq		P/F	Next
(b)	1	0	Type	P/F	Next
(c)	1	1	Type	P/F	Modifier

(a) Information frame (b) Supervisory frame (c) Unnumbered frame

Point-to-Point Protocol (PPP)

- ***PPP provides following services:***
  - ***Framing method***
    - Frame format handles error detection
  - ***Link Control Protocol (LCP)***
    - Bringing lines up, testing them, negotiating options, bringing them down
  - ***Authentication***
    - Defines how two devices authenticate each other
  - ***Network Control Protocol (NCP)***
    - Negotiating network layer options

Point-to-Point Protocol (PPP)

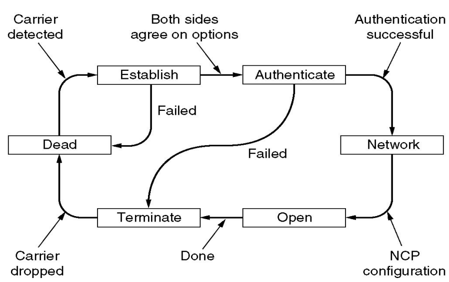
- ***Several things are missing:***
  - ***No flow control***
  - ***No error control***
    - Only detects error
  - ***No sequence numbering***
  - ***No sophisticated addressing mechanism for multipoint configuration***

Point-to-Point Protocol (PPP)

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

Frame format

Point-to-Point Protocol (PPP)



Transition phases

Point-to-Point Protocol (PPP)

- ***Authentication protocol:***
  - ***Password Authentication Protocol (PAP)***
    - User sends user name and password
    - System checks for validity of user name and password, accept/deny
  - ***Challenge Handshake Authentication Protocol (CHAP)***
    - Greater security than PAP
    - System sends a challenge to the user
    - User applies predefined function on challenge and password, sends the result to the system
    - System does the same
      - **If result matches, accept otherwise deny**

Local Area Network

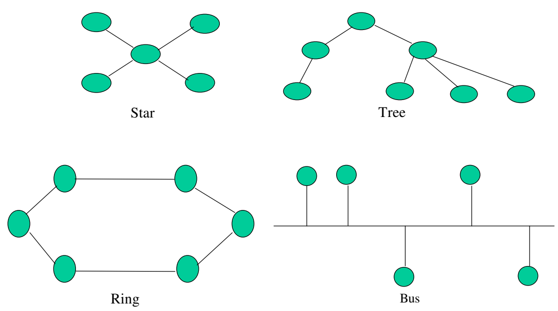
LAN overview

- Small networks, covers few kilometers
- Owned by single organization
- Shared medium
- Low propagation delay
- Low error rate

LAN overview

- Typically used to connect computers in office or factory to share resources and exchange information
- Traditional LANs operate at 10Mbps-100Mbps
- Newer LANs operate at up to 10Gbps

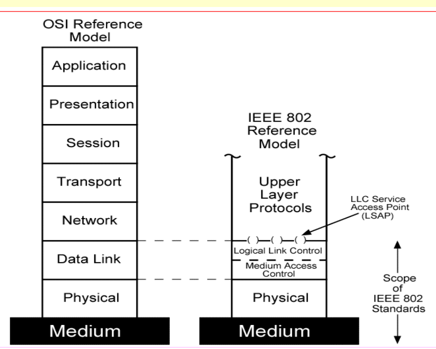
Topology



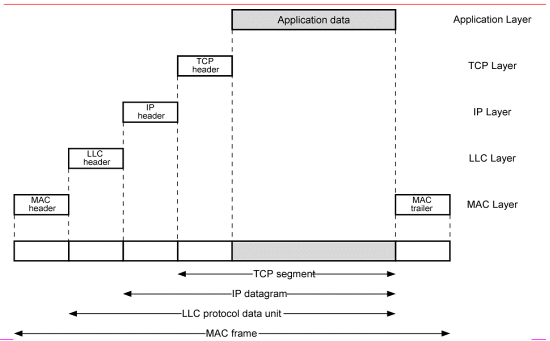
Protocol Architecture

- Lower layers of OSI model
- IEEE 802 reference model
- Physical
- Media Access Control (MAC)
- Logical Link Control (LLC)

IEEE 802 vs OSI



LAN Protocols in Context



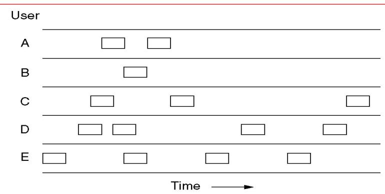
The Channel Allocation Problem

- Static channel allocation
  - ***No interference between users***
  - ***Efficient for small, fixed number of users***
  - ***Poses problem for large and continuously varying number of senders***
- Solution: Dynamic channel allocation

Dynamic Channel Allocation

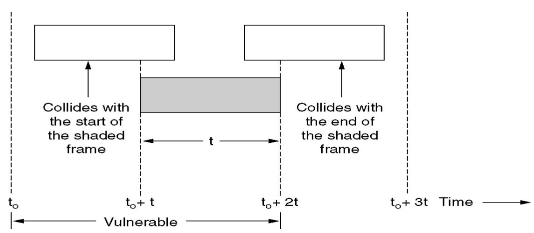
- Five key assumptions
  - ***Station model***
  - ***Single channel assumption***
  - ***Collision assumption***
  - ***Timing assumption***
    - Continuous time
    - Slotted time
  - ***Channel status testing***
    - Carrier sensing
    - No carrier sensing

Pure ALOHA



- Frames are transmitted at arbitrary time, Collisions possible
- Collisions are detected due to feedback property
- Wait random amount of time before retrying

Pure ALOHA



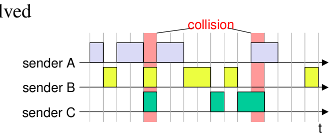
Vulnerable period for the shaded frame

Pure ALOHA

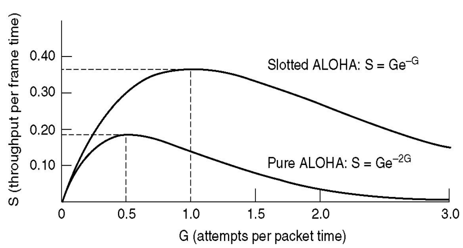
- Let G=offered load per frame time (mean)
- Probability that k frames are generated in a frame time is $P[k] = \frac{G^k e^{-G}}{k!}$
- So, $P[0] = e^{-G}$
- Mean number of frames generated in the vulnerable period=*2G*
- So, $P[0] = e^{-2G}$ (in collision zone)
- Throughput $S = GP[0] = Ge^{-2G}$
- S_{max}=*1/2e*, when *G=0.5*

Slotted ALOHA

- Divide time into discrete intervals
- Each interval corresponds to one frame time
- Collision zone is now halved
- So, $S = GP[0] = Ge^{-G}$
- S_{max}=*1/e*, when *G=1.0*
- throughput is double than Pure ALOHA



Throughput vs Offered Load



Carrier Sense Multiple Access Protocol (CSMA)

- First listen for clear medium (Carrier Sensing)
- If medium is idle
  - ***Transmit frame***
  - ***else wait***
- Two stations start at the same time
  - ***Collision***
  - ***Wait random amount of time***
- Maximum utilization depends on propagation time
  - ***Shorter propagation gives better utilization***

Non-Persistent CSMA

- If medium is idle, transmit
- If not, wait random amount of time
  - *Random delays reduces probability of collision*
  - *Capacity is wasted*
    - **Medium remains idle following end of transmission**
- If collision, wait random amount of time

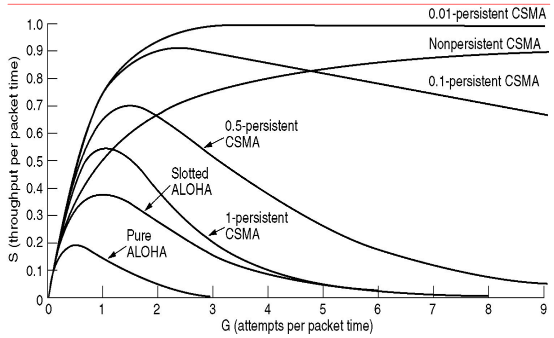
1-Persistent CSMA

- Stations wishing transmit listens and obeys following rules:
  1. *if medium is idle, transmit;*
  2. *If medium busy, listen until it becomes idle; then transmit immediately*
     - If two or more stations waiting, collision guaranteed
     - Gets sorted out waiting random time after collision

p-Persistent CSMA

- Reduces collision like non-persistent and reduce idle time like 1-persistent
- Rules
  1. *If medium is idle, transmit with probability p, and defer until next slot with probability 1-p*
  2. *If medium is busy, continue listening*
  3. *If collision, wait random amount of time.*

Throughput vs offered load

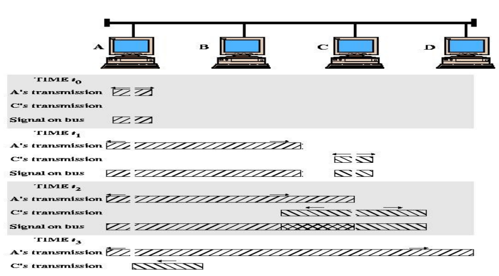


CSMA/CD

- With CSMA, collision occupies medium for duration of transmission
- Stations listen while transmitting

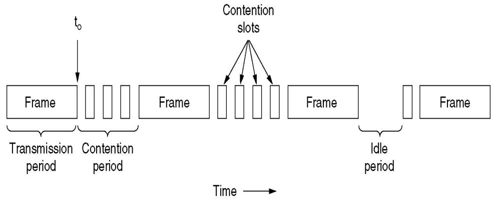
1. ***If medium is idle, transmit***
2. ***If busy, listen for idle, then transmit***
3. ***If collision, stop transmission immediately***
4. ***Backoff***

CSMA/CD



CSMA/CD Operation

CSMA/CD



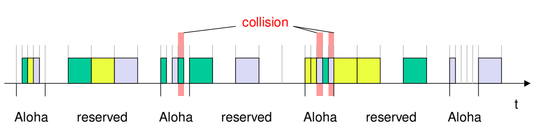
- A station must continue transmission after 2τ time where τ is the propagation time

Reservation Protocols

- Reservation can increase efficiency to 80%
  - *a sender* **reserves** *a future time-slot*
  - *sending within this reserved time-slot is possible without collision*
  - *reservation also causes higher delays*
  - *typical scheme for satellite links*
- Examples for reservation algorithms:
  - *Reservation-ALOHA*
  - *Reservation-TDMA*

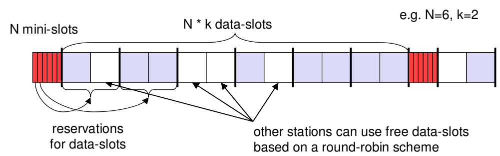
Reservation ALOHA

- ***two modes:***
  - **ALOHA mode** *for reservation: competition for small reservation slots, collisions possible*
  - **reserved mode** *for data transmission within successful reserved slots (no collisions possible)*



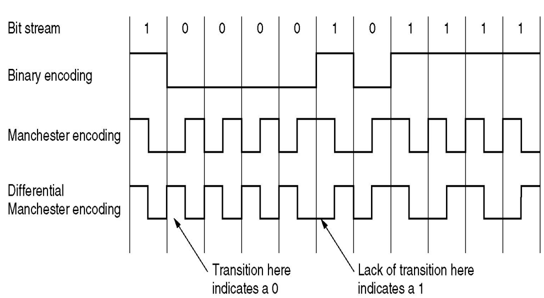
Reservation TDMA

- Reservation Time Division Multiple Access
  - *every frame consists of N mini-slots and x data-slots*
  - *every station has its own mini-slot and can reserve up to k data-slots using this mini-slot (i.e. x = N * k).*
  - *other stations can send data in unused data-slots according to a round-robin sending scheme (best-effort traffic)*



Ethernet: IEEE 802.3

Signal Encoding



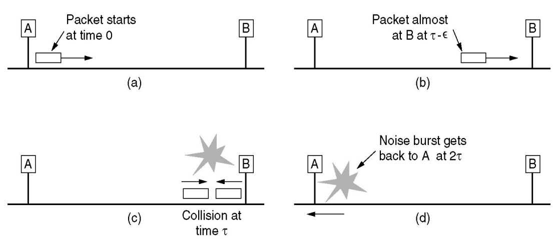
Ethernet MAC Sublayer Protocol

Bytes	8	6	6	2	0-1500	0-46	4
(a)	Preamble	Destination address	Source address	Type	Data	Pad	Check-sum

(b)	Preamble	SOF	Destination address	Source address	Length	Data	Pad	Check-sum

Frame formats. (a) DIX Ethernet, (b) IEEE 802.3

Ethernet MAC Sublayer Protocol



Collision detection can take as long as 2τ.

Binary Exponential Backoff

- After first collision, each station waits *0 or 1 time slots*
- After second collision, each station waits 0, 1, 2, or 3 time slots
- After third collision, each station picks random numbers from 0 to 2^3-1
- After i^{th} collision, each station picks random number from 0 to 2^i-1

Ethernet Performance

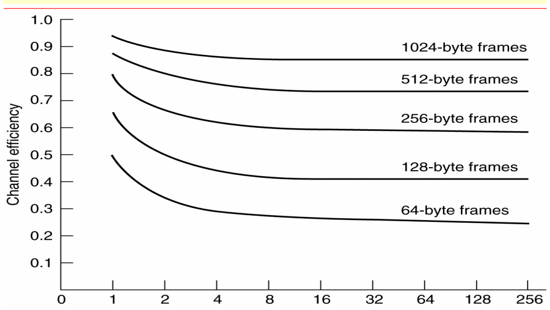
- K stations ready to transmit
- Mean frame time = T
- Prob. of transmission during a contention slot=p
- Prob. that some station acquire the channel in that slot $A = Kp(1-p)^{k-1}$
- Prob. That the contention interval has exactly j slots $= A(1-A)^{j-1}$

Ethernet Performance

- Mean number of slots per contention= $\sum_{j=0}^{\infty} jA(1-A)^{j-1}$
- Contention slot=2τ, mean contention interval=$2\tau/A$
- Efficiency= $\dfrac{T}{T + \dfrac{2\tau}{A}}$

$$= \frac{1}{1 + \dfrac{2BLe}{cF}}$$

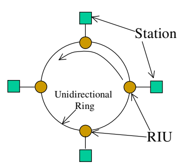
Ethernet Performance



Token Ring: IEEE 802.5

Token Ring Architecture

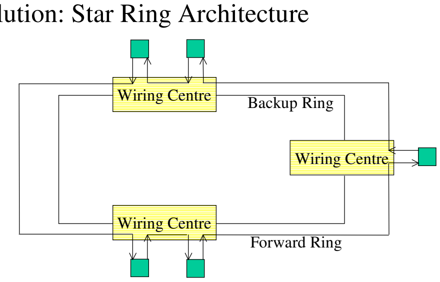
- A ring consists of a number of Ring Interface Units (RIU)
- Stations are connected to RIU
- Links are unidirectional
- point-to-point subnet
- 1 bit delay at each RIU



- Walk time=prop. Delay + 1 bit delay by each RIU

Token Ring Architecture

- Link failure → a serious problem
- Solution: Star Ring Architecture



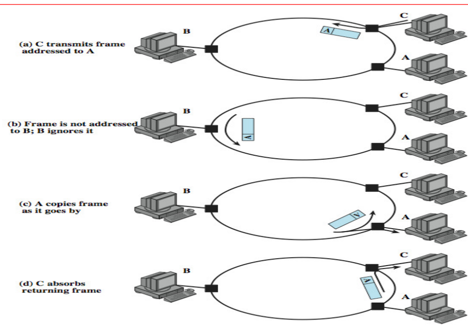
Token Ring MAC Protocol

- Small frame (token) circulates when idle
- Stations wait for token
- If a station seizes the token
  - ***Transmit data frames***
  - ***Frames are absorbed by transmitting station***
  - ***Transmitting station inserts new token***

Token Ring MAC Protocol

- Shielded twisted pair – DM encoding
- Maximum token holding time = 10 msec

Token Ring Operation



Frame Format

1	1	1
SD	AC	ED

Token Format

1	1	1	2 or 6	2 or 6	No limit	4	1	1
SD	AC	FC	Destination address	Source address	Data	Checksum	ED	FS

Frame Format

P	P	P	T	M	R	R	R

AC

F	F	rr	Control

FC

A	C	r	r	A	C	r	r

FS

Listen State

- Scan passing bit stream for patterns
  - ***Address of attached station***
  - ***Token permission to transmit***
- Copy incoming bits and send to attached station
- Modify bits as it passes
  - ***e.g. ACK***

Transmit State

- Station has data
- RIU has permission
- Transmit data
- May receive incoming bits
- Transmit token

Bypass State

- Signal propagate past repeater with no delay (other than prop. delay)
- Solution to reliability problem
- Improved performance

Token Ring Management

- One station is designated as Active Monitor (AM), others as Standby Monitor (SM)
- Functions of AM:
  - ***Ring Initialization:*** *PURGE MAC frame*
  - ***Lost Token:*** *Timer*
  - ***Orphan removal:*** *set M bit in AC byte*

AM recovery

- AM sends AMP frame periodically
- SM has two timers -- AMP/Token
- If timer expires
  - ***SM sends CLAIM TOKEN FRAME***
- Contention resolved by station address

BEACONING

- Used to isolate faulty station/ring
- Stations send BEACON frame giving predecessor's address
- Predecessor isolates & check itself
- If OK, switches to backup ring
- If not, bypass relay bypasses the station

Priority Management Scheme

- Real time data → high priority
- Priority schemes : P_T, P_D, P_{TR}
- Stations with $P_D \geq P_T$ ($PPP=P_T$) seizes the token
- Sends data frame ($PPP=P_D$)
- Regenerates token ($P_T \leftarrow P_{TR}$)
- Stations may makes reservation ($RRR \leftarrow P_D$)

Neighbor Notification

- AM sends AMP frame : A=0, C=0
- 1st station notes address (M) – sets A=1, C=1
- 1st station sends SMP frame : A=0, C=0
- 2nd station notes address (1) – sets A=1, C=1
- Finally AM notes address (N-1)

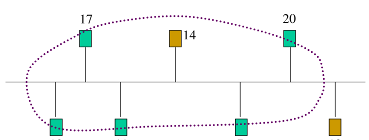
Token Bus: IEEE 802.4

Token Bus Evolution

- Problem with Ethernet
  - ***Probabilistic MAC protocol***
    - Worst case is unbounded
  - ***No priority Schemes***
    - Important frames are held up for unimportant frames
- Problem with Ring
  - ***Physical implementation***
    - Break in the cable !!
- Solution
  - ***Combine robustness of Ethernet and worst case behavior of ring***
    - Token Bus

Token Bus Architecture

- Stations are physically connected to a linear cable but Logically form a ring
- Each station knows address of left and right station



Token Bus MAC Sublayer Protocol

- Stations are added to the logical ring in order of addresses
- To transmit a station must have the token
- Transmit for token holding time
- Pass the token to successor explicitly
- Defines four priority classes 0, 2, 4 and 6
  - *Each type of traffic sent for a predefined duration*
  - *Lower priority frames can have the unused portion of high priority traffic*

Frame Format

Preamble	Start Delimiter	Frame Control	Destination Address	Source Address	Data	Checksum	End Delimiter
1	1	1	6	6	0-8174	4	1

Logical Ring Maintenance

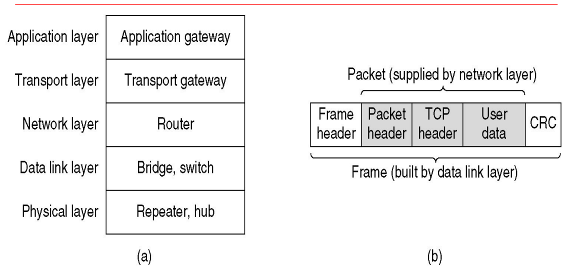
- Joining the ring
  - **Token holder sends SOLICIT_SUCCESSOR frame**
  - **The frame contains sender and its successors address**
  - **New station bids to enter within slot time (2τ)**
  - **If more than one station bits ----collision**
    - Resolved using RESOLVE_CONTENTION
- Leaving the ring
  - **Y has predecessor X and Successor Z**
  - **Y sends SET_SUCCESSOR frame to X**
  - **Z becomes successor of X**

Logical Ring Maintenance

- Ring initialization
  - **First station comes on-line**
  - **There is no traffic for certain duration**
  - **It sends CLAIM_TOKEN frame**
  - **If no competitor, it creates a token and sets up the ring containing itself**
- Detecting and Removing faulty station
  - **After passing the token if its successor does not sends frame or token, token is sent for second time**
  - **If it fails again---successor is down**
  - **Send WHO_FOLLOWS frame giving successor address**
  - **Failed stations successor replies with SET_SUCCESSOR frame**

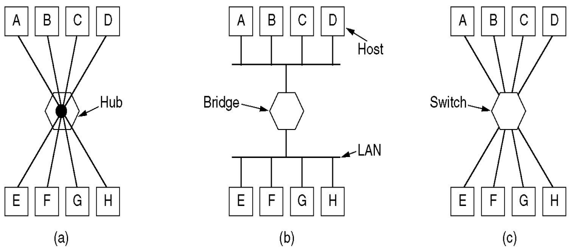
Local Internetworking

Internetworking Devices



(a) Which device is in which layer (b) Frames, packets, and headers.

Internetworking Devices

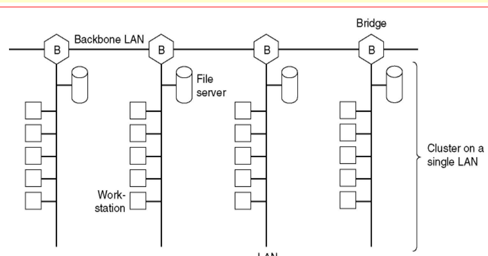


(a) A hub (b) A bridge (c) a switch

Bridges

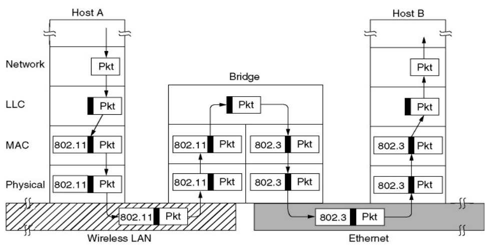
- Different department has different needs
  - *Different types of LANs*
  - *There is a need for interaction!!!*
- Departments may be geographically spread over several buildings
- Separate LANs to accommodate load
- Physical distance between two stations is large
- Question of reliability
  - *Defective node sends garbage continuously*
  - *Insert bridges at critical places; just like fire-doors*
- Organization's security
  - *LAN interfaces have a promiscuous mode*

Bridges



Multiple LANs connected by a backbone to handle a total load higher than the capacity of a single LAN

Bridges



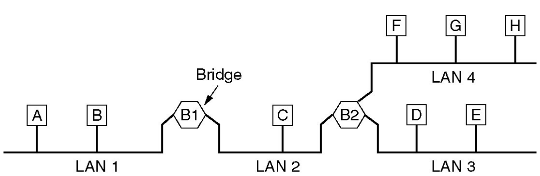
Operation of a LAN bridge from 802.11 to 802.3

Bridges

- Frame formats are different
- LANs do not run at same speed
  - *Timers in the higher layers*
- Different maximum frame length

Transparent Bridges

- Operates in promiscuous mode
- Forwarding decision is made with the help of a table
- Uses backward learning algorithm
  - *Look at the source address to decide which machine is on which LAN*

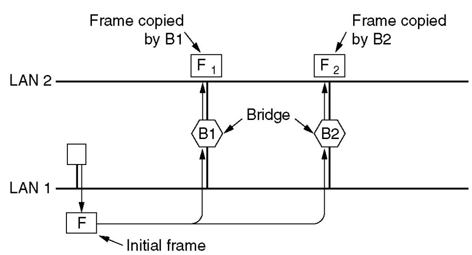


A configuration with four LANs and two bridges

Transparent Bridges

- Routing procedure
  - *If destination and source LANs are the same, discard the frame*
  - *If the destination and source LANs are different, forward the frame*
  - *If the destination LAN is unknown, use flooding*

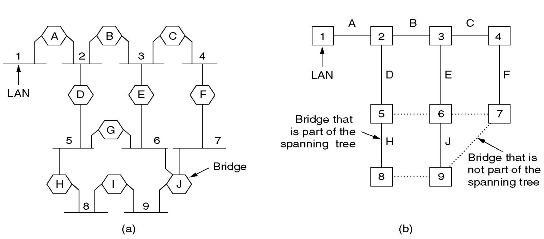
Transparent Bridges



Two parallel transparent bridges.

Spanning Tree Bridges

- Remove loops in the graph

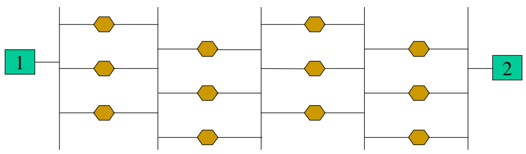


(a) Interconnected LANs (b) A spanning tree covering the LANs

Source Routing Bridge

- High order bit of source address = 1
- Frame header includes exact path
- Route is discovered using Discovery frame

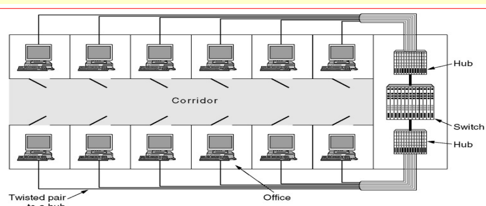
Frame Explosion Problem



Comparison of 802 Bridges

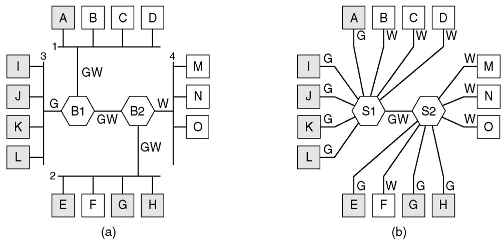
Issue	Transparent Bridge	Source Routing Bridge
Orientation	Connectionless	Connection-oriented
Transparency	Fully transparent	Not transparent
Configuration	Automatic	Manual
Locating	Backward Learning	Discovery Frames
Failure	Handled by bridges	Handled by hosts

Virtual LANs



- Issues for VLANs are
  - ***Security***
  - ***Traffic load***
  - ***broadcasting***

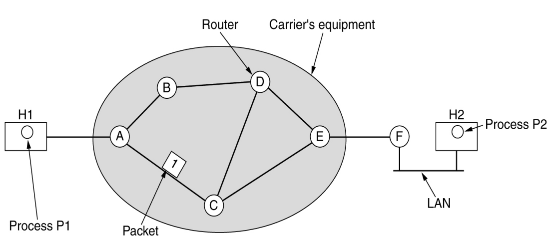
Virtual LANs



(a) Four physical LANs organized into two VLANs, gray and white, by two bridges. (b) The same 15 machines organized into two VLANs by switches

Network Layer

Design Issues



The environment of the network layer protocols

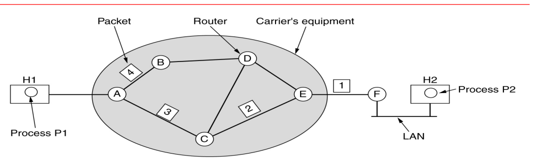
Design Issues

- Following goals must be satisfied while designing network layer services:
  - ***Independent of the subnet technology***
  - ***Transport layer must be shielded from the number, type and topology of the subnets present***
  - ***Network addresses should use a uniform numbering plan***

Design Issues

- Subnets can be organized as
  - ***Datagram Subnet***
    - **Connectionless, packets are called datagrams**
    - **Each packet contains full source and destination address**
  - ***Virtual Circuit subnet***
    - **Connection-oriented**
    - **Virtual circuits are numbered, routers must remember virtual circuits passing through it**
    - **Each packet must contain virtual circuit number**

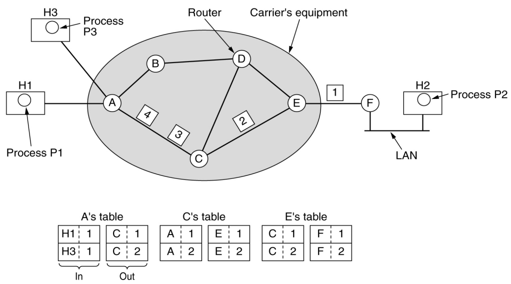
Design Issues



A's table initially (Dest.)	(Line)	A's table later (Dest.)	(Line)	C's table (Dest.)	(Line)	E's table (Dest.)	(Line)
A	–	A	–	A	A	A	C
B	B	B	B	B	A	B	D
C	C	C	C	C	–	C	C
D	B	D	B	D	D	D	D
E	C	E	B	E	E	E	–
F	C	F	B	F	E	F	F

Routing within a diagram subnet

Design Issues



Routing within a virtual-circuit subnet

Design Issues

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing

Routing Algorithms

- Properties:

  1. Correctness **2. Simplicity** **3. Robustness**
  4. Stability **5. Fairness** **6. Optimality**

- Types:
  - ***Static or Non-Adaptive Algorithm***
    - Routes are computed in advance, offline
    - Fails when topology changes frequently
  - ***Dynamic or Adaptive Algorithm***
    - Reflects frequent changes in topology

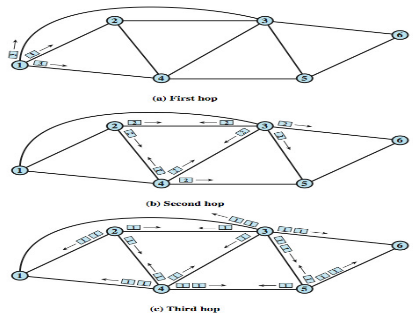
Shortest Path Routing

- Simple algorithm
- Basic idea:
  - **Build a graph of the subnet**
    - Node represents a router, arc represents a communication link
  - **Find the shortest path between nodes**
    - Dijkstra's algorithm
    - Metric: hops, delay

Flooding

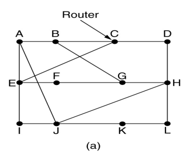
- Incoming packets retransmitted on every link except incoming link
- Eventually a number of copies will arrive at destination
  - ***Each packet is uniquely marked, so duplicates can be discarded***
  - ***Include a hop count in packets***

Flooding



Distance Vector Routing

- Routing tables are updated by exchanging routing tables and re-computing shortest path



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

(a) A subnet (b) Input from A, I, H, K, and the new routing table for J

Distance Vector Routing

- The algorithm suffers from ***Count-to-Infinity*** problem

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b)

Link State Routing

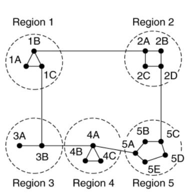
- Each router must do the following:
  1. ***Discover its neighbors, learn their network address***
  2. ***Measure the delay or cost to each of its neighbors***
  3. ***Construct a packet telling all it has just learned***
  4. ***Send this packet to all other routers***
  5. ***Compute the shortest path to every other router***

Hierarchical Routing

- With growing size of networks routing table grows proportionately
  - **Memory consumption becomes high**
  - **More CPU time needed to scan the table**
  - **More bandwidth required to exchange routing tables**

- Solution:
  - ***Divide routers into region***
    - Routers know the topology of its own region but knows nothing about topology of the other regions
  - ***For large networks divide regions into clusters, clusters into zones, zones into groups……..***

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

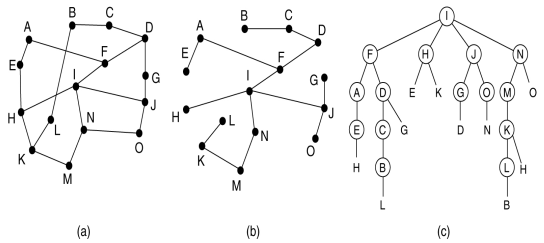
(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

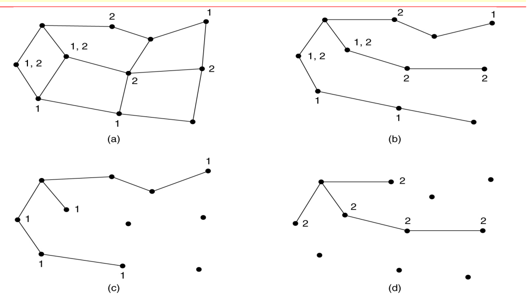
(c)

Broadcast Routing



Reverse path forwarding: (a) A subnet (b) a Sink tree (c) The tree built by reverse path forwarding

Multicast Routing

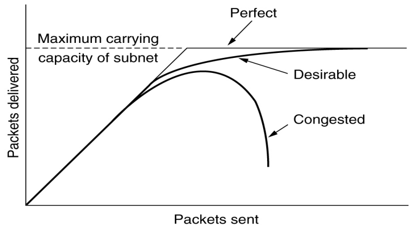


(a) A network (b) A spanning tree for the leftmost router (c) A multicast tree for group 1 (d) A multicast tree for group 2

Congestion Control

Definition

- Too many packets in the subnet than its carrying capacity results in ***congestion***



Factors Affecting Congestion

- Insufficient memory
  - ***Even infinite memory makes thing worse***
- Slow processor
  - ***Queue build up***
- Low bandwidth
- Congestion tends to feed upon itself

General Principles of Congestion Control

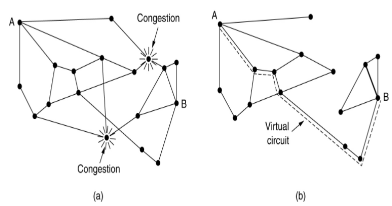
- Two approaches
  - ***Open loop***
    - **Make sure congestion does not occur**
  - ***Closed loop***
    - **Monitor the system**
      - detect when and where congestion occurs.
    - **Pass information to where action can be taken.**
      - Send a packet to the source
      - Load increased
    - **Adjust system operation to correct the problem.**

Congestion Prevention Policies

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Congestion Control in Virtual Circuit Subnet

- Admission control
- Allow new circuits avoiding problem area



- Negotiate an agreement
  - *Reserve resources*

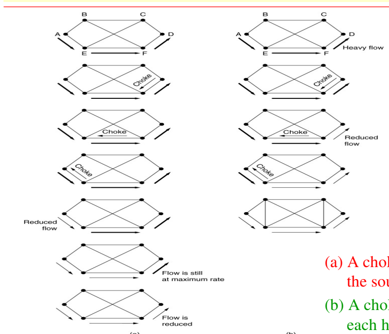
The Warning Bit

- Each router monitor utilization of outgoing line
- $u_{new} = a*u_{old} + (1-a)*f$
- If $u > threshold$
  - *The output line enters Warning state*
  - *Each newly arrived packets are then marked*
    - Turn on one bit in header
  - *Receiver copies this information into ACK*
  - *Sender cuts down transmission rate*

Choke Packets

- Tell the sender directly about congestion
- Congested router sends back choke packet
  - *Sender cuts down transmission rate for certain interval*
  - *If more choke packet received, rate is reduced further*
  - *If not, sender may increase the rate*

Hop-by-Hop Choke Packet



(a) A choke packet that affects only the source

(b) A choke packet that affects each hop it passes through

Load Shedding

- Congested router drops packet at random
- It can do better
  - *Which packet to discard?*
    - Depends on the application
    - FTP Vs Multimedia traffic
  - *Implement intelligent discard policy*
    - Sources are required to mark packets with priority
    - Low priority packets are discarded when congestion occur

Random Early Detection (RED)

- Take action before it is too late
- Each router maintain a running average of queue length
- If average queue length hits threshold it starts discarding packets randomly
  - *Works better when source assumes congestion for lost packets – reduce transmission rate*
  - *Not appropriate in wireless network*

Jitter Control

- Audio and video streaming require constant transit time
- Variation in packet arrival time is called jitter
- Compute the expected transit time for each hop
- Routers check to see how much the packet is behind or ahead of schedule
  - *If the packet is ahead of schedule, it is held up*
  - *If the packet is behind of schedule, forward immediately*

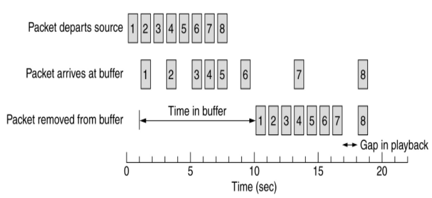
Quality of Service (QoS)

Requirements

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Buffering

- To smooth out jitter, buffer packets at destination
- Increases delay

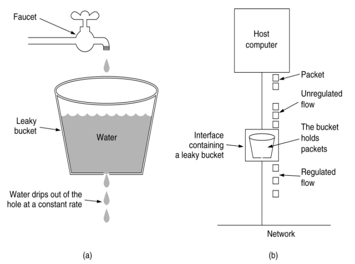


- Delay start of play as much as you can
  - ***Commercial sites buffer for 10 sec***

Traffic Shaping

- Irregular output from server may cause congestion
- Buffering is not possible for some application, e.g. video conferencing
- To achieve good QoS use traffic shaping
  - ***Regulate the average transmission rate***
  - ***Compare it with sliding window protocol***
  - ***Reduces congestion***

The Leaky Bucket Algorithm



- Host can produce data at rate *25MB/s*
- Routers can work best for rate *2MB/s*
- Data comes in *1MB* burst
- Use a leaky bucket with *ρ=2MB/s* and capacity *C=1MB*

(a) A leaky bucket with water
(b) A leaky bucket with packets

The Leaky Bucket Algorithm

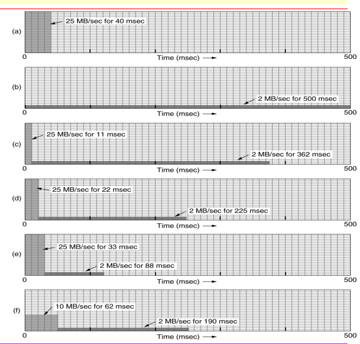
(a) Input to a leaky bucket

(b) Output from a leaky bucket
Output from a token bucket with capacities of
(c) 250 KB

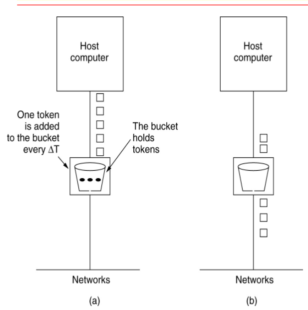
(d) 500 KB

(e) 750 KB

(f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket



The Token Bucket Algorithm



(a) Before (b) After

- Allows output to speedup when needed
- Leaky bucket holds token, one token generated every *ΔT sec*
- Allows saving up to maximum bucket size *n*

- Token bucket capacity =*C bytes*
- Burst length = *S sec*
- Token arrival rate = *ρ bytes/sec*
- Max. output rate = *M bytes/sec*
- Then we have

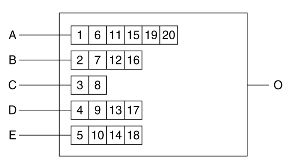
  C+ ρS=MS

Packet Scheduling

- Router handles multiple flows
  - *One flow will hog too much of its capacity*
- Processing packets in order of arrival
  - *Aggressive sender capture most of its capacity*
- Use ***fair queuing*** algorithm
  - *Routers have multiple queues for each output line, one for each flow*
  - *Scan the queues in round robin fashion*

Packet Scheduling

- Fair queuing gives more bandwidth to hosts with large packets
- Use byte-by-byte round robin



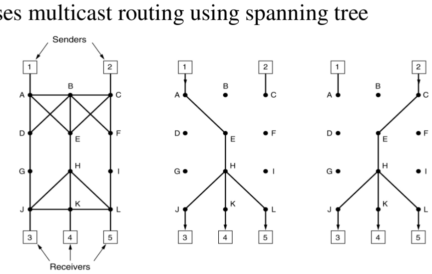
(a)

Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

Resource reSerVation Protocol (RSVP)

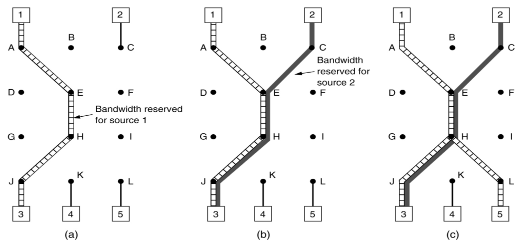
- Allows multiple senders to send to multiple receiver
- Uses multicast routing using spanning tree



(a) (b) (c)

Resource reSerVation Protocol (RSVP)

- For better reception receivers send reservation message to the sender
- Message is forwarded using reverse path forwarding
- Packets can flow without congestion



Internetworking

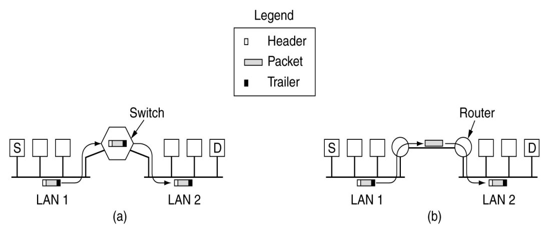
Why internetworking

- Can we use a single network all over the world?
  - **Many different network exists**
    - TCP/IP, SNA, Satellite, Cellular
    - Different protocols are in use
  - **Installation base of different networks is large and growing**
  - **Vendors do not want their customer to switch to another vendors system**
  - **Hardware development forces new software to be created**
    - Computer, Telephone, Television may be interconnected
    - You need different protocol

How networks differ

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

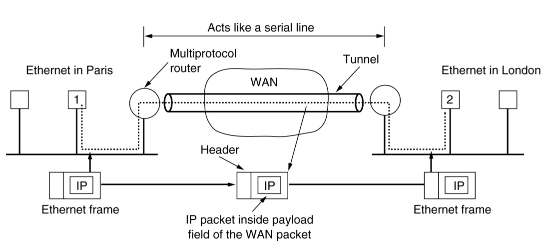
Connecting networks



(a) Two Ethernets connected by a switch (b) Two Ethernets connected by routers

Tunneling

- Encapsulating a packet inside another packet is called *tunneling*

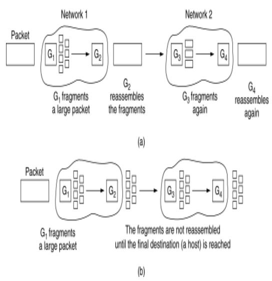


Fragmentation

- Break packets into smaller fragments
  - ***Each fragment is treated as a separate packet***
- Reasons for fragmentation
  - **Hardware**
  - **Operating system**
  - **Protocols**
  - **Desire to reduce error rate**
  - **Prevent a packet from occupying channel too long**

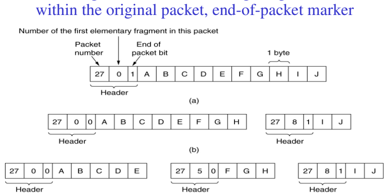
Fragmentation

- Two strategies:
  - ***Transparent fragmentation***
    - **Subsequent networks are not aware of fragmentation**
    - **All packets must exit through same gateway where they are reassembled**
    - **Overhead due to repeated fragmentation and reassembly**
  - ***Non-transparent fragmentation***
    - **Reassembly takes place only at the receiver**
    - **Overhead due to header for every packet**
    - **Overhead remains for rest of the journey**
    - **Multiple exit gateways can be used**



Fragmentation

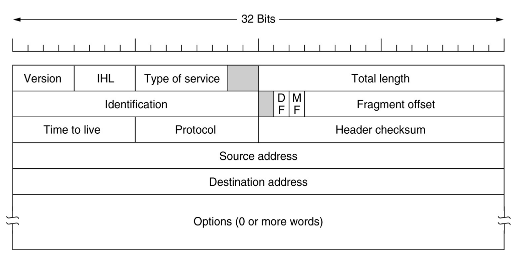
- Fragments must be number carefully so that original packet can be reconstructed
  - Each fragment must contain original packet number, offset within the original packet, end-of-packet marker



(a) Original packet, containing 10 data bytes (b) Fragments after passing through a network with maximum packet size of 8 bytes (c) Fragments after passing through a size 5 gateway

The Internet Protocol Version 4 (IPv4)

IPv4 Header



IP Options

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

IP Addresses

- IPv4 addresses are 32 bits long
  - Unique and universal
  - Contains two parts: network ID and Host ID
- Addresses are written in dotted decimal notation
  - **Each byte is converted into decimal separated by dots**
  - **Example:**
    - Address in binary: 10000000 00001011 00000011 00011111
    - Address in Dotted decimal notation: 128.11.3.31
- Address space is divided into five classes
  - **class A, class B, class C, class D, and class E**

IP Addresses

32 Bits

Class	Format			Range of host addresses
A	0	Network	Host	1.0.0.0 to 127.255.255.255
B	10	Network	Host	128.0.0.0 to 191.255.255.255
C	110	Network	Host	192.0.0.0 to 223.255.255.255
D	1110	Multicast address		224.0.0.0 to 239.255.255.255
E	1111	Reserved for future use		240.0.0.0 to 255.255.255.255

IP address formats

IP Addresses

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		This host
0 0 . . . 0 0	Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		Broadcast on the local network
Network	1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127	(Anything)	Loopback

Special IP addresses

IP Addresses

- *Private Addresses:* Some addresses are reserved for internal use
- Packet containing private addresses are forwarded by the routers in the Internet

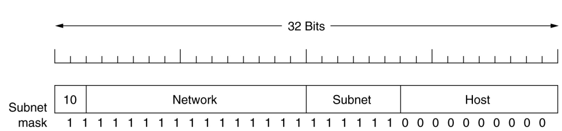
Addresses for Private Networks			
Range			Total
10.0.0.0	*to*	*10.255.255.255*	2^{24}
172.16.0.0	*to*	*172.31.255.255*	2^{20}
192.168.0.0	*to*	*192.168.255.255*	2^{16}

Subnets and Subnet Masks

- Allows a network within organization to split into several parts
  - ***Each part is called subnet***
  - ***host portion of address partitioned into subnet number and host number***
  - ***each LAN or subnet assigned subnet number***
  - ***subnet mask indicates which bits are subnet number and which are host number***
- The network looks to rest of Internet like single unit
- local routers route within the subnetted network

Subnets and Subnet Masks

- To get *subnet mask*, set all network bit and subnet bits to *1*

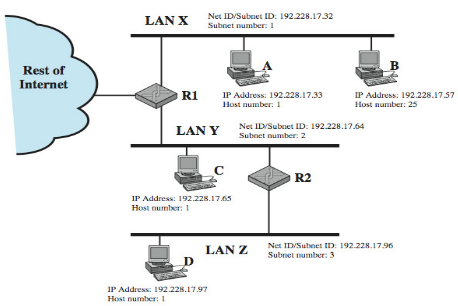


Subnets and Subnet Masks

- To route packet routers perform boolean AND operation of destination address with subnet mask

	Binary Representation	Dotted Decimal
IP address	11000000.11100100.00010001 .00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111 .11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001 .00100000	192.228.17.32
Subnet number	11000000.11100100.00010001 .001	1
Host number	00000000.00000000.00000000 .00011001	25

Subnets and Subnet Masks



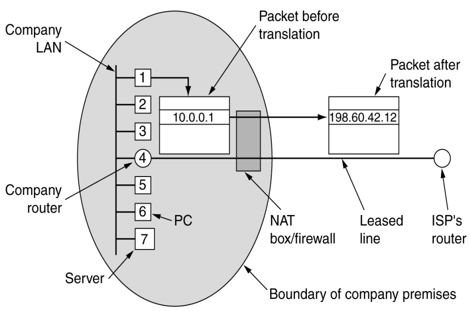
Network Address Translation (NAT)

- Question: how can a site provide multiple computers access to Internet services without assigning each computer a globally-valid IP address?
- Answer:
  - *Network Address Translation (NAT)*
    - **Extension to IP addressing**
    - **IP-level access to the Internet through a single IP address**
    - **Transparent to both ends**
    - **Implementation**
      - *Typically software*
      - *Usually installed in IP router*
      - *Special-purpose hardware for high speed*

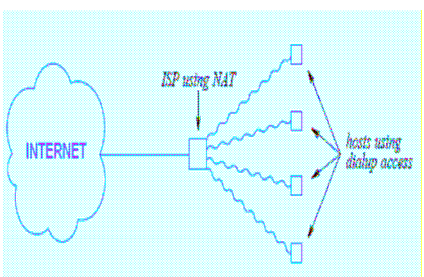
Network Address Translation (NAT)

- Organization
  - *Obtains one globally valid address per Internet connection*
  - *Assigns private addresses internally*
  - *Runs NAT software in router connecting to Internet*
- NAT
  - *Replaces source address in outgoing datagram*
  - *Replaces destination address in incoming datagram*
  - *Also handles higher layer protocols (e.g. TCP or UDP)*

Network Address Translation (NAT)



Network Address Translation (NAT)



Network Address Translation (NAT)

- Problems
  - *It violates the architectural model of IP*
  - *It changes the Internet to a kind of connection-oriented network*
  - *It violates the most fundamental rule of protocol layering*
  - *Applications are not required to use TCP/UDP*

Classless Inter-Domain Routing (CIDR)

- Addresses are granted in blocks irrespective of classes
- Restriction:
  - ***Addresses in block must be contiguous***
  - ***The number of addresses in a block must be a power of 2***
  - ***The first address must be divisible by the number of addresses***

First	205.16.37.32	11001101 00010000 00100101 00100000	16 Addresses
	205.16.37.33	11001101 00010000 00100101 00100001	
	.	.	
	.	.	
	.	.	
Last	205.16.37.47	11001101 00010000 00100101 00101111	

Classless Inter-Domain Routing (CIDR)

- IP addresses in a block is defined as x.y.z.t/n
  - **x.y.z.t is one of the addresses and /n is the mask**
  - **The address and the mask completely defines the whole block**
    - » First address, last address and number of address
- To get the first address, set rightmost 32-n bits to 0s
- To get the last address, set rightmost 32-n bits to 1s
- Number of addresses is 2^{32-n}

Classless Inter-Domain Routing (CIDR)

- Example:
- Suppose one of the address is 205.16.37.39/28
- In binary: 11001101 00010000 00100101 00100111
- First address: 11001101 00010000 00100101 00100000
  - » i.e. 205.16.37.32
- Last address: 11001101 00010000 00100101 00101111
  - » i.e. 205.16.37.47
- Number of addresses: $2^{32-28}=16$

Classless Inter-Domain Routing (CIDR)

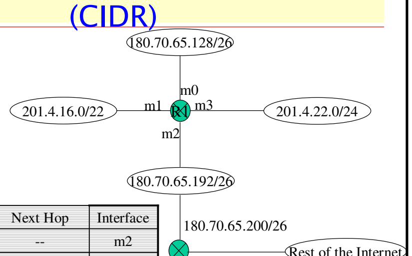
- **Example:** An ISP is granted a block of addresses starting with 190.100.0.0/16. The ISP needs to distribute these addresses to three groups as follows:
  - The first group has 64 customers; each needs 256 addresses
  - The second group has 128 customers; each needs 128 addresses
  - The third group has 128 customers; each needs 64 addresses

Group 1
Each customer needs 256 addresses
8 bits needed to define each host
Mask length 32-8=24
190.100.0.0/24 190.100.0.255/24
190.100.1.0/24 190.100.1.255/24
:
:
190.100.63.0/24 190.100.63.255/24

Group 2
Each customer needs 128 addresses
7 bits needed to define each host
Mask length 32-7=25
190.100.64.0/25 190.100.64.127/25
190.100.64.128/25 190.100.64.255/25
:
:
190.100.127.128/25 190.100.127.255/25

Classless Inter-Domain Routing (CIDR)

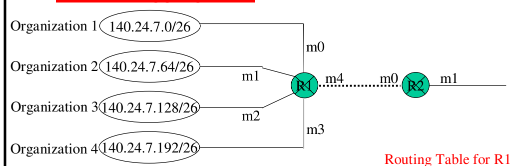
- Routing table is searched based on the network address
- To reduce the routing table size address aggregation is used



Mask	Network Address	Next Hop	Interface
/26	180.70.65.192	--	m2
/26	180.70.65.128	--	m0
/24	201.4.22.0	--	m3
/22	201.4.16.0	--	m1
Any	Any	180.70.65.200	m2

Classless Inter-Domain Routing (CIDR)

- ***Address Aggregation***



Routing Table for R1

Mask	Network Address	Next Hop	Interface
/26	140.24.7.0	---	m0
/26	140.24.7.64	---	m1
/26	140.24.7.128	---	m2
/26	140.24.7.192	---	m3
/0	0.0.0.0	Default	m4

Routing Table for R2

Mask	Network Address	Next Hop	Interface
/24	140.24.7.0	---	m0
/0	0.0.0.0	Default	m1

Internet Control Message Protocol (ICMP)

- IP provides best effort service
  - ***No error control mechanism***
  - ***What happens if something goes wrong?***
    - Destination is unreachable, time-to-live field has a zero value
  - ***No mechanism for management queries***
    - Determine if a router or host is alive
    - Network administrator needs information from another host or router
  - ***ICMP was designed to compensate these deficiencies***

Internet Control Message Protocol (ICMP)

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types

Internet Control Message Protocol (ICMP)

- Example debugging tools:
  - **Ping and traceroute**
- Ping:
  - ***Uses ICMP echo-request and echo-reply message***
  - ***Calculates RTT too.***
- Example:ping 172.16.15.8
  - PING 172.16.15.8 (172.16.15.8) 56(84) bytes of data.
  - 64 bytes from 172.16.15.8: icmp_seq=1 ttl=63 time=0.247 ms
  - 64 bytes from 172.16.15.8: icmp_seq=2 ttl=63 time=0.231 ms
  - 64 bytes from 172.16.15.8: icmp_seq=3 ttl=63 time=0.233 ms
  - 64 bytes from 172.16.15.8: icmp_seq=4 ttl=63 time=0.237 ms
  - 64 bytes from 172.16.15.8: icmp_seq=5 ttl=63 time=0.250 ms
  - 64 bytes from 172.16.15.8: icmp_seq=6 ttl=63 time=0.239 ms
  - --- 172.16.15.8 ping statistics ---
    - **6 packets transmitted, 6 received, 0% packet loss, time 4996ms**
    - **rtt min/avg/max/mdev = 0.231/0.239/0.250/0.016 ms**

Internet Control Message Protocol (ICMP)

- Traceroute:
  - ***Used to find a route to some destination***
  - ***Uses time exceeded and destination unreachable ICMP messages***
  - ***Time exceeded is used by routers and destination unreachable is used by receiving host (uses UDP port 1)***

Internet Control Message Protocol (ICMP)

- Example:
  - ***traceroute 172.16.15.8***

traceroute to 172.16.15.8 (172.16.15.8), 30 hops max, 38 byte packets

1 172.16.4.1 (172.16.4.1) 1.447 ms 1.347 ms 1.630 ms

2 172.16.15.8 (172.16.15.8) 0.232 ms 0.220 ms 0.277 ms

Address Resolution Protocol (ARP)

- Motivation
  - *Must use hardware (physical) addresses to communicate over network*
  - *Applications only use Internet addresses*
- Example
  - *Computers A and B on same network*
  - *Application on A generates packet for application on B*
  - *Protocol software on A must use B's hardware address when sending a packet*

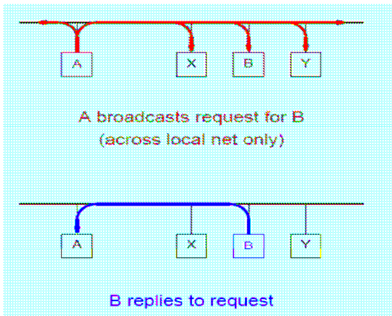
Address Resolution Protocol (ARP)

- Consequence
  - *Protocol software needs a mechanism that maps an IP address to equivalent hardware address*
  - *Known as* **address resolution** *problem*
- Standard for dynamic address resolution in the Internet
- Requires hardware broadcast
- Important idea: ARP only used to map addresses within a single physical network, never across multiple networks

Address Resolution Protocol (ARP)

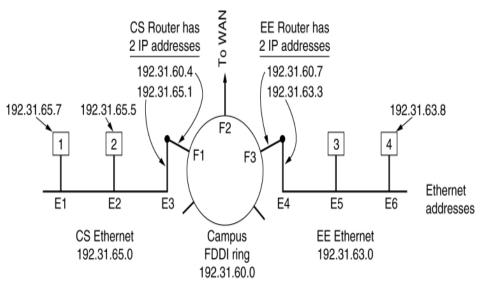
- Example:
  - *Machine A broadcasts ARP request with B's IP address*
  - *All machines on local net receive broadcast*
  - *Machine B replies with its physical address*
  - *Machine A adds B's address information to its table*
  - *Machine A delivers packet directly to B*

Address Resolution Protocol (ARP)



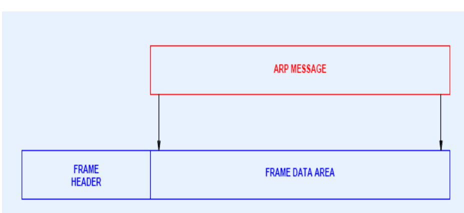
Address Resolution Protocol (ARP)

- Proxy ARP: An ARP that acts on behalf of a set of hosts



Address Resolution Protocol (ARP)

- ARP message travels in data portion of network frame
- We say ARP message is ***encapsulated***



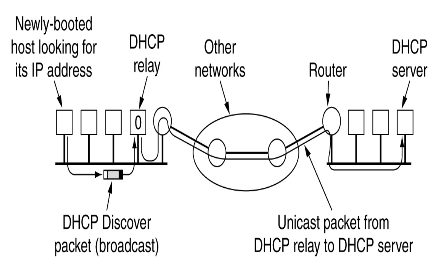
Address Resolution Protocol (ARP)

- Cannot afford to send ARP request for each packet
- Solution
  - *Maintain a table of bindings*
- Effect
  - *Use ARP one time, place results in table, and then send many packets*
- ARP table is a cache
- Entries time out and are removed
  - *Avoids stale bindings*
- Typical timeout: 20 minutes

Reverse Address Resolution Protocol (RARP)

- Maps Ethernet address to IP address
- Same packet format as ARP
- Intended for bootstrap
  - *Computer sends its Ethernet address*
  - *RARP server responds by sending computer's IP address*
- Disadvantage:
  - *RARP broadcast is not forwarded by the routers*
- Currently not used (replaced by DHCP)

Dynamic Host Configuration Protocol (DHCP)

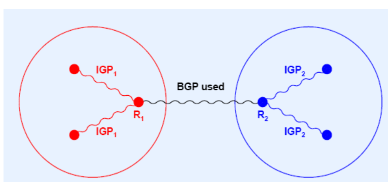


Routing Protocols in the Internet

- *Although it is desirable for routers to exchange routing information, it is impractical for all routers in an arbitrarily large internet to participate in a single routing update protocol.*
- Consequence: routers must be divided into groups
- Group of networks under one administrative authority is called Autonomous Systems (AS)
- Free to choose internal routing update mechanism
- Connects to one or more other autonomous systems

Routing Protocols in the Internet

- Two categories:
  - ***Interior Gateway Protocols***
    - RIP, OSPF
  - ***Exterior Gateway Protocols***
    - BGP



Routing Information protocol (RIP)

- Distance-vector protocol
  - ***Uses hop count metric***
- Uses split horizon and poison reverse techniques to solve inconsistencies
- Two Forms:
  - ***Active***
    - Form used by routers
    - Broadcasts routing updates periodically
  - ***Passive***
    - Form used by hosts
    - Does not send updates

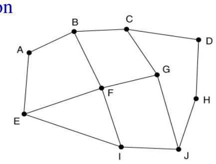
Open Shortest Path First (OSPF)

- Uses Link-State Routing algorithm
- More powerful than most predecessors

- Features
  - ***Type of service routing***
  - ***Load balancing across multiple paths***
  - ***Autonomous systems are partitioned into subsets called* areas**
    - Every AS has a backbone area (area 0)
    - All areas are connected to area 0
    - All area has an area border router

Border Gateway Protocol (BGP)

- The most popular (virtually the only) EGP in use in the Internet
- Allows two autonomous systems to communicate routing information
- Each AS designates a *border router* to speak on its behalf
- Uses Distance Vector Routing
- Sends path information



(a)

Information F receives from its neighbors about D

From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"

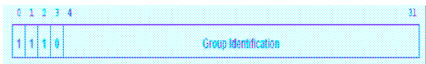
(b)

IP Multicasting

- Group address: each multicast group assigned an unique class D address
  - ***Up to 2^{28} simultaneous multicast groups***
- Dynamic group membership
  - ***Host can join or leave at any time***
- Uses hardware multicast where available
  - ***If not, use tunneling***
- Best-effort delivery semantics (same as IP)
- Arbitrary sender (does not need to be a group member)

IP Multicasting

- Class D addresses reserved for multicast
  - *224.0.0.0 through 239.255.255.255*
- General form:



Bhaskar Sardar, Information Technology Department, *Jadavpur University, India*

IP Multicasting

- Mapping IP Multicasting address to hardware multicasting address
  - ***Place low-order 23 bits of IP multicast address in low-order 23 bits of the special Ethernet address:***
    - **First 25 bits are 00000001 00000000 01011110 0**
    - **01.00.5E.00.00.00$_{16}$**
  - ***Example IP multicast address 224.0.0.2 becomes Ethernet multicast address***
    - **01.00.5E.00.00.02$_{16}$**
  - ***Example IP multicast address 238.212.24.9 becomes Ethernet multicast address***
    - **01:00:5E:54:18:09**

Bhaskar Sardar, Information Technology Department, *Jadavpur University, India*

Internet Group Management protocol (IGMP)

- Allows host to register participation in a group
- Four types of messages
  - ***General Query, Special Query, Membership Report, Leave Report***
- Message Format

Type	Maximum Response Time	Checksum
Group address (all 0's in general query)		

Bhaskar Sardar, Information Technology Department, *Jadavpur University, India*

Internet Group Management protocol (IGMP)

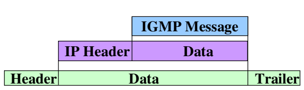
- Joining a group:
  - ***Hosts maintain a list group of membership***
  - ***When a process wants to join a group***
    - **It sends request to host**
    - **Host adds the process to to the requested group**
    - **If this is the first process in the group, host sends a *membership report message***
    - **If not, no need to send *membership report message***

Internet Group Management protocol (IGMP)

- Leaving a group
  - ***Host sees that no process is interested in a group***
  - ***Host sends a leave report message***
  - ***Router sends a special query message to the group***
  - ***If no response, router purges the group and sends leave report***
- Monitoring membership
  - ***What happens if a host is removed from the network (this the only host in the group)***
    - **Router do not receive leave report**
  - ***Router periodically sends a general query message***
  - ***Hosts reply with membership report (may be delayed response)***

Internet Group Management protocol (IGMP)

- Encapsulation
  - ***IGMP packet is encapsulated in an IP packet***
  - ***Value of protocol field is 2***
  - ***TTL must be 1***



Type	*Destination IP address*
Query	224.0.0.1 all system on this subnet
Membership Report	The multicast address of the group
Leave Report	224.0.0.2 all routers on this subnet

The Internet Protocol Version 6 (IPv6)

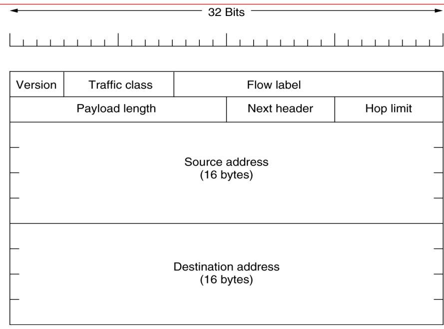
Deficiencies of IPv4

- Address depletion is a long term problem
- Lack of support for real-time audio and video transmission
- Lack of security for some applications

Advantages of IPv6 over IPv4

- Longer address space
  - **128 bit address Vs 32 bit address**
- Simplified header format
  - **8 fields Vs 14 fields**
- Better support for options
  - **Simplifies and speeds up the routing process**
- Support for security
  - **Security is an integral part of IPv6**
- Support for resource allocation
  - **Allows special treatment for some packets**

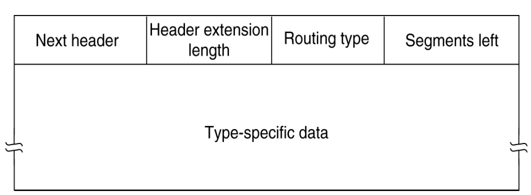
IPv6 Main Header



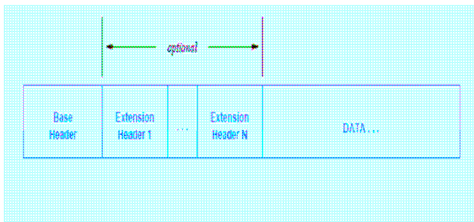
Extension Header

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

Extension Header for Routing



General form of IPv6 Datagram

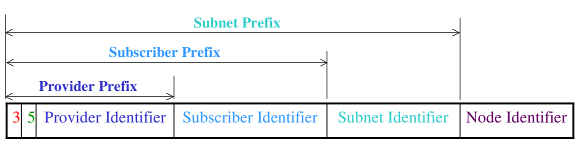


IPv6 Addresses

- Uses Hexadecimal Colon notation
- Example: dotted decimal notation
  104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255
- Becomes
  68E6:8C64:FFFF:FFFF:0000:1180:096A:FFFF
- Successive zeroes are indicated by a pair of colons
  - **Example: FF05:0000:0000:0000:0000:0000:0000:00B3**
  - **Becomes FF05::B3**
- IPv4 addresses can be written as
  – ***::192.31.20.46***

IPv6 Addresses

- Entire address space is divided into several categories
  - **Few leftmost bits, called prefix, identifies categories**
- Unicast addresses:
  – ***Geographic based and provider based***



IPv6 Addresses

- Multicast addresses

8 bits	4 bits	4 bits	112 bits
11111111	Flag	Scope	Group ID

- Reserved addresses

8 bits	120 bits	
00000000	All 0s	a. unspecified

8 bits	120 bits	
00000000	000000000000000…………00000000001	b. Loopback

8 bits	88 bits	32 bits	
00000000	All 0s	IPv4 address	c. Compatible

8 bits	72 bits	16 bits	32 bits	
00000000	All 0s	All 1s	IPv4 address	d. Mapped

Bhaskar Sardar, Information Technology Department, Jadavpur University, India

IPv6 Addresses

- Local Addresses

10 bits	70 bits	48 bits	
1111111010	All 0s	Node address	a. Link Local

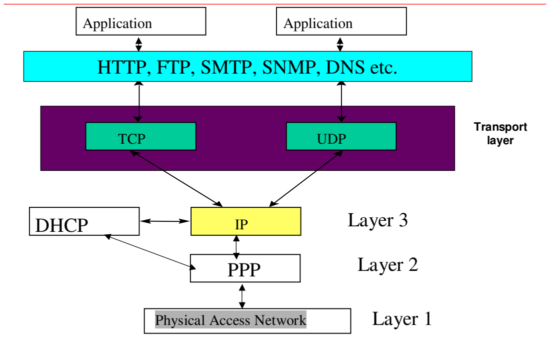
10 bits	38 bits	32 bits	48 bits	
1111111011	All 0s	Subnet address	Node address	b. Site Local

Bhaskar Sardar, Information Technology Department, Jadavpur University, India

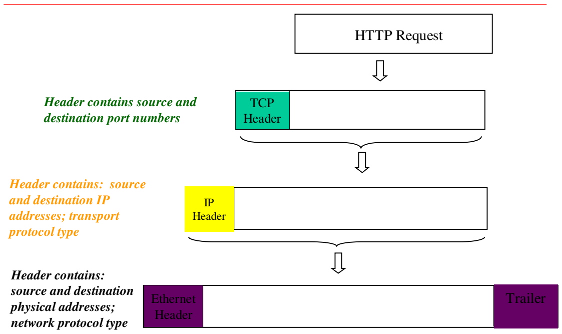
Transmission Control Protocol (TCP)

Bhaskar Sardar, Information Technology Department, Jadavpur University, India

Architectural Relationship



PDU Flow



Prime Design Goals of TCP

- Absorb the shortcomings of IP to make them transparent to the upper layer(s)
  - ***Interface between IP and Application***
- Provide end-to-end (user-user) significance
  - ***Lowermost user-oriented layer***

Limitations of Internet Protocol (IP)

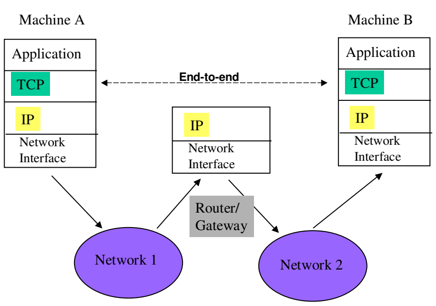
- Best-effort network protocol
  - *Packets may be lost/dropped*
  - *Packets may be delivered out-of-order*
  - *Packets may be duplicated*
  - *Limits messages to some finite size*
  - *Messages may be delayed arbitrarily long*

TCP has to take care of these limitations so that applications do not see them

+

TCP is to provide end-to-end support

TCP is End-to-End



TCP Overview

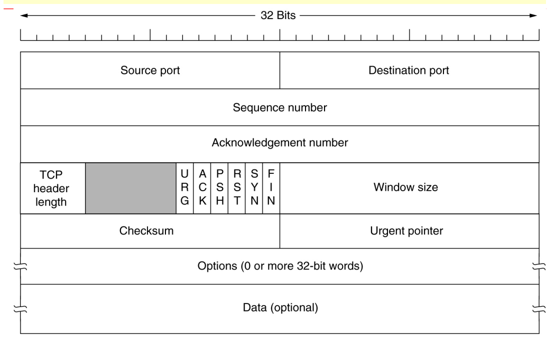
- Used for most Internet Applications
  - *FTP, TELNET etc.*
- Connection Oriented
  - *Full duplex*
- Byte Stream, not a message stream
- Provides Reliable Service

TCP Overview

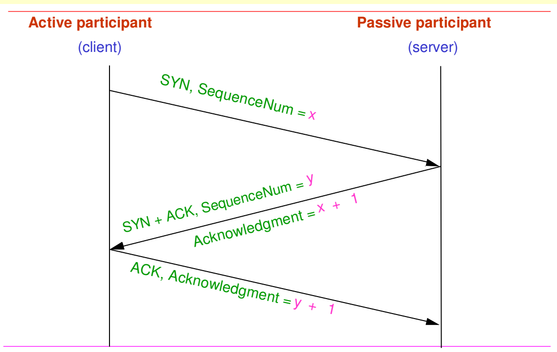
- Flow control
  – ***prevent sender from overrunning receiver***
- Congestion control
  – ***prevent sender from overrunning network***

TCP PDU is conventionally known as segment

TCP Header

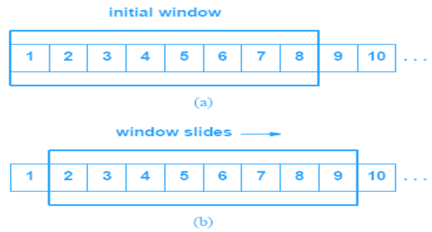


Connection Establishment and Termination

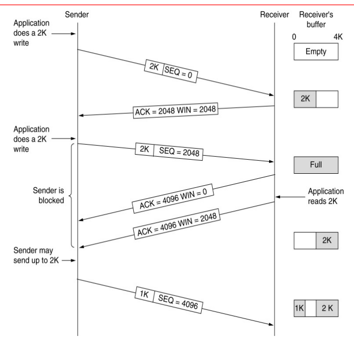


Flow Control

- Uses Sliding Window
- Receiver advertises window size
- As ACK's arrive window move forward



Flow Control

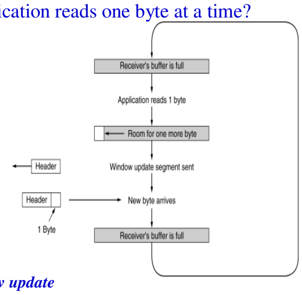


Nagle's Algorithm

- What happens when the application does a 1 byte write?
  - *Interactive editor reacts on every key stroke*
  - *For each character a TCP segment is sent*
- Solution: Nagle's algorithm
  - *Send the first byte*
  - *Buffer rest of the bytes until acknowledgement comes*
  - *Then send buffered bytes in one TCP segment*
  - *Start buffering again*

Silly Window Syndrom

- What happens when application reads one byte at a time?



- Clark's solution:
  - ***Do not send 1 byte window update***
  - ***Wait until decent amount of space is available***

Acknowledgement (ACK) in TCP

- Cumulative acknowledgements
- An acknowledgement ack's all contiguously received previous data
- TCP assigns byte sequence numbers

See the following examples

Example of Flow

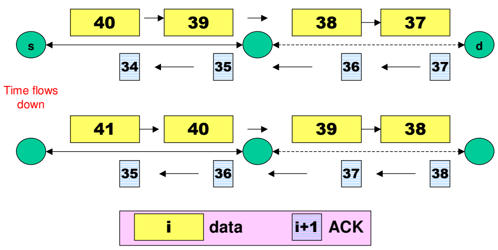
- *ACK[i+1]* acknowledges receipt of packets through packet *i*

For simplicity, we assume following deviations from the normal TCP syntax:-

- We will assign packet sequence numbers
  - ***Not byte sequence numbers***

Example of Cumulative ACK's

- A new cumulative acknowledgement is generated only on receipt of a **new in-sequence** packet

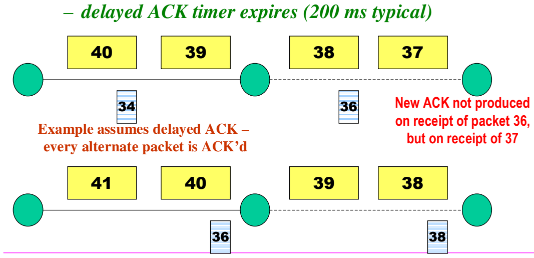


Advantage Of Cumulative ACK

- If some of the ACK's are lost, there is no harm.
- If ACK 49, 51 and 56 are lost, but ACK 60 reaches safely
  - *What will happen?*

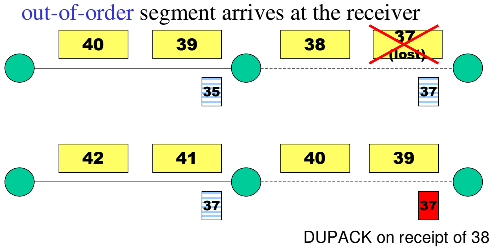
Delayed ACK's

- To reduce ACK traffic, it is delayed
- An ACK is delayed until
  - *another packet is received, or*
  - *delayed ACK timer expires (200 ms typical)*



Duplicate ACK's

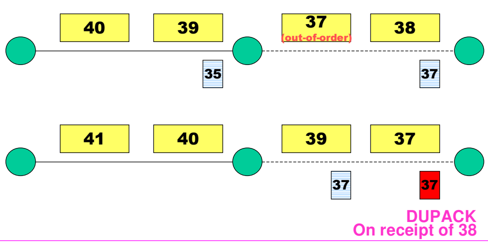
- A DUPACK is generated whenever an out-of-order segment arrives at the receiver



(Above example assumes *delayed acks*)

Duplicate ACK

- Duplicate ACKs are **not delayed**
- Duplicate ACKs may be generated when
  - *a packet is lost, or*
  - *a packet is delivered out-of-order (OOO)*



Congestion Avoidance and Control

- Each sender maintains two windows:
  - **The window the receiver has granted, *rwnd***
  - **The congestion window, *cwnd***
  - ***Effective window= min (rwnd, cwnd)***

Congestion Avoidance and Control

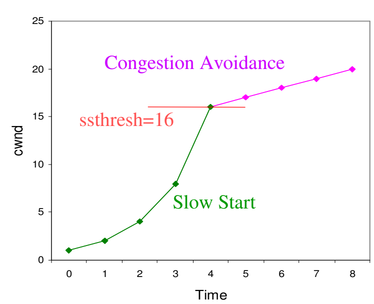
- Slow Start
  - *Initial cwnd is set to 1*
  - *Each new ACK doubles cwnd by 1 Segment*
    - **Exponential Growth**
- How long exponential increase takes place?
  - *Until Time out occurs (Slow Start takes place)*

    or
  - *Until window reaches threshold value, ssthresh, initially 64 KB (Congestion Avoidance takes place)*

Congestion Avoidance and Control

- During Congestion Avoidance
  - ***On each new ACK, increase cwnd by 1/cwnd packets***
  - ***cwnd increases** linearly **with time during congestion avoidance***

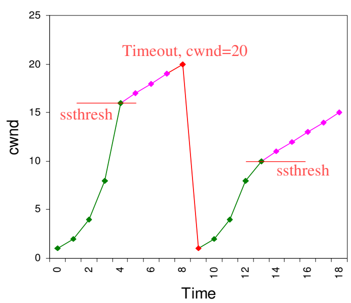
Congestion Avoidance and Control



Congestion Avoidance and Control

- On a timeout
  - *cwnd* **is reduced to the initial value of** *1 MSS*
  - *ssthresh* **is set to half the window size before packet loss**
  - *Slow start* **is initiated**

Congestion Avoidance and Control



TCP Tahoe

- Timeouts can take too long
  - ***how to initiate retransmission sooner?***
- Implement Fast Retransmit
  - ***Packet losses are detected via three DUPACKs***
  - ***Retransmits the lost segment immediately***
  - ***Window reduction is same, applies slow start***

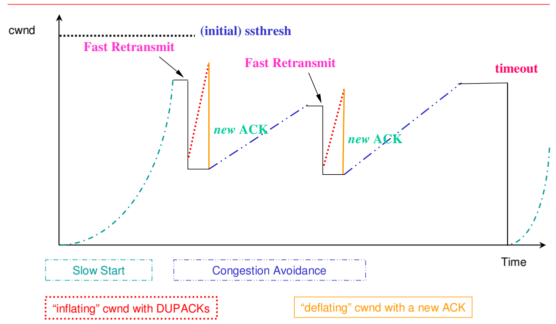
TCP Reno

- Implements Fast Retransmit followed by Fast Recovery
- Different from timeout : slow start follows timeout
  - *timeout occurs when no more packets are getting across*
  - *fast retransmit occurs when a packet is lost, but latter packets get through*

- Observations:
  - *Receiver is still getting TCP segments. There can't be overwhelming congestion.*

- Concept:
  - *After fast retransmit, reduce cwnd by half, and continue sending segments at this reduced level.*

TCP Reno



TCP New-Reno

- Fast recovery can result in a timeout with multiple losses per RTT

- New-Reno implements Fast Retransmit and Modified Fast Recovery

  - *stay in fast recovery until all packet losses in window are recovered*

  - *can recover 1 packet loss per RTT without causing a timeout*

TCP Selective ACK (TCP SACK)

- TCP acknowledgements are cumulative
  - *go-back-n ARQ, thus wasting bandwidth*
- Selective retransmission as one solution
  - *Receiver informs the lost packets in blocks*
  - *sender can now retransmit only the missing packets*
- Advantage:
  - *much higher efficiency*
- Disadvantage:
  - *more complex software in a receiver, more buffer needed at the receiver*

User Datagram Protocol (UDP)

Identifying The Ultimate Destination

- IP address only specifies a computer
- Need a way to specify an application program (process) on a computer
- Unfortunately
  - *Application programs can be created and destroyed rapidly*
  - *Each operating system uses its own identification*
- TCP/IP introduces its own specification
- Destination point known as *port number*
- Each OS binds port number to specific application program

UDP

- Transport-layer protocol (Layer 4)
- Connectionless service: provides application programs with ability to send and receive messages
- Allows multiple, application programs on a single machine to communicate concurrently
- Best-effort semantics as IP
  - *Message can be delayed, lost, or duplicated*
  - *Messages can arrive out of order*
- Does not provide-
  - *Error Control*
  - *Flow Control*

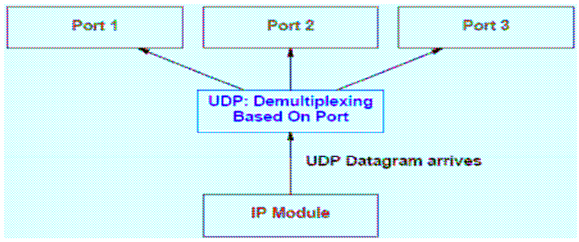
UDP Message Format

0	16	31
UDP SOURCE PORT	UDP DESTINATION PORT	
UDP MESSAGE LENGTH	UDP CHECKSUM	
DATA		
...		

UDP Pseudo Header

0	8	16	31
SOURCE IP ADDRESS			
DESTINATION IP ADDRESS			
ZERO	PROTO	UDP LENGTH	

De-Multiplexing

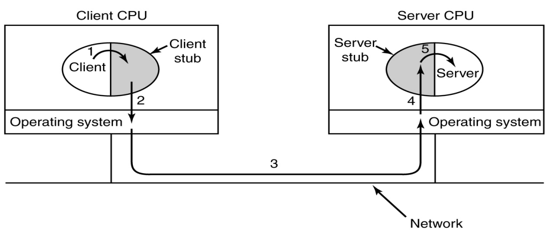


Remote Procedure Call (RPC)

Remote Procedure Call (RPC)

- A technology in which a program on one machine invokes procedure residing in another machine.
- The client procedure is bound with a small library procedure, called client stub.
- The server procedure is bound with a small library procedure, called server stub.
- Together client and server stub are responsible for hiding the fact that procedure call is not local.

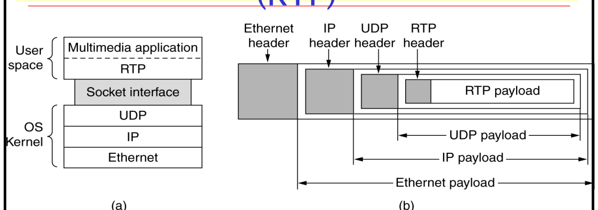
RPC steps



Problems with RPC

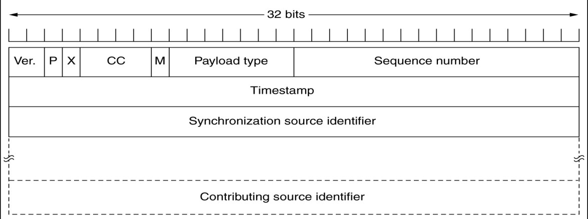
- Passing a pointer
  - *Call-by-reference is replaced by copy-restore*
- It is not always possible to deduce the types of the parameters
- What about global variables??

The Real-Time Transport Protocol (RTP)



(a) The position of RTP in the protocol stack.
(b) Packet nesting.

The RTP Header



Any Doubt ?

- Please feel free to write to me:

bhaskargit@yahoo.co.in