CBCS SCHEME

USN | 1 | C | R | 1 | 8 | 1 | S | 0 | 6 | 4 | 18CS52

Fifth Semester B.E. Degree Examination, Jan./Feb. 2021
Computer Networks and Security

Time: 3 hrs. Max. Marks: 100

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. What are the different transport services available to applications? Explain. **(07 Marks)**

b. Explain HTTP request and response message format. **(08 Marks)**

c. Write a note on FTP and discuss about FTP command and replies. **(05 Marks)**

OR

2 a. What are the steps involved between client and server in order to fetch 10 JPEG images, which are residing in the same server by using non-persistent HTTP connection. The URL for base HTML file is http : //www.xyz.edu/department/base.index. **(07 Marks)**

b. With a neat diagram and explain, explain how DNS server will interact to various DNS server hierarchically. **(05 Marks)**

c. Illustrate how user1 can send mail to user2, and how user2 receives the mail by using SMTP. **(08 Marks)**

Module-2

3 a. How multiplexing and demultiplexing for a connectionless oriented will be performed at transport layer? **(06 Marks)**

b. Describe the various fields of UDP segment and also explain about UDP checksum with an example. **(07 Marks)**

c. Explain how TCP provides a flow control service by using different variables. **(07 Marks)**

OR

4 a. Explain the operation of selective repeat protocol. **(06 Marks)**

b. Explain all the fields in a TCP segment. **(07 Marks)**

c. How TCP connection management is done for three way handshake by the client and server for establishing and closing a connection. Explain. **(07 Marks)**

Module-3

5 a. Explain distance vector algorithm with an example. **(08 Marks)**

b. Explain the three switching techniques in a router. **(06 Marks)**

c. Draw IPV_6 datagram format, mention the significance of each fields. **(06 Marks)**

OR

6 a. Explain link state algorithm with an example. **(08 Marks)**

b. Describe the intra-AS routing protocol : RIP in detail. **(06 Marks)**

c. Discuss about uncontrolled flooding and controlled flooding in broadcast routing algorithm. **(06 Marks)**

Module-4

7 a. Classify the different network attacks and explain denial of service attack. (07 Marks)

b. What are the two different techniques used to protect network from attacks? Explain. (07 Marks)

c. Write the steps involved in Data Encryption Standard (DES) along with a diagram. (06 Marks)

OR

8 a. Explain key generation, encryption and decryption phases in RSA algorithm. Illustrate with an example. (07 Marks)

b. Explain the technique involved in Hash function for authentication along with a diagram. (07 Marks)

c. Discuss about packet filtering and proxy server with respect to firewalls. (06 Marks)

Module-5

9 a. What are the classification in multimedia network applications? Explain. (08 Marks)

b. What are the two types of loss anticipation schemes? Explain. (07 Marks)

c. What do you mean by a Jitter and how to remove the Jitter at the receiver for audio by fixed and adaptive play out delay? (05 Marks)

OR

10 a. Explain the working of CDN. (08 Marks)

b. Explain about HTTP streaming in case of streaming stored video. (07 Marks)

c. Discuss about the properties of audio and video in multimedia networking. (05 Marks)

* * * * *

VTU Answer Key-Feb/2021 Date: 22.02.2021

18CS52-Computer Network and Security

Total marks: 100

Module 1

1. a. What are the different transport services available to applications? Explain? (7 marks)

What transport service does an app need?

data integrity
- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer
- other apps (e.g., audio) can tolerate some loss

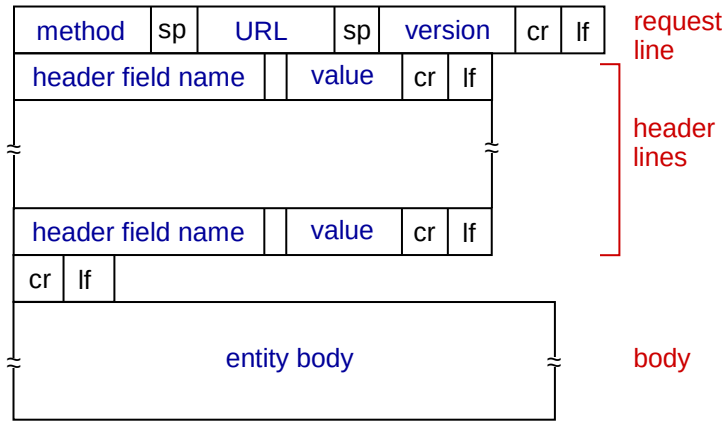
timing
- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

throughput
- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

security
- encryption, data integrity, …

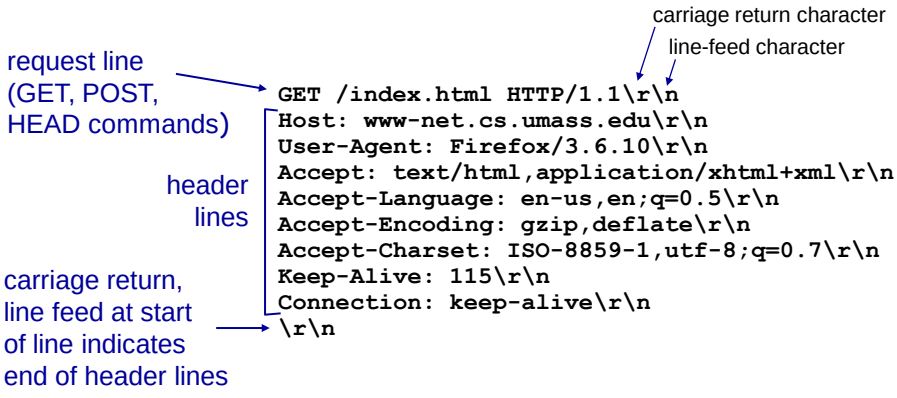
b. Explain HTTP request and response message format. (8 marks)

HTTP request message: general format

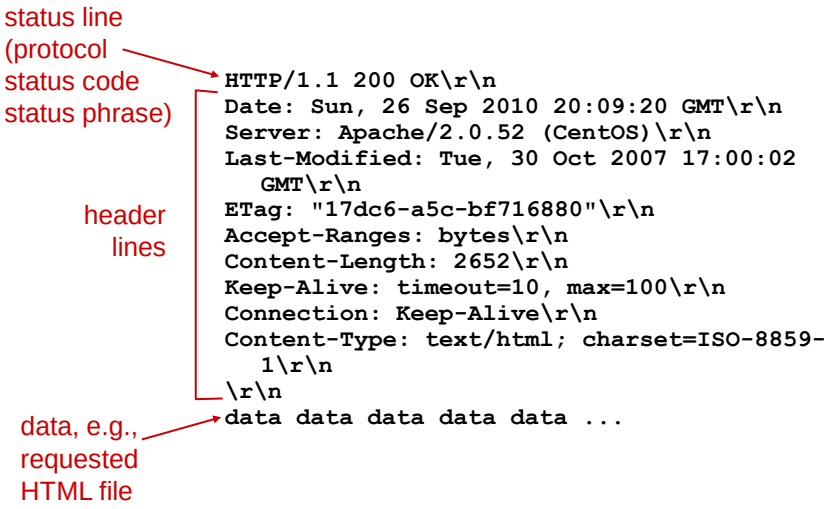


HTTP request message

- two types of HTTP messages: *request, response*
- HTTP request message:
  - ASCII (human-readable format)



HTTP response message



HTTP response status codes

- status code appears in 1st line in server-to-client response message.
- some sample codes:

200 OK
- request succeeded, requested object later in this msg

301 Moved Permanently
- requested object moved, new location specified later in this msg (Location:)

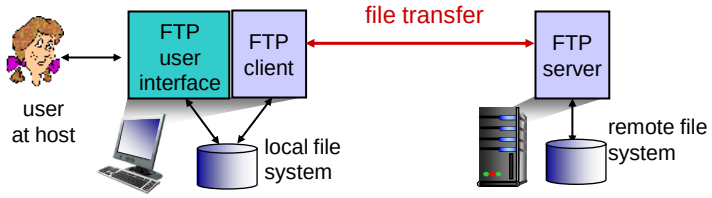
400 Bad Request
- request msg not understood by server

404 Not Found
- requested document not found on this server

505 HTTP Version Not Supported

c. Write a note on FTP and discuss about FTP command and replies. (5 marks)

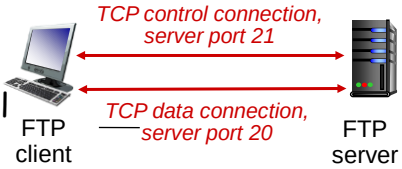
FTP: the file transfer protocol



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, using TCP
- client authorized over control connection
- client browses remote directory, sends commands over control connection
- when server receives file transfer command, *server* opens 2nd TCP data connection (for file) to client
- after transferring one file, server closes data connection



- server opens another TCP data connection to transfer another file
- control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

FTP commands, responses

sample commands:

- sent as ASCII text over control channel
- **USER** ***username***
- **PASS** ***password***
- **LIST** return list of file in current directory
- **RETR** **filename** retrieves (gets) file
- **STOR** **filename** stores (puts) file onto remote host

sample return codes

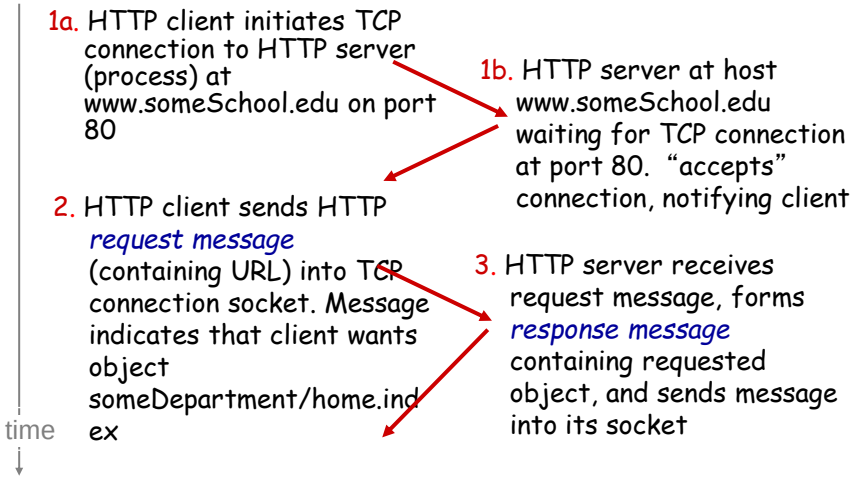
- status code and phrase (as in HTTP)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

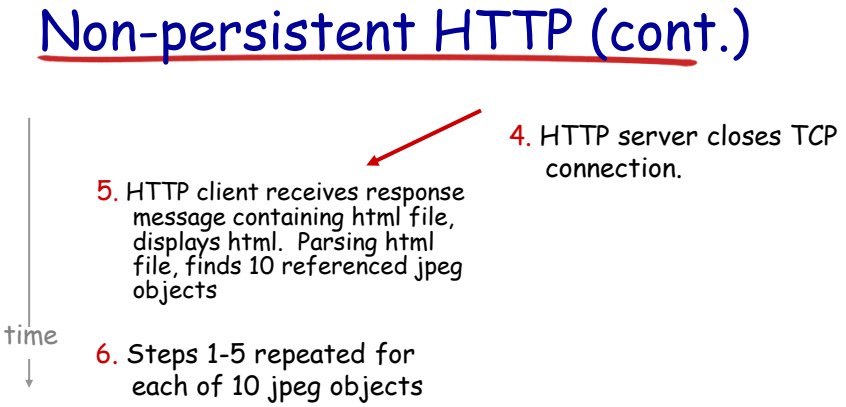
2. a. What are the steps involved between client and server in order to fetch 10 JPEG images, which are residing in the same server by using non-persistent HTTP connection. The URL for base HTML file is http:// www.xyz.edu/department/base.index (7 marks)

Non-persistent HTTP

suppose user enters URL:
www.xyz.edu/department/base.index

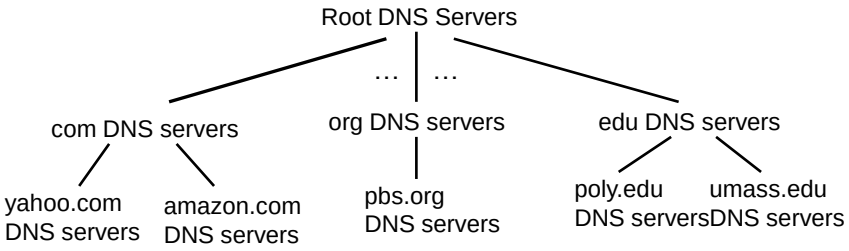
(contains text, references to 10 jpeg images)





b. With a neat diagram and explain, how DNS server will interact with various DNS servers hierarchically. (5 marks)

DNS: a distributed, hierarchical database

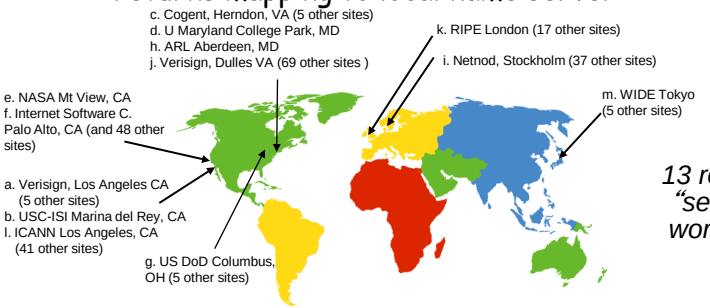


client wants IP for www.amazon.com; 1st approx:

- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



13 root name "servers" worldwide

TLD, authoritative servers

top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD

authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name server

- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

c. Illustrate how user1 can send mail to user2, and how user2 receives the mail using SMTP. (8 marks)

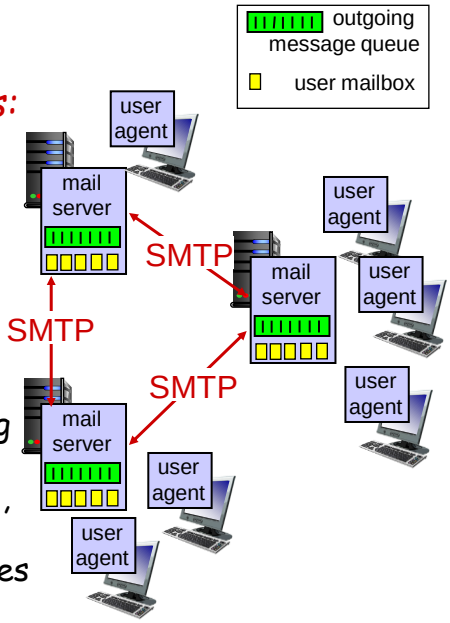
Electronic mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

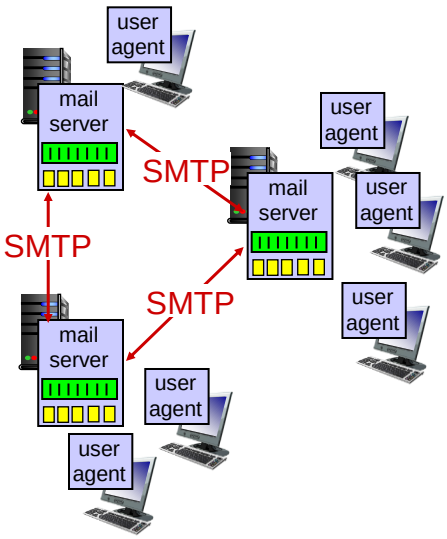
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird, iPhone mail client
- outgoing, incoming messages stored on server



Electronic mail: mail servers

mail servers:

- *mailbox* contains incoming messages for user
- *message queue* of outgoing (to be sent) mail messages
- *SMTP protocol* between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server

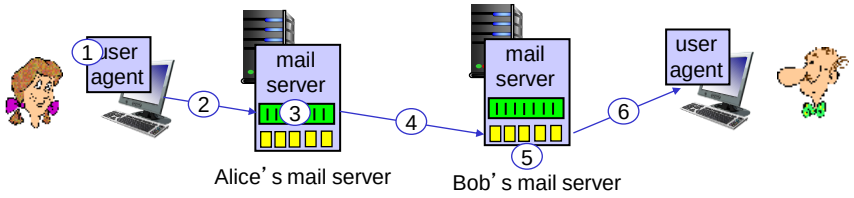


Electronic Mail: SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction (like HTTP, FTP)
  - commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCI

Scenario: Alice sends message to Bob

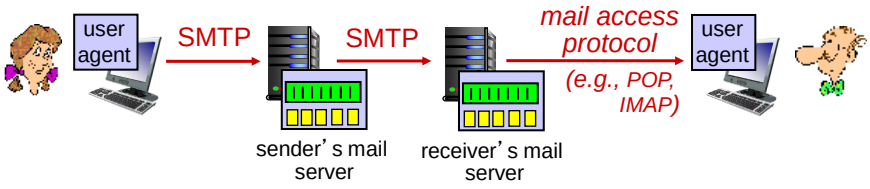
1) Alice uses UA to compose message "to" `bob@someschool.edu`
2) Alice's UA sends message to her mail server; message placed in message queue
3) client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message



Sample SMTP interaction

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Mail access protocols

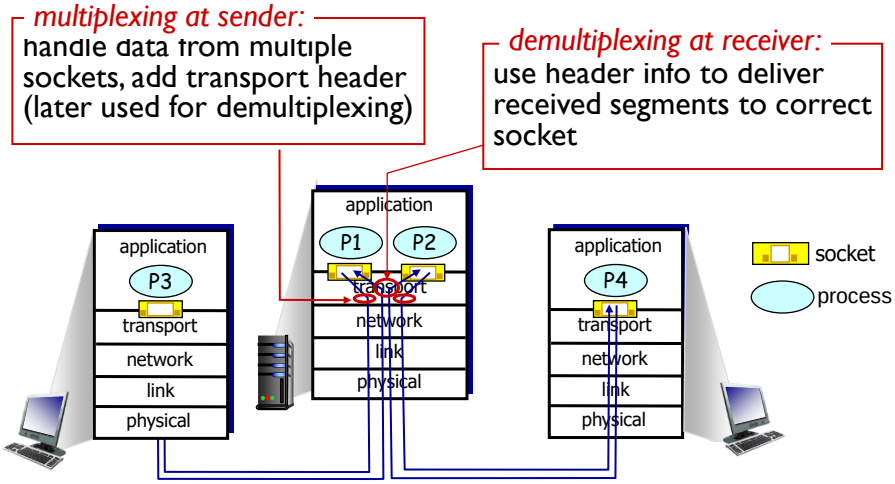


- SMTP: delivery/storage to receiver's server
- mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]: authorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

Module 2

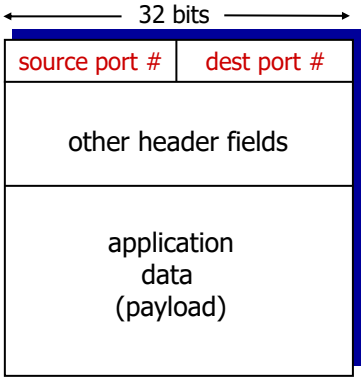
3. a. How multiplexing and de-multiplexing for a connection-less oriented will be performed at transport layer? (6 marks)

Multiplexing/demultiplexing



How demultiplexing works

- host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries one transport-layer segment
  - each segment has source, destination port number
- host uses *IP addresses & port numbers* to direct segment to appropriate socket



TCP/UDP segment format

Connectionless demultiplexing

- *recall:* created socket has host-local port #:

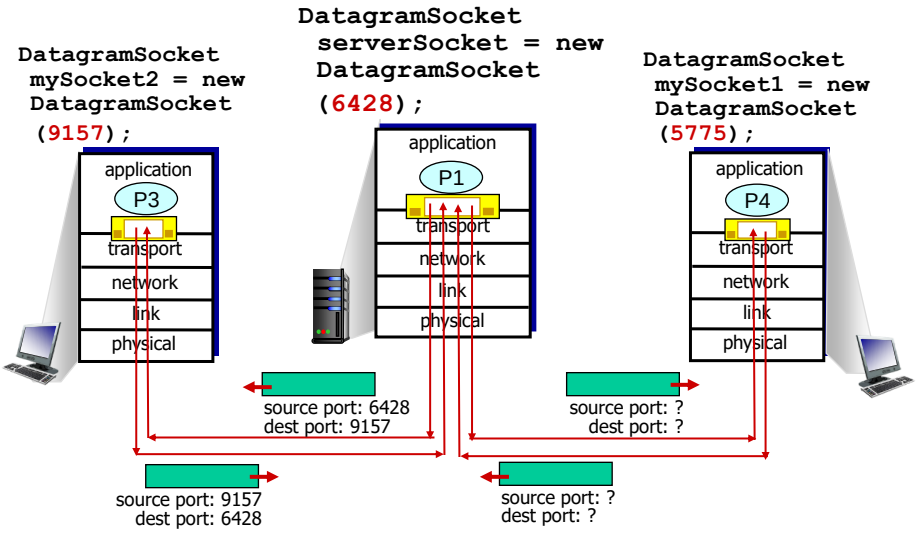
```
DatagramSocket mySocket1
= new DatagramSocket(12534);
```

- *recall:* when creating datagram to send into UDP socket, must specify
  - destination IP address
  - destination port #

- when host receives UDP segment:
  - checks destination port # in segment
  - directs UDP segment to socket with that port #

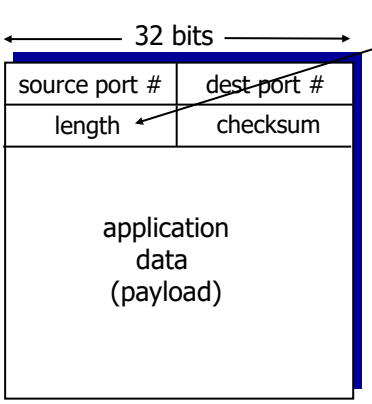
IP datagrams with *same dest. port #,* but different source IP addresses and/or source port numbers will be directed to *same socket* at dest

Connectionless demux: example



b. Describe the various fields of UDP segment and also explain about UDP checksum with an example. (7 marks)

UDP: segment header



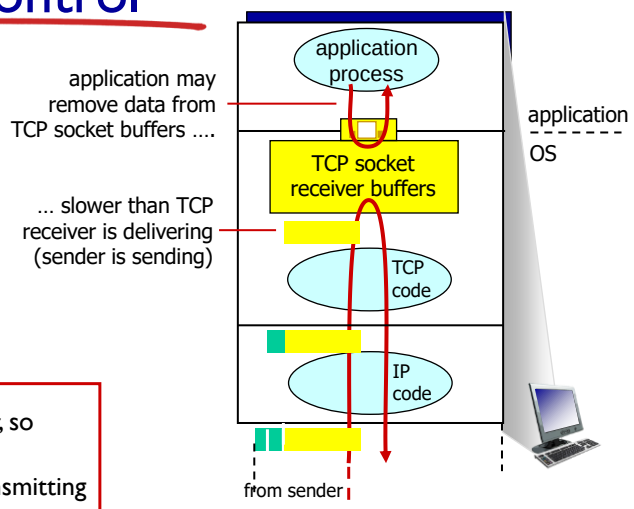
UDP segment format

why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired

c. Explain how TCP provides a flow control service by using different variables. (7 marks)

TCP flow control



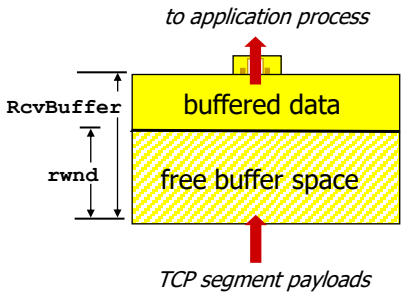
receiver protocol stack

flow control

receiver controls sender, so sender won't overflow receiver's buffer by transmitting too much, too fast

TCP flow control

- receiver "advertises" free buffer space by including **`rwnd`** value in TCP header of receiver-to-sender segments
  - **`RcvBuffer`** size set via socket options (typical default is 4096 bytes)
  - many operating systems autoadjust **`RcvBuffer`**
- sender limits amount of unacked ("in-flight") data to receiver's **`rwnd`** value
- guarantees receive buffer will not overflow



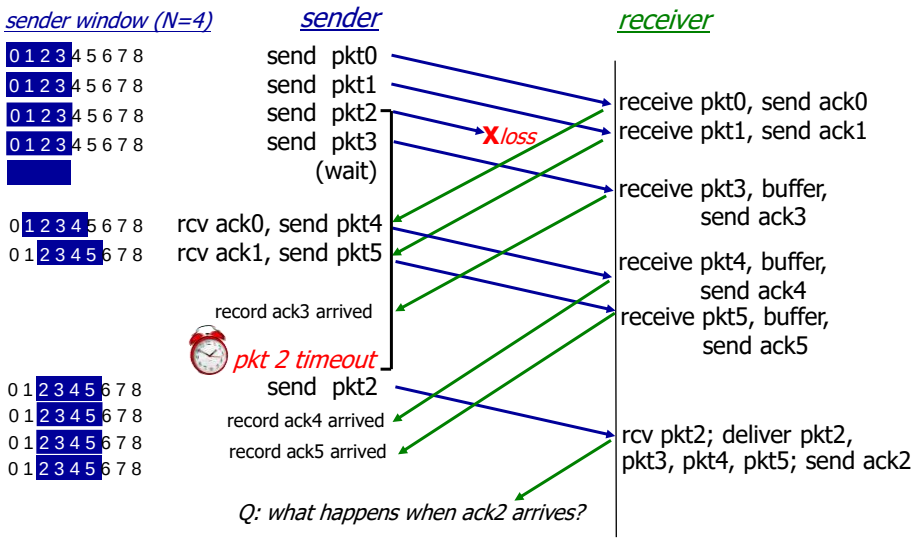
receiver-side buffering

4. a. Explain the operation of selective repeat protocol. (6 marks)

Selective repeat

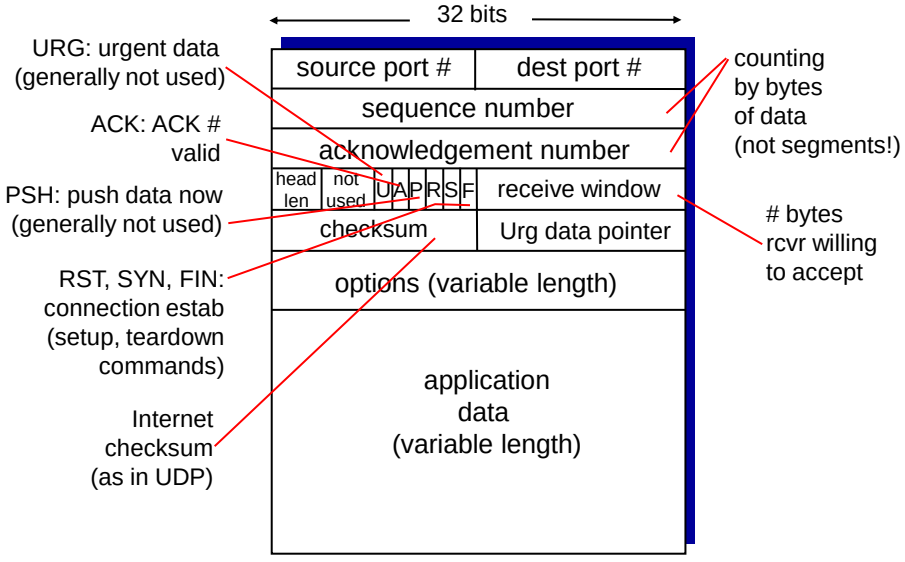
- receiver *individually* acknowledges all correctly received pkts
  - buffers pkts, as needed, for eventual in-order delivery to upper layer
- sender only resends pkts for which ACK not received
  - sender timer for each unACKed pkt
- sender window
  - *N* consecutive seq #'s
  - limits seq #s of sent, unACKed pkts

Selective repeat in action



b. Explain all the fields in a TCP segment. (7 marks)

TCP segment structure

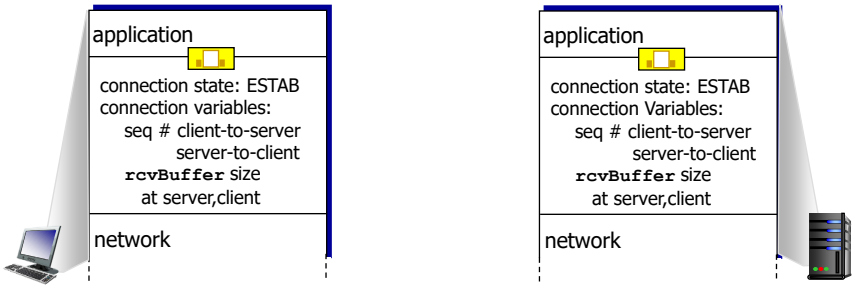


c. How TCP connection establishment is done for 3-way handshake for establishing and closing a connection. Explain. (7 marks)

Connection Management

before exchanging data, sender/receiver "handshake":

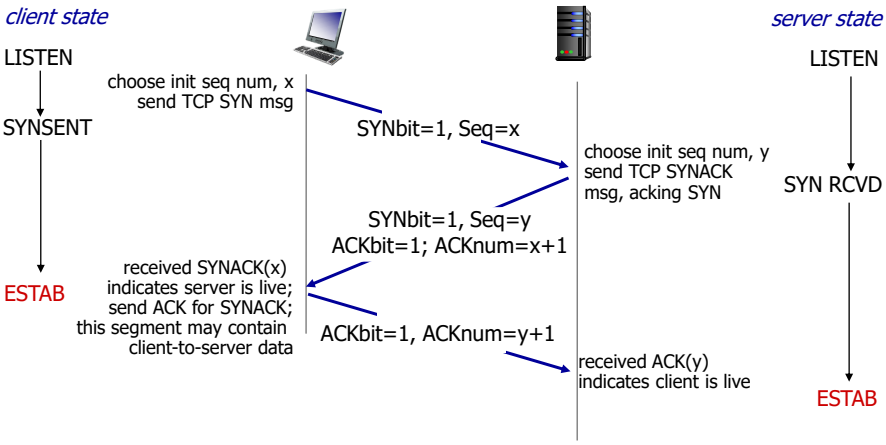
- agree to establish connection (each knowing the other willing to establish connection)
- agree on connection parameters




```
Socket clientSocket =
  newSocket("hostname","port
   number");
```

```
Socket connectionSocket =
   welcomeSocket.accept();
```

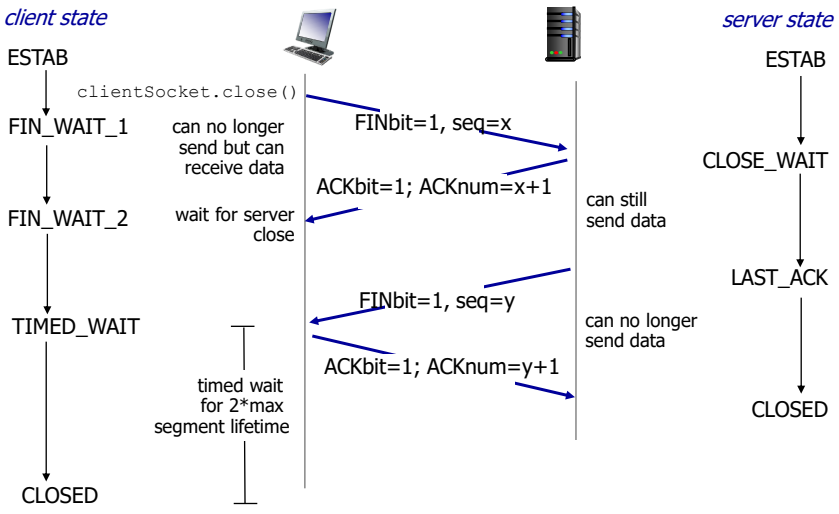
TCP 3-way handshake



TCP: closing a connection

- client, server each close their side of connection
  - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
  - on receiving FIN, ACK can be combined with own FIN
- simultaneous FIN exchanges can be handled

TCP: closing a connection



Module 3

5. a. Explain distance vector algorithm with an example. (8 marks)

Distance vector algorithm

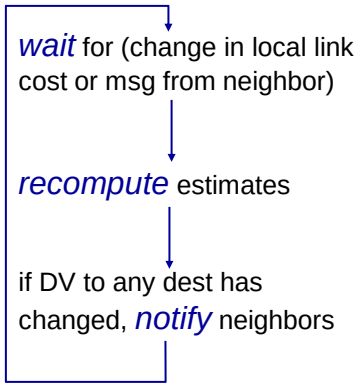
iterative, asynchronous:
each local iteration caused by:

- local link cost change
- DV update message from neighbor

distributed:

- each node notifies neighbors *only* when its DV changes
  - neighbors then notify their neighbors if necessary

each node:



Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let
$d_x(y)$:= cost of least-cost path from x to y
then

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

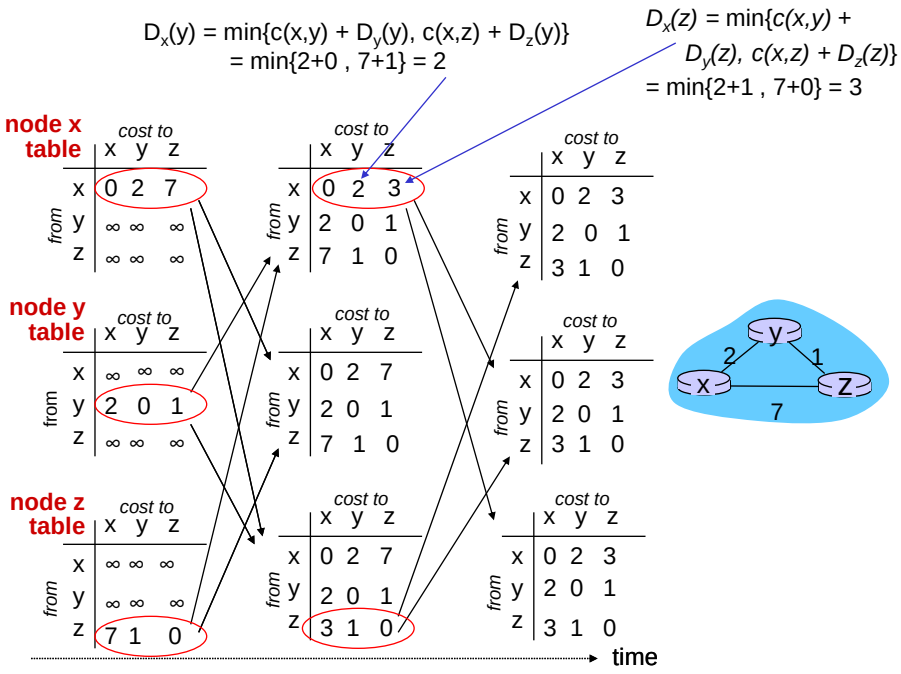
$d_v(y)$: cost from neighbor v to destination y

$c(x,v)$: cost to neighbor v

$\min_v$: *min* taken over all neighbors v of x

Algorithm

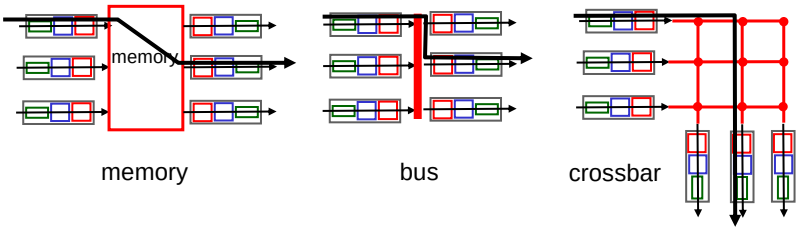
1 ***Initialization:***
2 for all destinations y in N:
3 $D_x(y)$: c(x,y)
4 for each neighbor w
5 $D_w(y)$:? for all destinations y in N
6 for each neighbor w
7 **send distance vector $\mathbf{D}_x$ = $[D_x(y)$: y ϵ N] to w**
8 loop
9 wait until I see a link cost change to some neighbor w or
10 until I receive a distance vector from some neighbor w
11 for each y in N:
12 $D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\}$
13 *if* $D_x(y)$ changed for any destination y
14 send distance vector $\mathbf{D}_x$ = $[D_x(y)$: y ϵ N] to all neighbors
15 forever



b. Explain three switching techniques in a router. (6 marks)

Switching fabrics

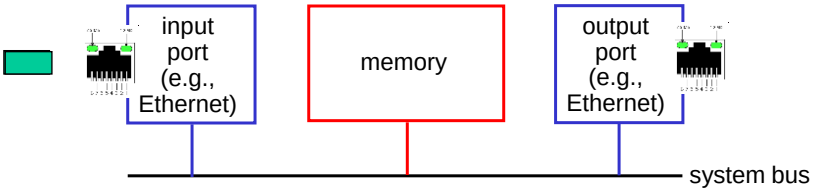
- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
- three types of switching fabrics



Switching via memory

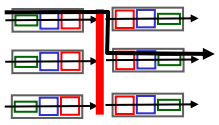
first generation routers:

- traditional computers with switching under direct control of CPU(routing processor)
- packet copied to system's memory
- only one packet can be send at a time



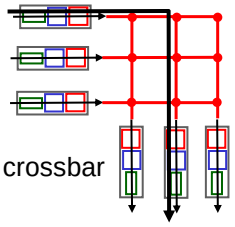
Switching via a bus

- datagram from input port memory to output port memory via a shared bus
- *if multiple packets arrives they have to wait as one packet can cross the bus*



Switching via interconnection network

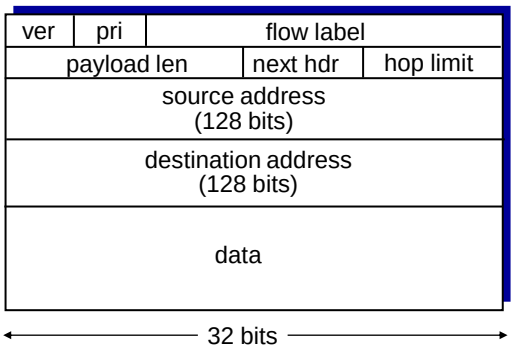
- overcome bus bandwidth limitations
- a cross bar switch is an interconnection network consisting of 2N buses that connect N input ports to N output ports.
- capable of forwarding multiple packets in parallel.



c. Draw IPV6 packet format, mention the significance of each field. (6 marks)

IPv6 datagram format

priority: identify priority among datagrams in flow
flow Label: identify datagrams in same "flow."
(concept of"flow" not well defined).
next header: identify upper layer protocol for data



6. a. Explain link state algorithm with an example. (8 marks)

A Link-State Routing Algorithm

Dijkstra's algorithm

- computes least cost paths from one node ('source") to all other nodes
- iterative: after k iterations, know least cost path to k dest.' s

notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

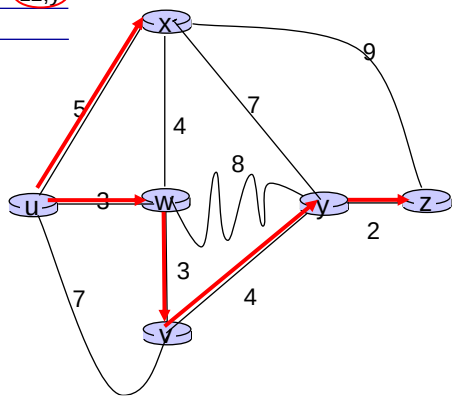
Dijsktra's Algorithm

```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v adjacent to u
5         then D(v) = c(u,v)
6     else D(v) = ∞
7
8   Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N' :
12      D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    shortest path cost to w plus cost from w to v */
15  until all nodes in N'
```

Dijkstra's algorithm: example

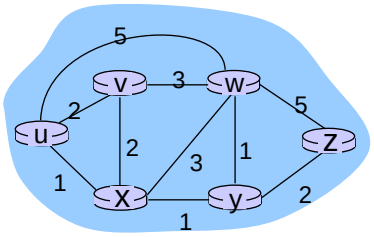
Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

Complexity: $O(n^2)$



Dijkstra's algorithm

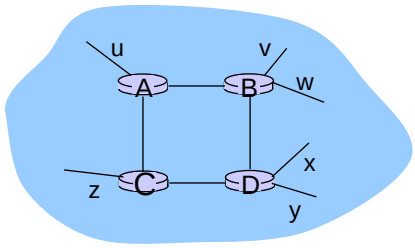
Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



b. Describe intra AS routing protocol: RIP in detail. (6 marks)

RIP (Routing Information Protocol)

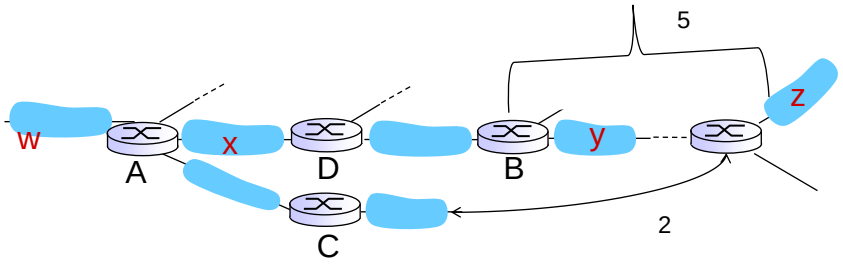
- distance vector algorithm
  - distance metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing sense)*



from router A to destination *subnets:*

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP: example



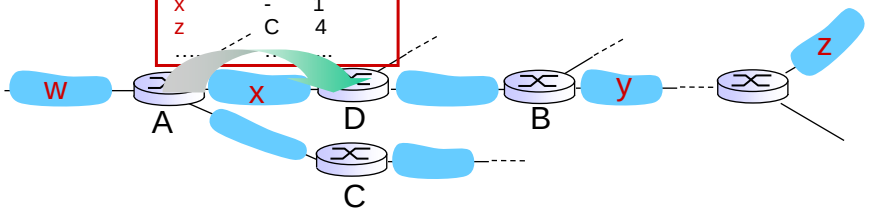
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

RIP: example

A-to-D advertisement

dest	next	hops
w	-	1
x	-	1
z	C	4
….	….	….



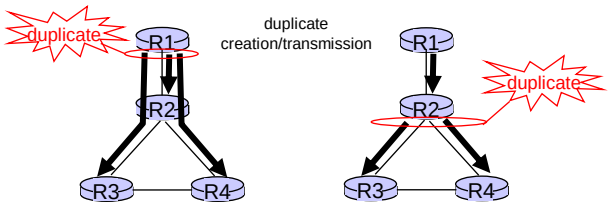
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	….

c. Describe about uncontrolled flooding and controlled flooding in broadcast routing algorithm. (6 marks)

Broadcast routing

- deliver packets from source to all other nodes
- source duplication is inefficient:



source duplication

in-network duplication

- source duplication: how does source determine recipient addresses?

In-network duplication

- *flooding:* when node receives broadcast packet, sends copy to all neighbors
  - problems: cycles & broadcast storm
- *controlled flooding:* node only broadcasts pkt if it hasn't broadcast same packet before
  - Sequence number controlled flooding: node keeps track of packet ids already broadcasted
  - or reverse path forwarding (RPF): only forward packet if it arrived on shortest path between node and source
- *spanning tree:*
  - no redundant packets received by any node

Module 4

7. a. Classify the different network attacks and explain denial of service attack. (7 marks)

- **Denial-of-Service Attacks**
  - ❖A denial-of-service attack is a type of security breach that prohibits a user from accessing normally provided services.
  - ❖The denial of service does not result in information theft or any kind of information loss but can be very dangerous, as it can cost the target person a large amount of time and money.
  - ❖Denial-of-service attacks affect the destination rather than a data packet or router.
  - ❖E.g. Buffer overflow

- The attack take important servers out of action for few hours.
- There are yet a few other situations that can cause this kind of attack, such as UDP flood, a TCP flood and ICMP flood.
- Denial-of-service attacks are two types:
- Single-source. An attacker sends a large number of packets to a target system
- Distributed. In this type of attack, a large number of hosts are used to flood unwanted traffic to a single target. The target cannot then be accessible to other users in the network, as it is processing the flood of traffic.

- The flood may be either a UDP flood or a TCP SYN flood.
- UDP flooding is used against two target systems and can stop the services offered by either system.
- Normally, a SYN packet is sent by a host to a user who intends to establish a connection. The user then sends back an acknowledgment.
- In the TCP SYN flood, a hacker sends a large number of SYN packets to a target user. Since the return addresses are spoofed, the target user queues up a SYN/ACK packet and never processes it.
- Therefore, the target system keeps on waiting. The result may be a hard disk crash or reboot.

b. What are the two different techniques used to protect network from attacks? Explain. (7 marks)

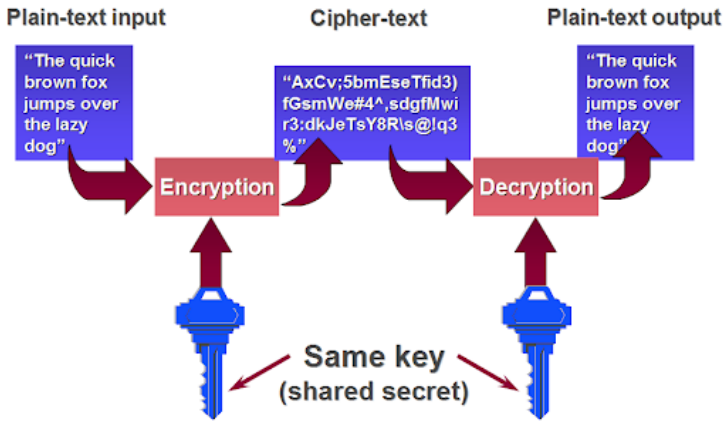
Overview of Security Methods

- Common solutions that can protect computer communication networks from attacks are classified as
  - ❖cryptographic techniques
  - ❖authentication techniques (verification)

Cryptographic Techniques

- Centuries ago, cryptography was used as a tool to protect national secrets and strategies. Today, network engineers focus on cryptography methods for computer communication networks.
- Cryptography is the process of transforming a piece of information or message shared by two parties into some sort of code.
- The message is scrambled before transmission so that it is undetectable by outside watchers.
- This kind of message needs to be decoded at the receiving end before any further processing.

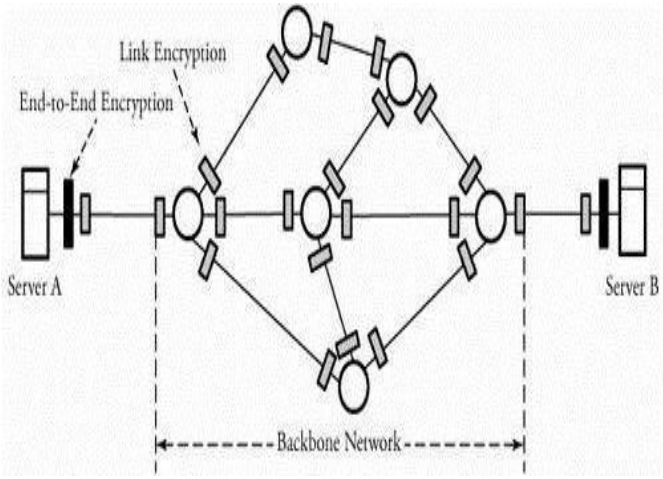
Encryption/Decryption



- The main tool that network security experts are using to encrypt
  - a message M is a secret key K; the fundamental operation often used to encrypt a message is the Exclusive-OR ( ).
  - Suppose that we have one bit, M, and a secret bit,K.
  - A simple encryption is carried out using M K. To decrypt this message, the second party if he/she has the key, K can easily detect M by performing the following:

$$(M \oplus K) \oplus K = M.$$

Figure: Overview of encryption points in a communication network



- In this figure, server A encodes its data, which can be decoded only at the other end server. The second phase of encryption is link encryption, which secures all the traffic passing over that link.
- Types:
  - ❖Secret-key encryption/Symmetric key encryption
  - ❖Public- key encryption/Asymmetric key encryption

Authentication Techniques

- Message Confidentiality
  - ❖ Encryption methods
- Authentication & Integrity
  - ❖ Message digest (hash function)
  - ❖ Digital Signature

  These methods do not necessarily use keys

Secret-Key Encryption Protocols

- Secret-key encryption protocols, sometimes known as symmetric encryption/ single-key encryption protocols/ conventional encryption models.
- They typically consist of an encryption algorithm, a key, and a decryption algorithm.
- At the end point, the encrypted message is called ciphertext.
- The security of conventional encryption depends on the secrecy of the key, not on the secrecy of the encryption algorithm.

- Several standard mechanisms can be used to implement a secret-key encryption algorithm.
- Here, we focus on two protocols:
  - ❖Data Encryption Standard (DES)
  - ❖Advanced Encryption Standard (AES)

c. Write the steps involved in Data Encryption Standard (DES) along with a diagram. (6 marks)

Data Encryption Standard (DES)

- Plain text: 64 bit blocks
- Key: 56 bits (reduced from 64 to 56 bits)
- Number of rounds: 16
- Steps:
- Begin DES Algorithm
- Initialize. Before round 1 begins, all 64 bits of an incoming message and all 56 bits of the secret key are separately permuted (shuffled).
- Each incoming 64-bit message is broken into two 32-bit halves denoted by L_i and R_i, respectively.

- The 56 bits of the key are also broken into two 28-bit halves, and each half is rotated one or two bit positions, depending on the round.
- All 56 bits of the key are permuted, producing version k_i of the key on round i.
- In this step, is a logic Exclusive-OR, and the description of function F() appears next. Then, L_i and R_i are determined by

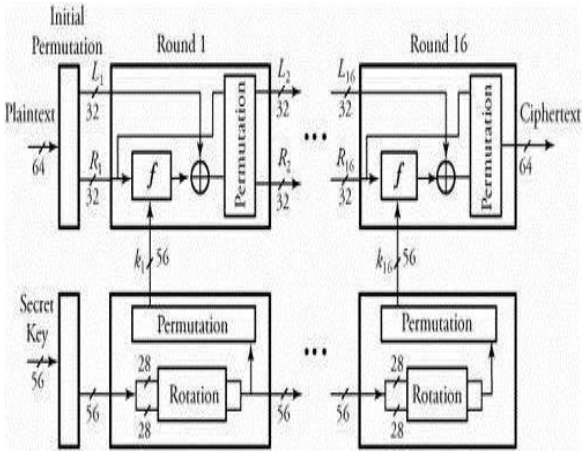
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, k_i).$$

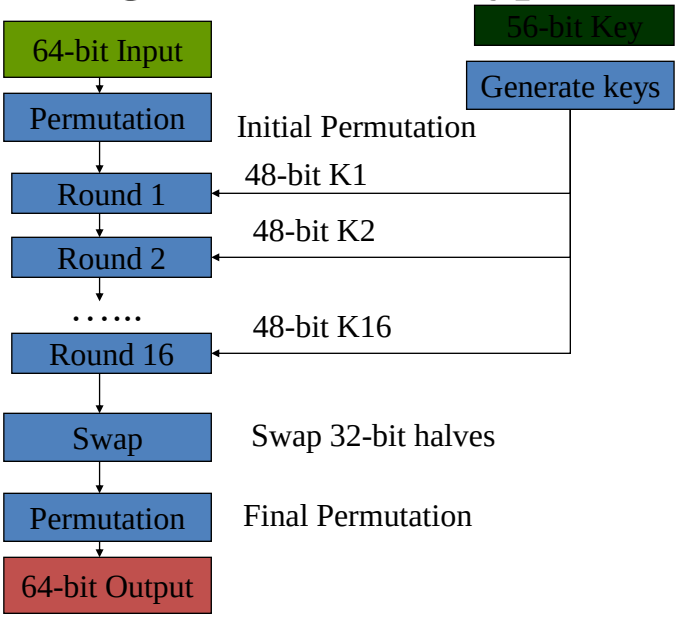
Fiestel Structure

- Block size-64 bits
- Key size-56 bits
- Stages:
- Initial Permutation
- 16 rounds of a given function
- 32 bit left-right swap
- Final permutation

Figure: The Data Encryption Standard (DES)

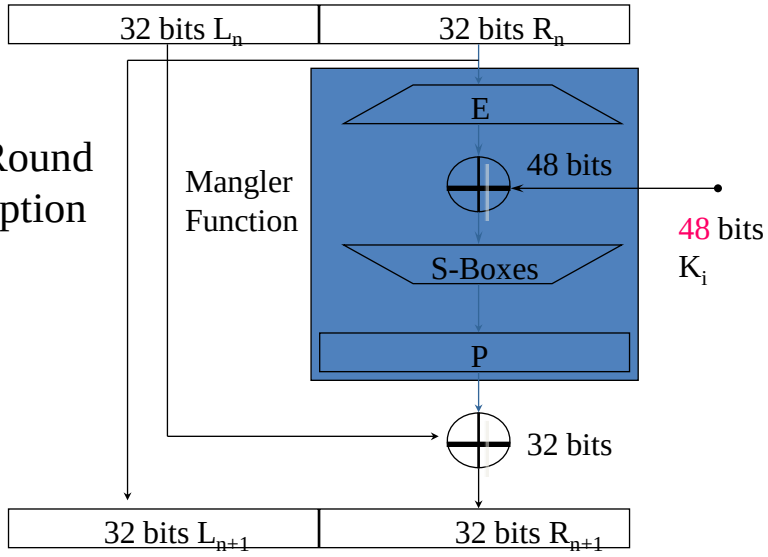


Stages in DES Encryption



Single round of DES

One Round Encryption

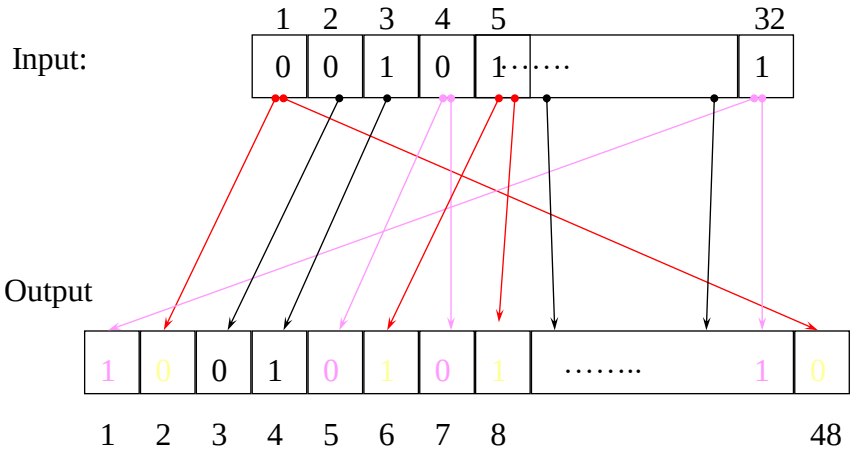


- In encryption
- Li=Ri-1
- Ri=Li-1 xor f(Ri-1,Ki)
- In Decryption
- Ri-1=Li
- Li-1=Ri xor f(Li,Ki)

Round function f(Ri-1,Ki)

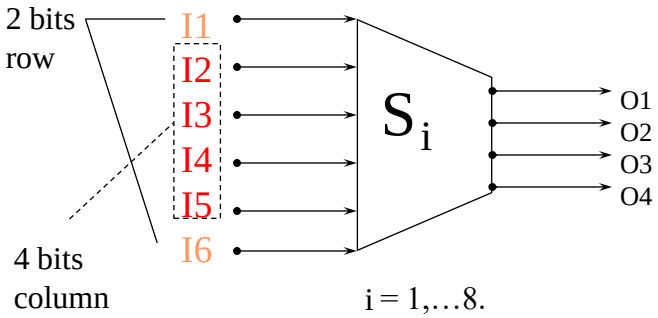
- Expansion-32 bit to 48 bit
- Xor with the round key- 48 bit key
- Substitution
- Permutation

Bits Expansion (1-to-m)

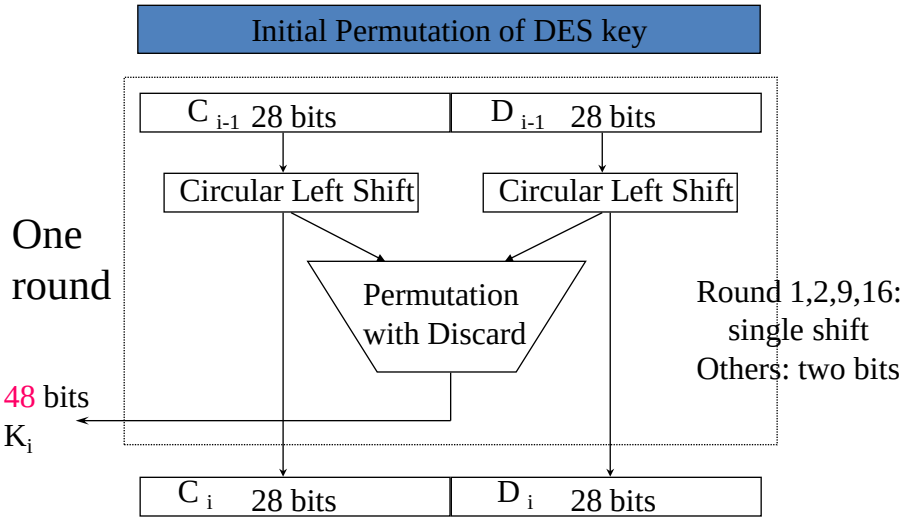


S-Box (Substitute and Shrink)

- 48 bits ==> 32 bits. (8*6 ==> 8*4)
- 2 bits used to select amongst 4 substitutions for the rest of the 4-bit quantity



Per-Round Key Generation



- The operation of function F() at any round i of DES is as follows.
- Out of 52 bits of k_i, function F() chooses 48 bits.
- The 32-bit R_{i-1} is expanded from 32 bits to 48 bits so that it can be combined with 48-bit k_i.
- The expansion of R_{i-1} is carried out by first breaking R_{i-1} into eight 4-bit chunks and then expanding each chunk by copying the leftmost bit and the rightmost bit from left and right adjacent chunks, respectively.
- Function F() also partitions the 48 bits of k_i into eight 6-bit chunks.
- The corresponding eight chunks of R_{i-1} and eight chunks of k_i are combined as follows:

$$R_{i-1} = R_{i-1} \oplus k_i.$$

- At the receiver, the same steps and the same key are used to reverse the encryption.
- It is now apparent that the 56-bit key length may not be sufficient to provide full security. This argument is still controversial.
- Triple DES provides a solution for this controversy: three keys are used, for a total of 168 bits.

8.a. Explain key generation, encryption and decryption phases in RSA algorithm. Illustrate with an example. (7 marks)

RSA Algorithm

- Rivert, Shamir, and Aldeman developed the RSA public-key encryption and signature scheme.
- This was the first practical public-key encryption algorithm.
- The RSA algorithm has three phases for this:
  - ❖key generation
  - ❖encryption
  - ❖decryption

RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random: p, q
- computing their system modulus n=p.q
  - note ø(n)=(p-1)(q-1)
- selecting at random the encryption key e
  - where 1<e<ø(n), gcd(e,ø(n))=1
- solve following equation to find decryption key d
  - e.d=1 mod ø(n) and 0≤d≤n
- publish their public encryption key: PU={e,n}
- keep secret private decryption key: PR={d,n}

RSA En/decryption

- to encrypt a message M the sender:
  - obtains **public key** of recipient PU={e,n}
  - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key PR={d,n}
  - computes: $M = C^d \bmod n$

RSA Example - Key Setup

1. Select primes: p=17 & q=11
2. Calculate $n = pq$ =17 x 11=187
3. Calculate $ø(n)=(p–1)(q-1)$=16x10=160
4. Select e: gcd(e,160)=1; choose e=7
5. Determine d: de=1 mod 160 and d < 160 Value is d=23 since 23x7=161= 10x160+1
6. Publish public key PU={7,187}
7. Keep secret private key PR={23,187}

RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message M = 88 (nb. 88<187)
- encryption:

$C = 88^7 \bmod 187 = 11$

- decryption:

$M = 11^{23} \bmod 187 = 88$

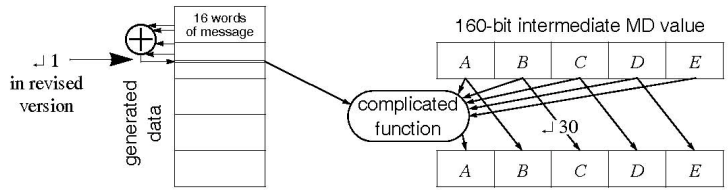
Exponentiation

- can use the Square and Multiply Algorithm
- a fast, efficient algorithm for exponentiation
- concept is based on repeatedly squaring base
- and multiplying in the ones that are needed to compute the result
- look at binary representation of exponent
- only takes $O(\log_2 n)$ multiples for number n
  - eg. $7^5 = 7^4.7^1 = 3.7 = 10 \bmod 11$
  - eg. $3^{129} = 3^{128}.3^1 = 5.3 = 4 \bmod 11$

b. Explain the technique involved in Hash function for authentication along with a diagram. (7 marks)

Secure Hash Algorithm (SHA)

- The Secure Hash Algorithm (SHA) was proposed as part of the digital signature standard. SHA-1, the first version of this standard, takes messages with a maximum length of 2^{24} and produces a 160-bit digest.



SHA-1 Algorithm

- 160 bit hash using 512 bit blocks and 32 bit operations
- Five passes (4 in MD5 and 3 in MD4) of 16 operations each
- Maximum message size is 2^{64} bit
- 512 bits are expanded to 5x512 bits:
  - n^{th} word = xor of n-3, n-8, n-14, and n-16
- In SHA-1 these words are rotated left by one bit before xor
- Total 80 words: W_0, ..., W_{79}

Secure Hash Algorithm

- SHA originally designed by NIST & NSA in 1993
- was revised in 1995 as SHA-1
- US standard for use with DSA signature scheme
  - standard is FIPS 180-1 1995, also Internet RFC3174
  - nb. the algorithm is SHA, the standard is SHS
- based on design of MD4 with key differences
- produces 160-bit hash values
- 2005 results on security of SHA-1 raised concerns on its use in future applications
- Secure Hash Algorithm Std-SHA-512

How SHA Works?

- Digest Length=***160 bit***
- I/P Text=512 bit
- Sub Block size=32bit
- 512/32=16 total Sub blocks
- No. Of Rounds=4
- Iteration per round=***20***
- Chaining Variable = 5*32=160
- K[t] constant= ***Where t=0 to 79***
- O/P-> 160 bit hash

SHA Overview

1. ***Padding:*** Length of the message is 64 bits short of multiple of 512 after padding.
2. ***Append*** a 64-bit ***length*** value of original message is taken.
3. ***Divide the input into 512-bit blocks***
4. ***Initialise IV*** 5-word (160-bit) buffer (A,B,C,D,E) to
   (***A***=01 23 45 67,
   B=89 AB CD EF,
   C=FE DC BA 98,
   D=76 54 32 10,
   E=C3 D2 E1 F0)

5. ***Process Blocks*** now the actual algorithm begins. message in 16-word (512-bit) chunks:

Copy IV into single register for storing temporary intermediate as well as the final results.

Divide the current 512-bit blocks into 16 sub-blocks, each consisting of 32 bits.

- Has No. Of Rounds=4, each round consisting of 20 ***bit /step iteration*** operations on message block & buffer
- expand 16 words into 80 words(20*4) by mixing & shifting.K[t] constant= ***Where t=0 to 79***
- Form new buffer value by adding output to input.

6. output hash value is the final buffer value

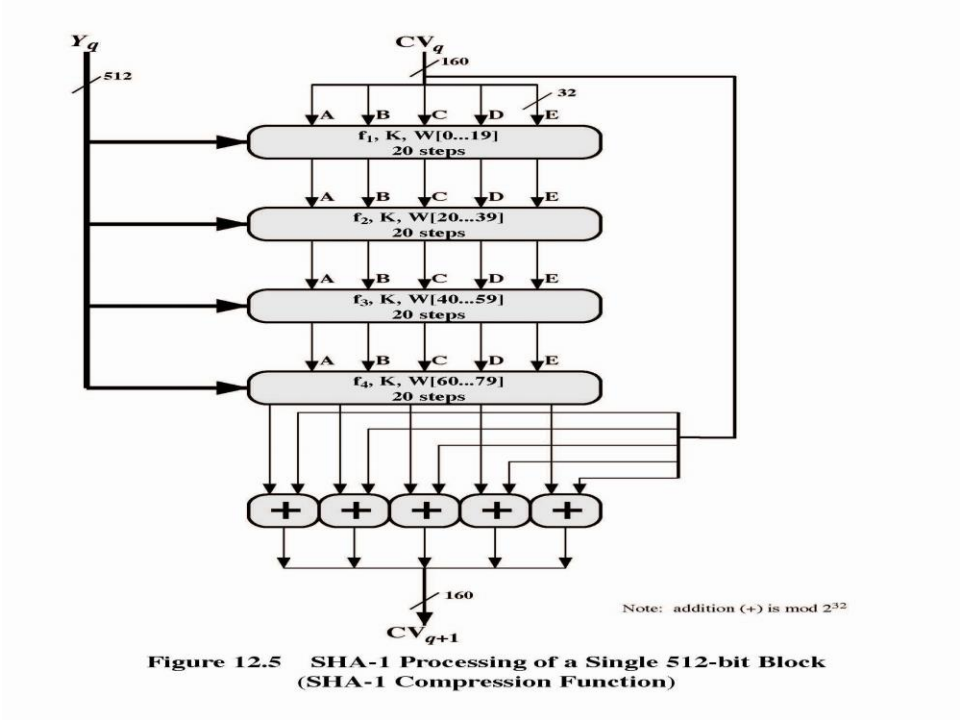
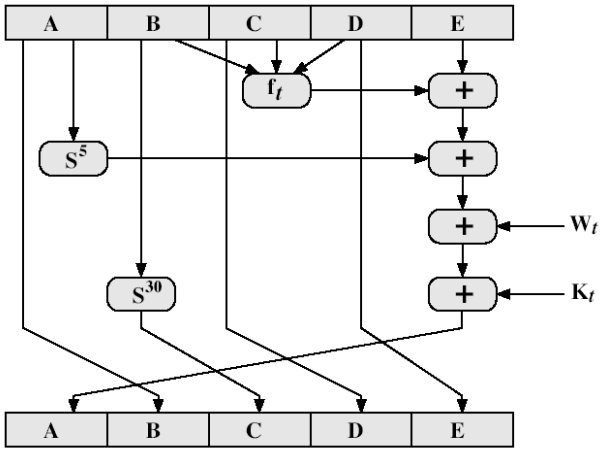


Figure 12.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

SHA-1 Compression Function



ABCDE=(F[t]+E+S5(A)+W[t]+K[t]),>>>Shift right by 1 bit for next iteration

SHA-1 Compression Function terms

- each round has 20 steps which replaces the 5 buffer words thus:
  $(A,B,C,D,E) \leftarrow (E+f(t,B,C,D)+(A<<5)+W_t+K_t),A,(B<<30),C,D)$
- ABCDE refer to the 5 words of the buffer
- t is the step number
- f(t,B,C,D) is nonlinear function for round
- W_t is derived from the message block
- K_t is a constant value
- S^t circular left shift of 32 bit sub-block by t bits

Process F(t) in each SHA-1 round

- where g can be expressed as:

ROUND 1: (b AND c) OR ((NOT b) AND (d)) same as MD5

ROUND 2: b XOR c XOR d

ROUND 3: (b AND c) OR (b AND d) OR (c AND d)

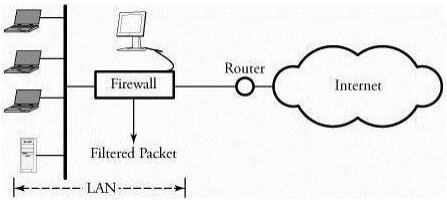
ROUND 4: b XOR c XOR d

c. Discuss about packet filtering and proxy server with respect to firewalls. (6 marks)

Firewalls

- As the name suggests, a firewall protects data from the outside world. A firewall can be a software program or a hardware device. A firewall a popular security mechanism for networks. A firewall is a simple router implemented with a special program. This unit is placed between hosts of a certain network and the outside world, as shown in Figure 5.18, and the rest of the network. The security issues faced by a smaller network like the one used at home are similar to larger networks. A firewall is used to protect the network from unwanted Web sites and potential hackers.

Figure 05.18. A simple configuration of a secured network using a firewall



- A firewall is placed on the link between a network router and the Internet or between a user and a router. The objective of such a configuration is to monitor and filter packets coming from unknown sources.
- Software firewall programs can be installed in home computers by using an Internet connection with these so-called gateways, the computer with such software can access Web servers only through this software firewall. But hardware firewalls are more secure than software firewalls. Moreover, hardware firewalls are not expensive. Some firewalls also offer virus protection. The biggest security advantage of installing a firewall in a business network is to protect from any outsider logging on to the network under protection.

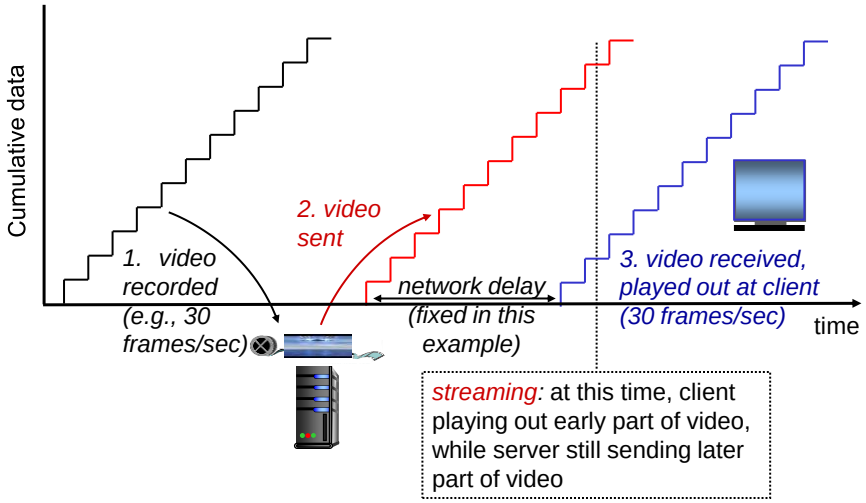
Module 5

9.a. What are the classification in multimedia network applications? Explain? (8 marks)

Multimedia networking: 3 application types

- *streaming, stored* audio, video
  - *streaming:* can begin playout before downloading entire file
  - *stored (at server):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *conversational* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype
- *streaming live* audio, video
  - e.g., live sporting event (futbol)

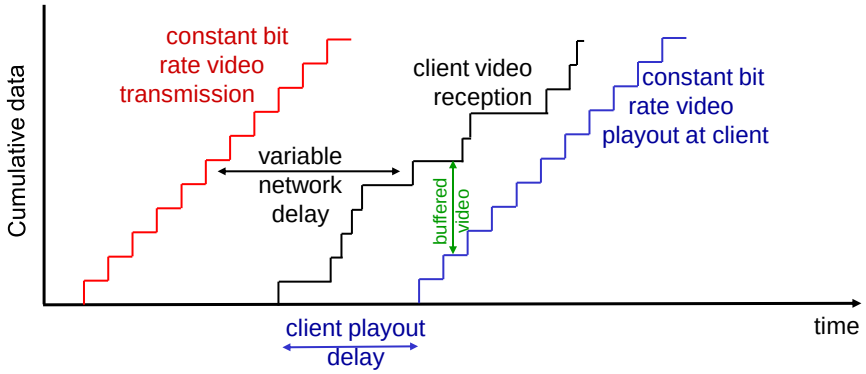
Streaming stored video:



Streaming stored video: challenges

- *continuous playout constraint*: once client playout begins, playback must match original timing
  - … but *network delays are variable* (jitter), so will need *client-side buffer* to match playout requirements
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted

Streaming stored video: revisted



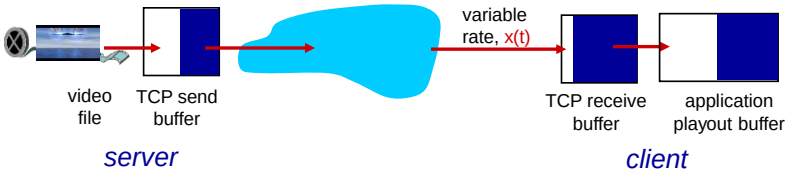
- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter

Streaming multimedia: UDP

- server sends at rate appropriate for client
  - often: send rate = encoding rate = constant rate
  - transmission rate can be oblivious to congestion levels
- short playout delay (2-5 seconds) to remove network jitter
- error recovery: application-level, timeipermitting
- RTP [RFC 2326]: multimedia payload types
- UDP may *not* go through firewalls

Streaming multimedia: HTTP

- multimedia file retrieved via HTTP GET
- send at maximum possible rate under TCP



- fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

Streaming multimedia: DASH

- *DASH: D*ynamic, *A*daptive *S*treaming over *H*TTP
- *server:*
  - divides video file into multiple chunks
  - each chunk stored, encoded at different rates
  - *manifest file:* provides URLs for different chunks
- *client:*
  - periodically measures server-to-client bandwidth
  - consulting manifest, requests one chunk at a time
    - chooses maximum coding rate sustainable given current bandwidth
    - can choose different coding rates at different points in time (depending on available bandwidth at time)

b. What are the two types of loss anticipation schemes? Explain? (7 marks)

VoIP: packet loss, delay

- *network loss:* IP datagram lost due to network congestion (router buffer overflow)
- *delay loss:* IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- *loss tolerance:* depending on voice encoding, loss concealment, packet loss rates between 1% and 10% can be tolerated

VoiP: recovery from packet loss (1)

Challenge: recover from packet loss given small tolerable delay between original transmission and playout

- each ACK/NAK takes ~ one RTT
- alternative: *Forward Error Correction (FEC)*
  - send enough bits to allow recovery without retransmission (recall two-dimensional parity in Ch. 5)

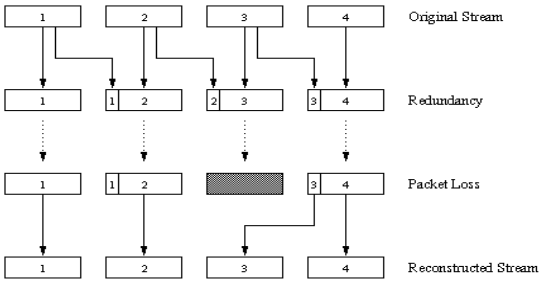
simple FEC

- for every group of n chunks, create redundant chunk by exclusive OR-ing n original chunks
- send $n+1$ chunks, increasing bandwidth by factor $1/n$
- can reconstruct original n chunks if at most one lost chunk from $n+1$ chunks, with playout delay

VoiP: recovery from packet loss (2)

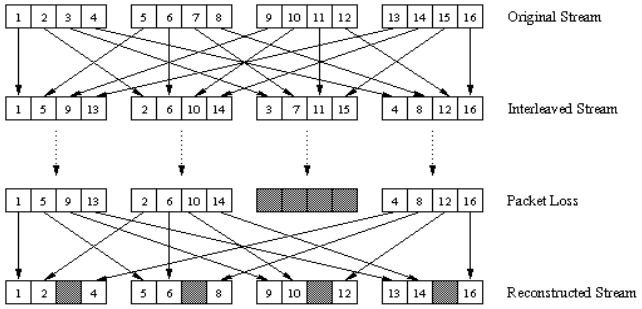
another FEC scheme:

- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
- e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps



- non-consecutive loss: receiver can conceal loss
- generalization: can also append (n-1)st and (n-2)nd low-bit rate chunk

VoiP: recovery from packet loss (3)

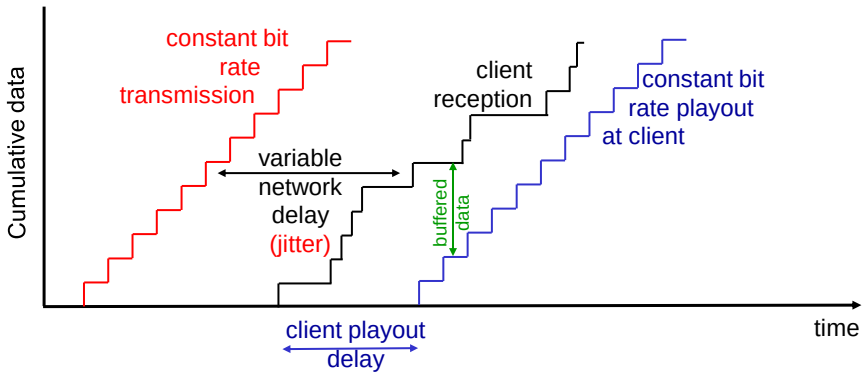


interleaving to conceal loss:

- audio chunks divided into smaller units, e.g. four 5 msec units per 20 msec audio chunk
- packet contains small units from different chunks
- if packet lost, still have *most* of every original chunk
- no redundancy overhead, but increases playout delay

c. What do you mean by a jitter and how to remove the jitter at the receiver for audio by fixed and adaptive playout delay? (5 marks)

Delay jitter



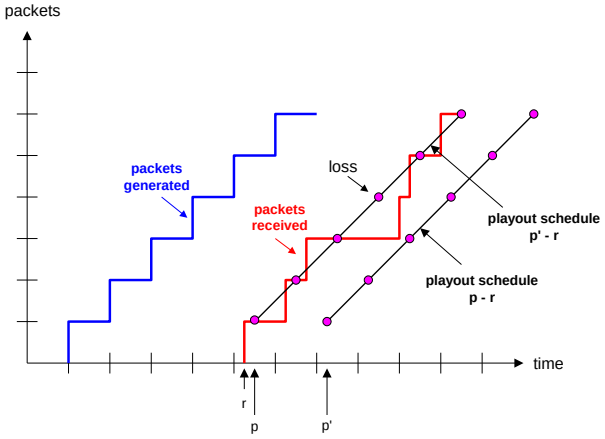
- end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

VoIP: fixed playout delay

- receiver attempts to playout each chunk exactly *q* msecs after chunk was generated.
  - chunk has time stamp *t*: play out chunk at *t*+*q*
  - chunk arrives after *t*+*q*: data arrives too late for playout: data "lost"
- tradeoff in choosing *q*:
  - *large q:* less packet loss
  - *small q:* better interactive experience

VoIP: fixed playout delay

- sender generates packets every 20 msec during talk spurt.
- first packet received at time *r*
- first playout schedule: begins at *p*
- second playout schedule: begins at *p'*



Adaptive playout delay (1)

- *goal:* low playout delay, low late loss rate
- *approach:* adaptive playout delay adjustment:
  - estimate network delay, adjust playout delay at beginning of each talk spurt
  - silent periods compressed and elongated
  - chunks still played out every 20 msec during talk spurt
- adaptively estimate packet delay: (EWMA - exponentially weighted moving average, recall TCP RTT estimate):

$$d_i = (1-\alpha)d_{i-1} + \alpha\,(r_i - t_i)$$

delay estimate after ith packet

small constant, e.g. 0.1

time received - time sent (timestamp)

measured delay of ith packet

Adaptive playout delay (2)

- also useful to estimate average deviation of delay, v_i :

$$v_i = (1-\beta)v_{i-1} + \beta\,|r_i - t_i - d_i|$$

- estimates d_i, v_i calculated for every received packet, but used only at start of talk spurt

- for first packet in talk spurt, playout time is:

$$playout\text{-}time_i = t_i + d_i + Kv_i$$

remaining packets in talkspurt are played out periodically

Adaptive playout delay (3)

Q: How does receiver determine whether packet is first in a talkspurt?

- if no loss, receiver looks at successive timestamps
  - difference of successive stamps > 20 msec -->talk spurt begins.
- with loss possible, receiver must look at both time stamps and sequence numbers
  - difference of successive stamps > 20 msec *and* sequence numbers without gaps --> talk spurt begins.

10. a. Explain the working of CDN. (8 marks)

Content distribution networks

- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

- *option 1:* single, large "mega-server"
  - single point of failure
  - point of network congestion
  - long path to distant clients
  - multiple copies of video sent over outgoing link

….quite simply: this solution *doesn't scale*

Content distribution networks

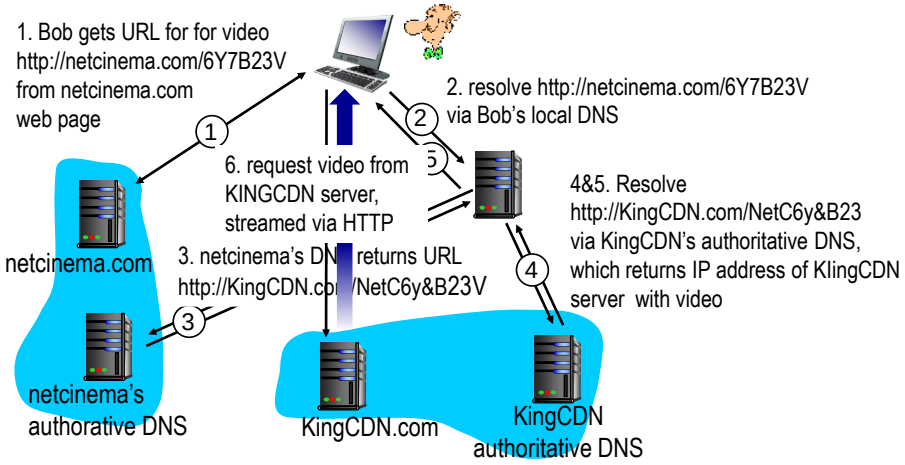
- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

- *option 2:* store/serve multiple copies of videos at multiple geographically distributed sites *(CDN)*
  - *enter deep:* push CDN servers deep into many access networks
    - close to users
    - used by Akamai, 1700 locations
  - *bring home:* smaller number (10's) of larger clusters in POPs near (but not within) access networks
    - used by Limelight

CDN: "simple" content access scenario

Bob (client) requests video http://netcinema.com/6Y7B23V

- video stored in CDN at http://KingCDN.com/NetC6y&B23V



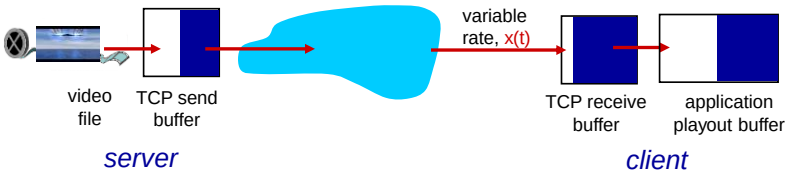
CDN cluster selection strategy

- *challenge:* how does CDN DNS select "good" CDN node to stream to client
  - pick CDN node geographically closest to client
  - pick CDN node with shortest delay (or min # hops) to client (CDN nodes periodically ping access ISPs, reporting results to CDN DNS)
  - IP anycast
- *alternative:* let *client* decide - give client a list of several CDN servers
  - client pings servers, picks "best"
  - Netflix approach

b. Explain about HTTP streaming in case of streaming stored video. (7 marks)

Streaming multimedia: HTTP

- multimedia file retrieved via HTTP GET
- send at maximum possible rate under TCP

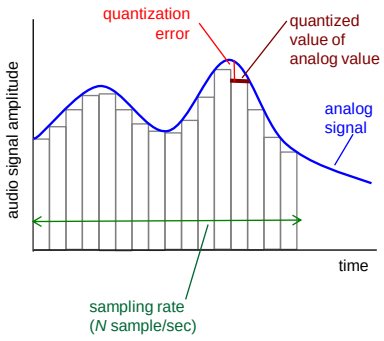


- fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

c. Discuss about the properties of audio and video in multimedia networking. (5 marks)

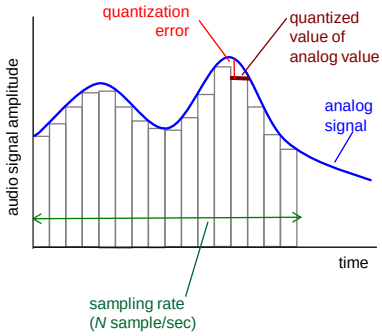
Multimedia: audio

- analog audio signal sampled at constant rate
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- each sample quantized, i.e., rounded
  - e.g., 2^8=256 possible quantized values
  - each quantized value represented by bits, e.g., 8 bits for 256 values



Multimedia: audio

- example: 8,000 samples/sec, 256 quantized values: 64,000 bps
- receiver converts bits back to analog signal:
  - some quality reduction



example rates

- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up

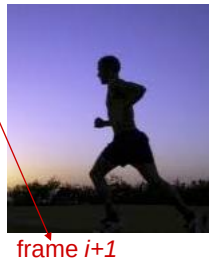
Multimedia: video

- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- coding: use redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)

frame *i*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i



frame *i+1*