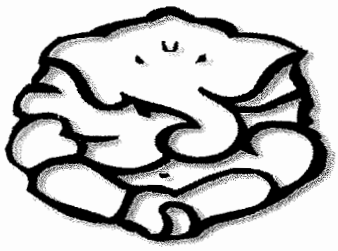


6TH CSE/ISE

COMPUTER NETWORKS 2

ASHOK KUMAR K
VIVEKANANDA INSTITUTE OF TECHNOLOGY
MOB: 9742024066
e-MAIL: celestialcluster@gmail.com

S. K. L. N ENTERPRISES
Contact: 9886381393

UNIT 1 PACKET SWITCHING NETWORKS- I

Syllabus.

* Network services and internal n/w operations.
* packet n/w Topology
* Datagrams and virtual circuits.
* Routing in packet networks.
* shortest path routing.
* ATM networks.

- 6 Hours.

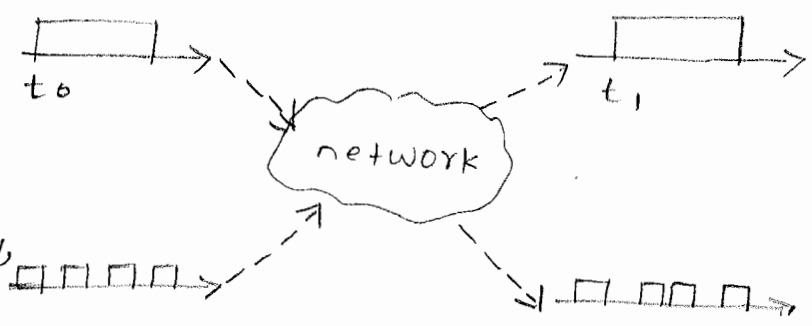
* Telephone network operates on circuit switching
* Here, complete resource is allocated to the transmitter and receiver. It can not be utilized by other users.
* inefficient when amount of information is small.
* packet-switching networks → Transfer blocks of information called packets.
* packet networks is viewed in two perspectives.

i) External view, concerned with
  * setting up of a connection
  * Transfer of data with quality of service.

ii) Internal view, concerned with
  * physical Topology of network
  * Interconnection of links, switches, and routers.
  * Addressing and routing procedures
  * congestion inside network.
  * managing Traffic

NETWORK SERVICES AND INTERNAL NETWORK OPERATIONS

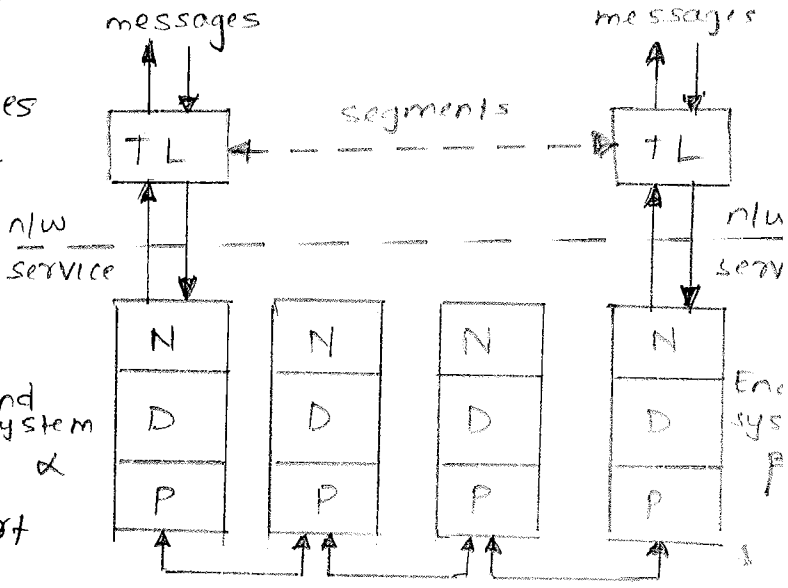
* the essential function of a network is to transfer information among the users that are attached to the network or internetwork.
* This transfer may be single block or sequence of blocks.
* In case of single block, we are interested in Block delivered correctly, and delay experienced in traversing the network.



* In case of sequence of blocks, we are also interested in sequence of blocks delivered

* Fig shows a transport layer protocol that operates end to end across a n/w.

* The Transport layer peer processes at the end systems accept messages from their higher layer and transfer these messages by exchanging segments end to end across the n/w.



* Fig shows the interface at which the n/w service is visible to the transport Layer

* Network services can be
  - → connection-oriented
  - → connectionless.

connection less service.

* very simple, with only two basic interaction ~~with~~ b/w the transport layer (user of service) and network layer (provider of service)
* user can request transmission of packet at any time and does not need to inform the n/w layer that the user intends to transmit information ahead of time.
* This service puts total responsibility for
  - → error control
  - → sequencing
  - → Flow control on end-system transport layer.

Connection-oriented service

* Transport layer cannot request transmission of information until a connection b/w end systems has been set up.
* the network layer must be informed about the new flow

* network layer maintains
  - → state information
  - → connection set up
  - → parameters related to usage and QoS
  - → connection release procedure.
    - ↳ required to terminate the connection.

* Network layer provides following services
  - i.) Best effort connection less service
  - ii.) Low delay connection less service
  - iii.) Connection oriented reliable stream service.
  - iv) connection oriented transfer of packets with delay and bandwidth guarantees.

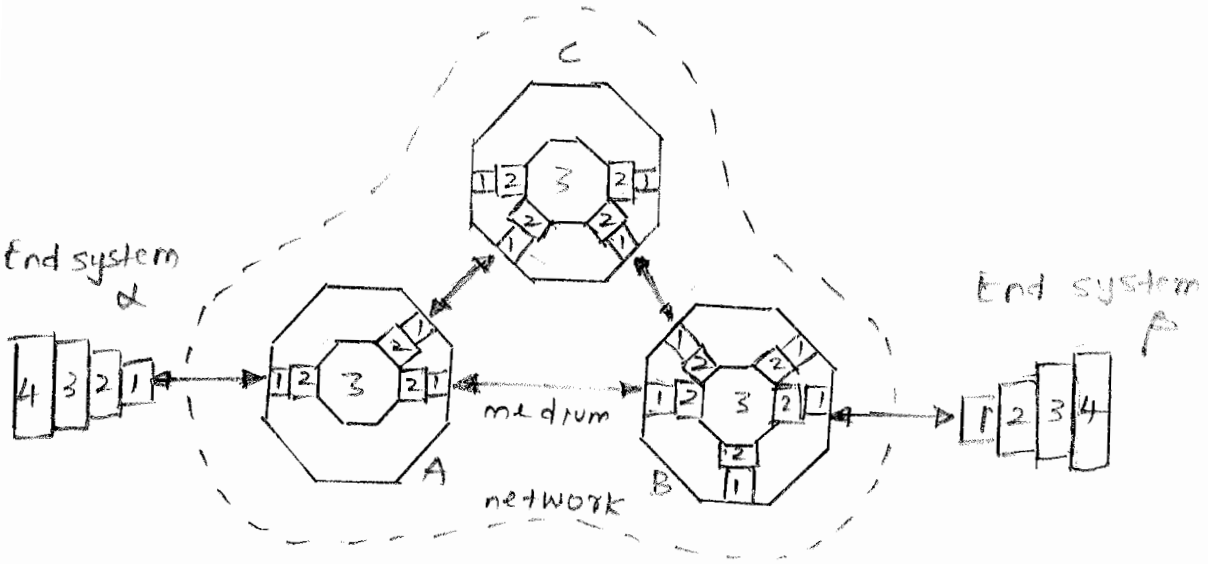
* Two inter related reasons to chose n/w services are
  - → End-to-End argument
  - → N/w scalability.

* End-to-End argument suggests that functions should be placed as close to the applications as possible, since it is the application that is in the best position to determine whether a function is being carried out completely and correctly.
  This suggests that as much functionality as possible should be located in the transport layer or higher layer and that the n/w services should provide the minimum functionality required to meet application performance.

Internal operation of the network.

* Fig below shows the relation b/w the service offered by the network and the internal operation.
* We say that the internal operation of a n/w is connection-less if packets are transferred within the network as datagrams.
  Thus, in the figure, each packet is routed independently, and may follow different paths from α to β & so may arrive out of order.

* We say that the internal operation of a n/w is connection-oriented if packets follow a virtual circuit along a forward path that has been established from source to destination.



1 Physical Layer entity

2 DataLink Layer entity

3 n/w Layer entity

4 Transport layer entity

(3) N/w Layer entity

fig: Layer 3 entities work together to provide network service to Layer 4 entities.

* Functions to be carried out at every node in the network must be in the Network layer.
* Thus functions that route and forward packets needs to be done in the n/w Layer.
* priority and scheduling functions also need to be in network Layer. ↳ directs how packets are treated (forwarded) in a node to ensure QoS.
* Functions that belong in the edge should (if possible) be implemented in the transport layer or higher
* Another set of functions is concerned with making the

PACKET NETWORK TOPOLOGY

* computers located in subscribers home are connected to an access mux located in the service provider n/w

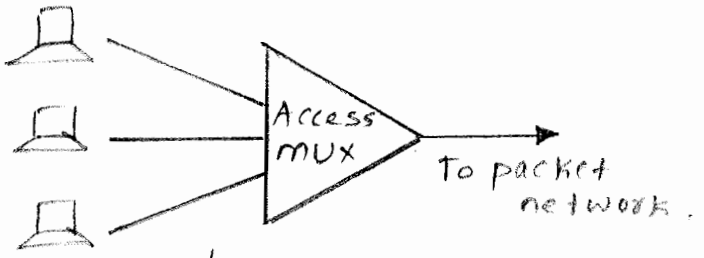


fig: Access mux.

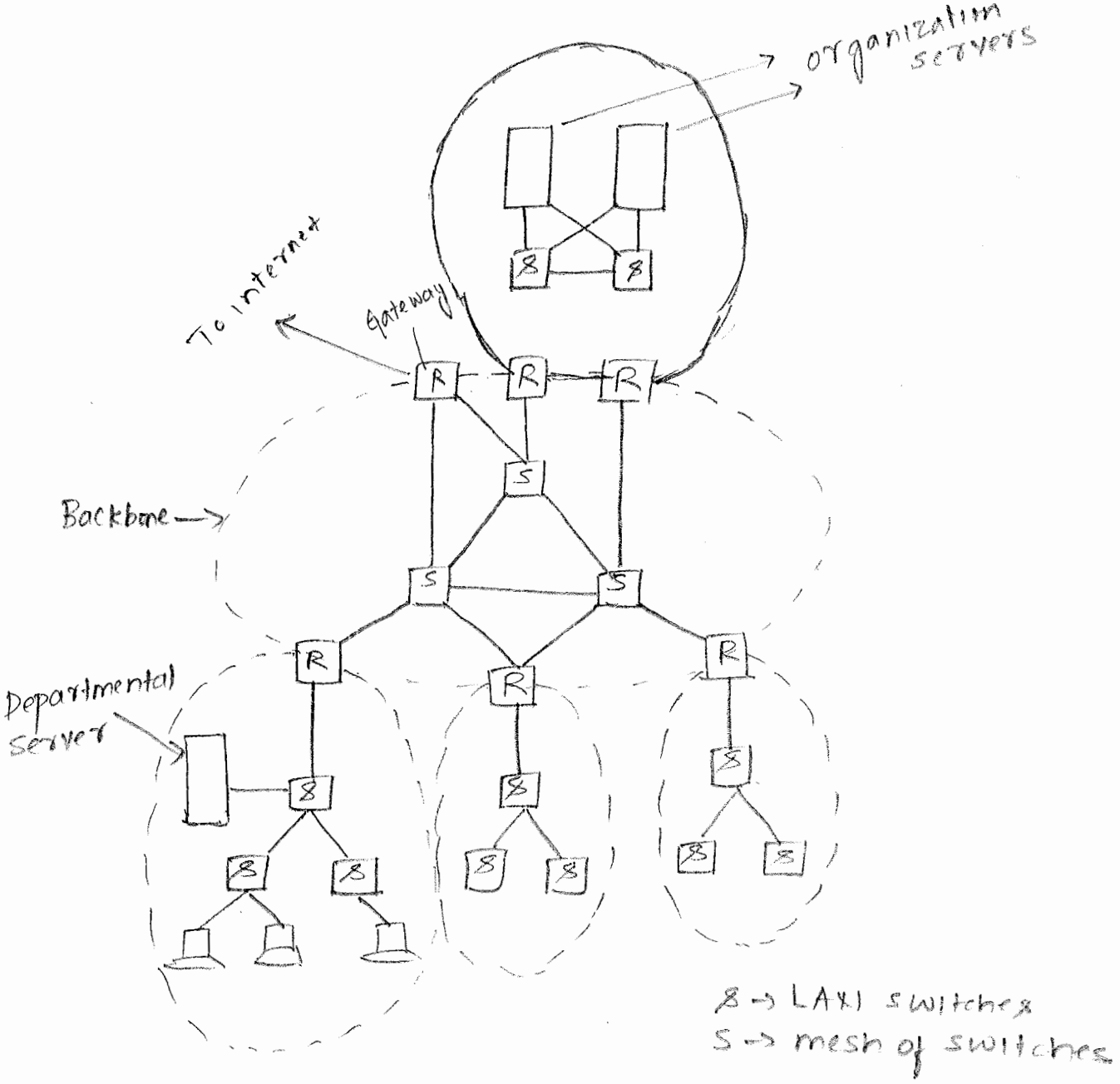
eg: Digital subscriber loop access mux (DSLAM) located in Telephone central office.

* computers are connected to DSLAM via an ADSL (Asymmetric digital subscriber line) modem.
* Multiple users may share the same Transmission Line to the access mux, resulting in a point-to-multipoint topology.
* Main function of access mux is to combine the bursty traffic flows from the individual computers. By this, efficiency of the line is increased.
* If a subscriber has multiple computers connected to the same access mux, another level of mux is used known as home network.
* Home network aggregates traffic from multiple computers and perform and performs a Network Address Translation (NAT) function
* Need for NAT: service provider assigns a single global network address to the subscriber to conserve address space. To accomodate multiple computers in the home network, the subscriber assigns a private n/w address that is only defined within the subscriber home network to each computer.
* The purpose of the gateway is to translate the private n/w address of each packet to the global n/w address when a packet leaves the home n/w, & vice versa when a packet arrives at the home n/w.

* The address translation usually uses higher layer information such as service access point (SAP)

* Local Area Networks (LANs) also provide the access to packet-switching networks.

* Fig shows a structure of a campus network that interconnects multiple LANs in an organization.



* LANs for a large group of users such as a department are interconnected in an extended LAN through the use of LAN switches (denoted by S)

* Resources such as servers and data buses that are primarily of use to this department are kept within the sub network.
* This approach reduces delay in accessing the resources.
* Each sub n/w has access to the rest of the organization through router R, that accesses the campus backbone network.
* A subnetwork uses campus backbone to reach the outside world such as the internet or other sites, through a border router.
* Depending upon the type of organization, the border router may implement firewall functions to control the traffic that is allowed into and out of the campus network.
* Servers containing critical resources that are required by the entire organization are located in a data center, with well maintainence and security. They are placed near to the backbone network to minimize the number of hops required to access them from the rest of the organization.
* The routers in the campus network are interconnected to form the campus backbone network depicted by the mesh of switches (S) eg: high speed LANs, Gigabit Internet, or ATM n/w.
* Routers use IP, which enables them to operate over various data links and n/w technologies.
* Router exchange information about the state of their links to dynamically calculate routing tables that direct packets across the campus network.
* This allows the n/w to adapt to changes in topology due to faults in transmission links or equipments.

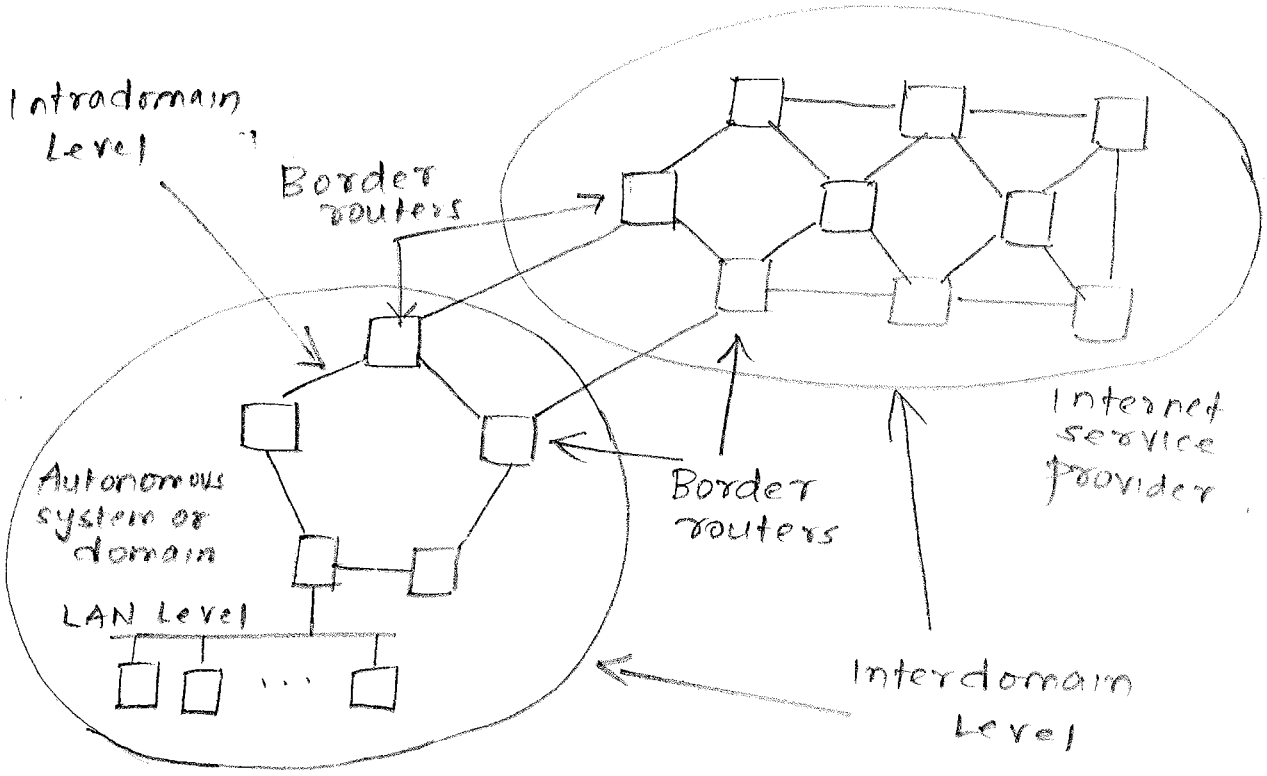
* Routers in the campus network may form a domain or autonomous system.

→ Domain means routers run the same routing protocol

→ The term autonomous system is used for one or more domains under a single administration.

* All routing and policy decisions inside the autonomous system are independent of any other network.

Intra domain and Inter domain Levels



* A campus network can be interconnected through routers interconnected by leased digital transmission lines or frame relay connections.

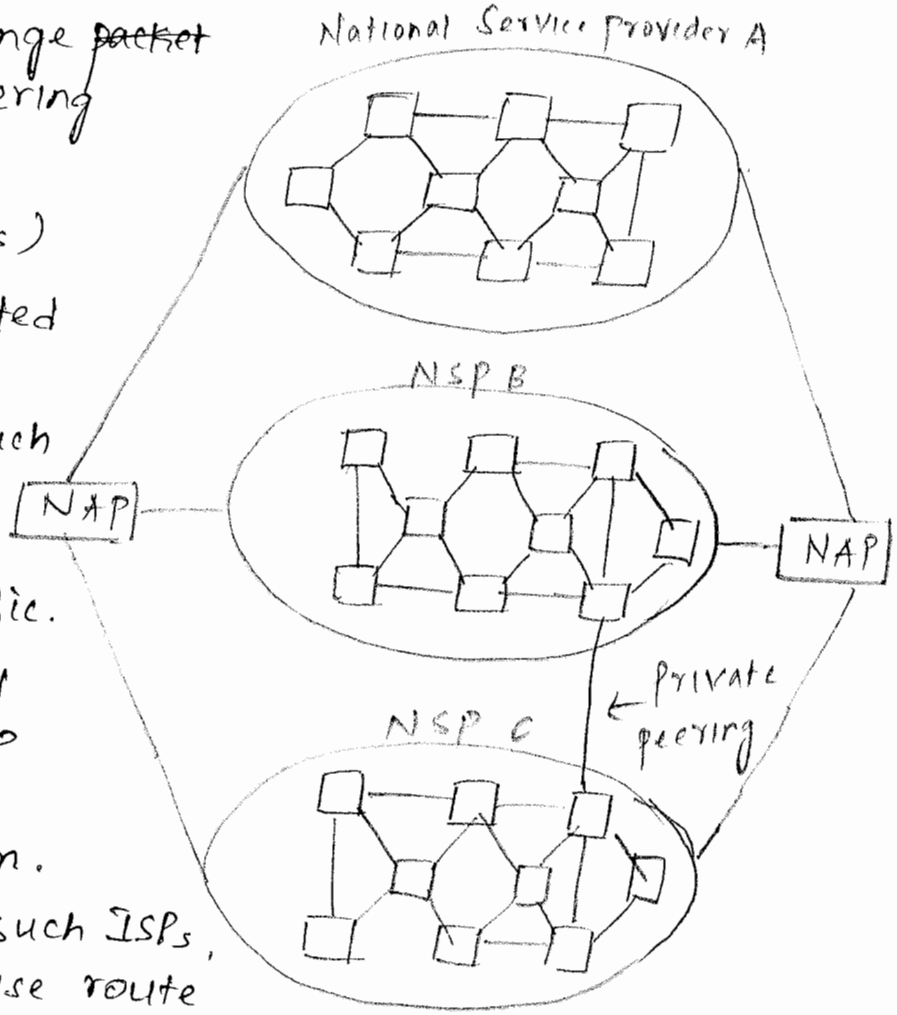
* Campus n/w may be connected to an ISP through one or more border routers.

* To communicate with other n/ws, the autonomous system must provide information about its network routes

* Border router communicates on an interdomain level.
* Other router operates at the intradomain level.

National ISP

* A national ISP provides point of presence (POP's) in various cities where customers can connect to their network.
* ISP has its own national backbone network for interconnecting its POPs. Backbone n/w could be an ATM or other technology.
* ISP inturn exchange ~~packet~~ traffic at public peering points called n/w access points (NAPs)
* NAP → is a co-located set of high speed routers through which routers from different ISPs can exchange traffic.
* ISPs interconnected to a ~~network~~ NAP need to exchange routing information.
* If there are n such ISPs, then $\frac{n(n-1)}{2}$ pairwise route exchanges are required. This peering relationship poses a scalability problem as no. of ISPs becomes very large.

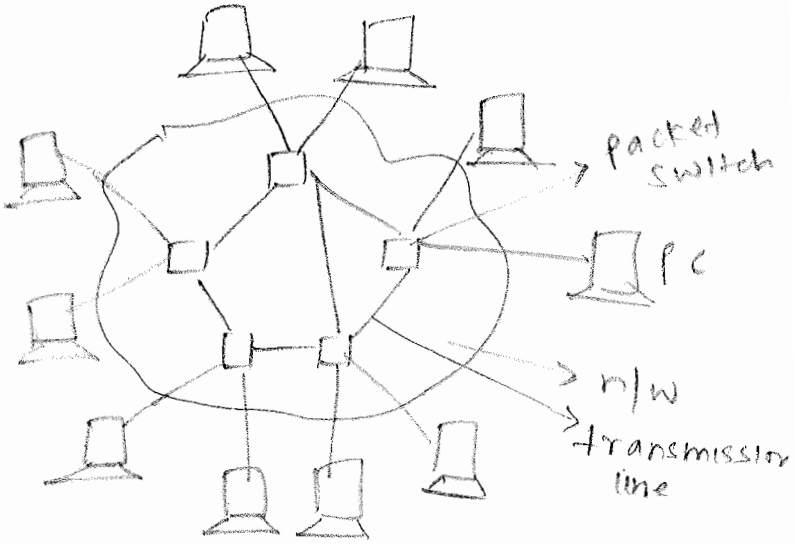


* A route server is used to solve problem, each ISP sends routing information to the route server, which knows the policies of every ISP. The route server inturn delivers the processed routing information to the ISPs.

* Nowadays, most major national ISPs increasingly use private peering points connecting two ISPs directly to exchange traffic.

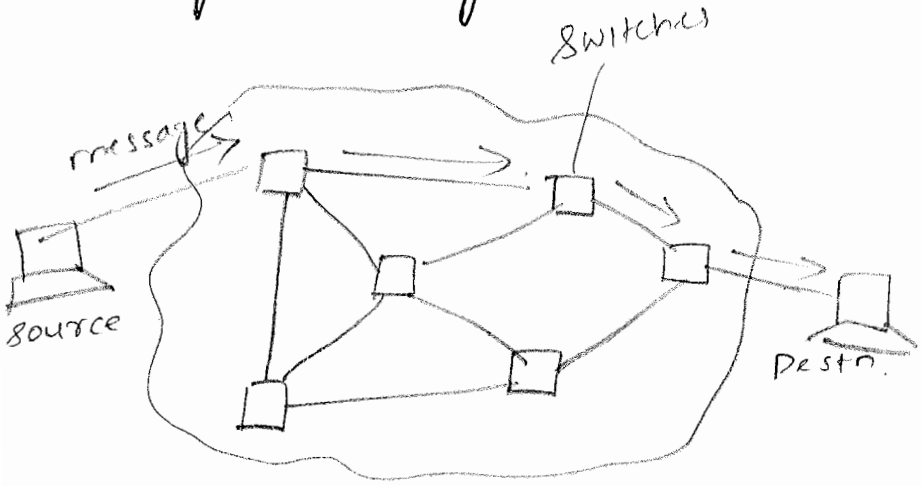
DATAGRAMS AND VIRTUAL CIRCUITS

* Fig shows switched network.



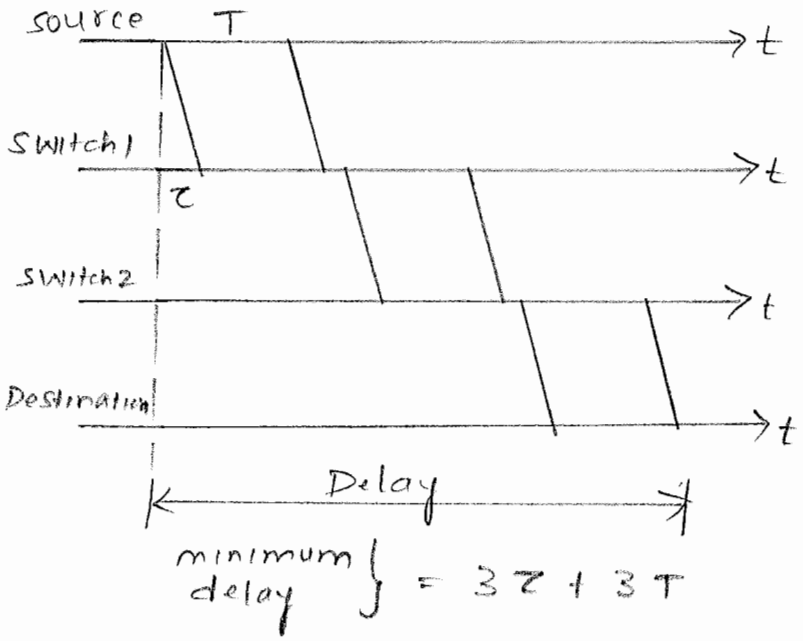
Connectionless packet switching

Message switching.



* message is transmitted from one switch to other.
* operates in store and forward fashion (ie receive complete and store, then forward to next switch)
* Each message has header (source addr, Destn addr, CRC bits)
* Each switch performs error check, if no errors,

* If errors are detected, retransmission is requested (ARQ)
* Does not involve call set up.
* High utilization of transmission line.
* Delay increases.
* Lose of message when a switch has inefficient buffering to store.



long message v/s packets

Large message $L = 10^6$ bits

Two hops are used.

Each hop has a error rate of $p = 10^{-6}$

Each hop does error checking & retransmission.

How many bits need to be transmitted using message switching.

Soln: $P_c = (1-p)^L \simeq e^{-LP}$

$= (1-10^{-6})^{10,00,000}$

$\simeq e^{-1} = 1/3$

on an avg, it will take 3 tries to get the message over the first hop. IIIrly will require 3 full message transmissions for second hop.

Thus 6 Mbits will need to be transmitted to get the 1 M bit message across two hops.

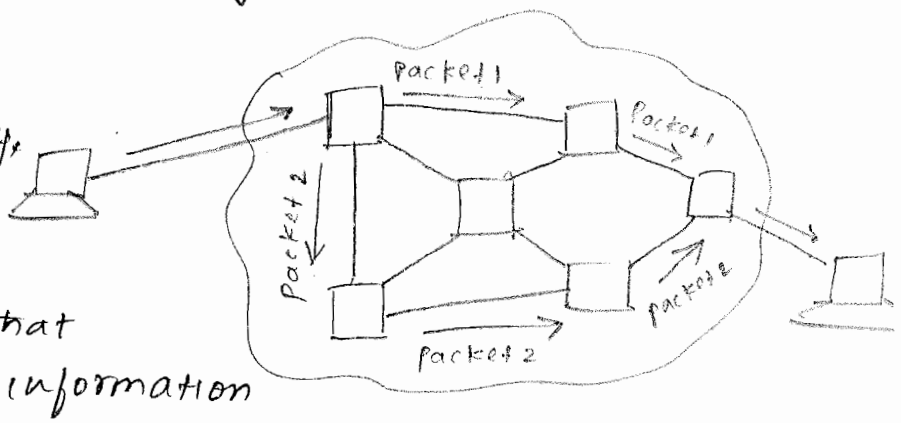
Suppose, message is broken down into 10^5 bit packets

probability that a packet arrives correctly after first hop is

$$P_c' = (1-10^{-6})^{10,0000}$$
$$= e^{-1/10}$$
$$\simeq 0.90$$

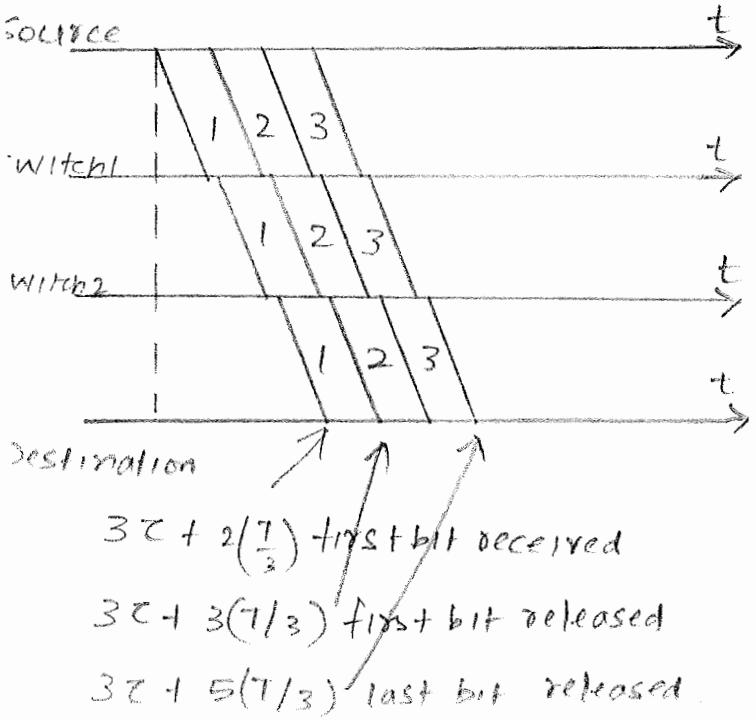
* probability of error increases with the block length.
* Long message means, Large rate of retransmission.
* Thus, we place a limit on max size of the blocks.
* Message switching is not suitable for interactive applications because of delays.
* packet switching reduces delays.

Datagram packet switching



* Each packet is routed independently, through the n/w.
* Each packet has an attached header that provides all of the information required to route the packet to its destination.
* When a packet arrives at packet switch, the destination address in the header are examined to determine the next hop in the path to the destination.
* The packet is made to wait in a queue until a transmission line becomes free.
* By sharing transmission line among multiple packets, packet switching can achieve high utilization at the

* Each packet is routed independently and may be received in out of order and resequencing may be required at the destination.



* Message is broken into three separate packets.
* Here we assume that three packets follow the same path and are transmitted in succession
* Suppose each packet requires $P = T/3$ seconds to transmit, the first packet arrive at the first switch after $\tau + P$ seconds

* First packet is received at the second packet switch at time $2\tau + 2P$
* First packet arrives at the destination at time $3\tau + 3P$.
* In the absence of transmission errors, the final packet will arrive at the destination at time,

$$3\tau + 3P + 2P$$
$$= 3\tau + 5P$$
$$= 3\tau + T + 2P$$ which is less than the delay incurred in message switching.

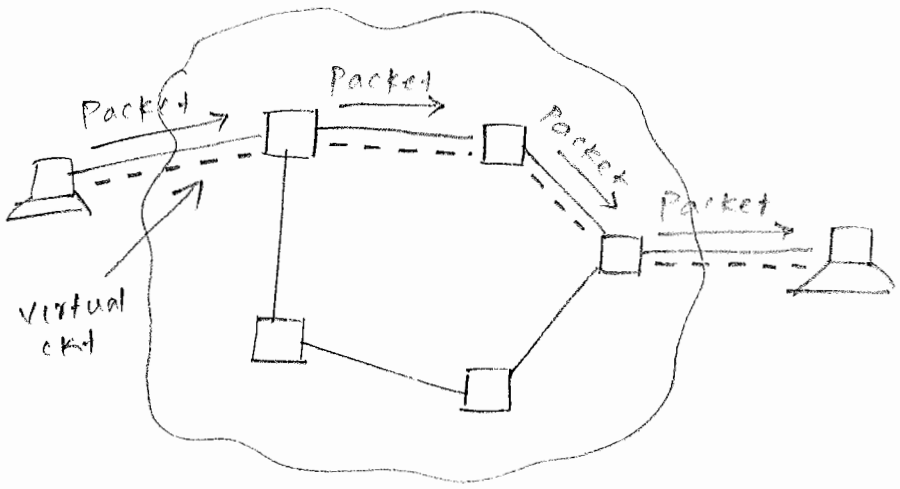
* In general, if the path followed by a sequence of packets consists of L hops with identical propagation delays and transmission speeds, then delay incurred by a message that consists of k packets is given by $L\tau + LP + (k-1)P$
In contrast, delay incurred using message switching is
$L\tau + LT = L\tau + L(k \cdot P)$
Thus message switching involves an additional delay of
$(L-1)(k-1)P$

Routing table

Dest Addr	O/P port
⋮	⋮
0785	7
⋮	⋮
1345	12
⋮	⋮

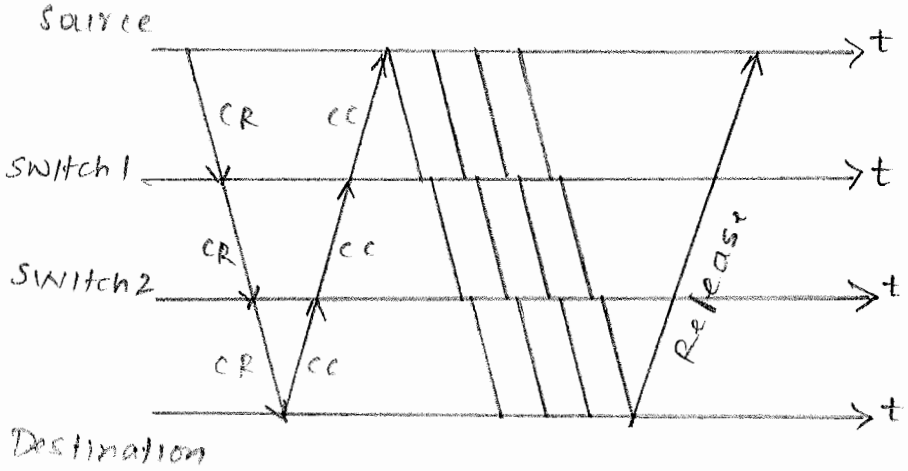
Virtual packet switching

* Establishes a virtual circuit or a connection between a source and a destination prior to the transfer of packets.
* In circuit switching, circuits reside at a physical layer where as virtual circuits reside at the n/w layer.



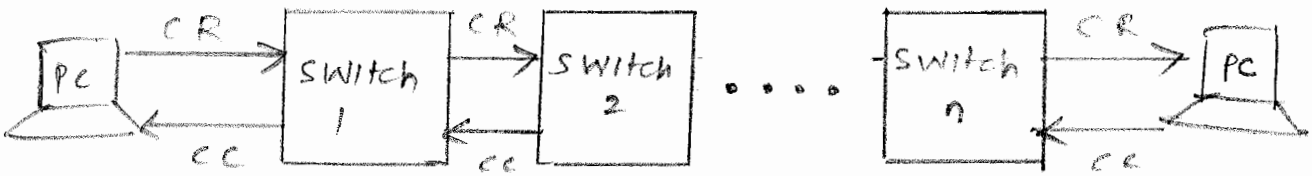
* minimum delay in virtual circuit packet switching is similar to that in datagram packet switching, except for an additional delay required to setup the virtual circuit.

* Determines a path through the n/w and then set parameters in the switches by exchanging connect-request and connect-confirm messages.
* Every switch along the path is involved in the exchange of signalling messages to set up the virtual circuit
* If a switch does not have enough resources to setup a virtual ckt, the switch alternately responds to a CR message ~~by~~ with a



Delays in virtual circuit packet switching

* In virtual circuit packet switching, the buffer and transmission resources need not be dedicated explicitly for the use of the virtual circuit, but the number of flows admitted may be limited to control the load on certain links.



Signalling message exchange in call setup.

* In datagram packet switching, each packet must contain the full address of the source and destination. In large n/ws these address can require a large no. of bits and result in significant packet overhead and hence wasted transmission band width.
* Advantage of virtual ckt packet switching is that abbreviated headers are used. The call set up procedure establishes a number of entries in routing tables located in the various switches along the path.
* At the input to every switch, the virtual circuit is identified by a virtual ckt identifier (VCI). When a packet arrives at an input port, the VCI in the header is used to access the table. The table look up provides the

output port to which the packet is to be forwarded and the VCI that is to be used at the input port of the next switch.

* Thus, the call set up procedure sets up a chain of pointers across the n/w that directs the flow of packets in a connection.
* The table entry for a VCI can also specify the type of priority that is to be given to the packet by the schedular that controls the transmission in the next output port.

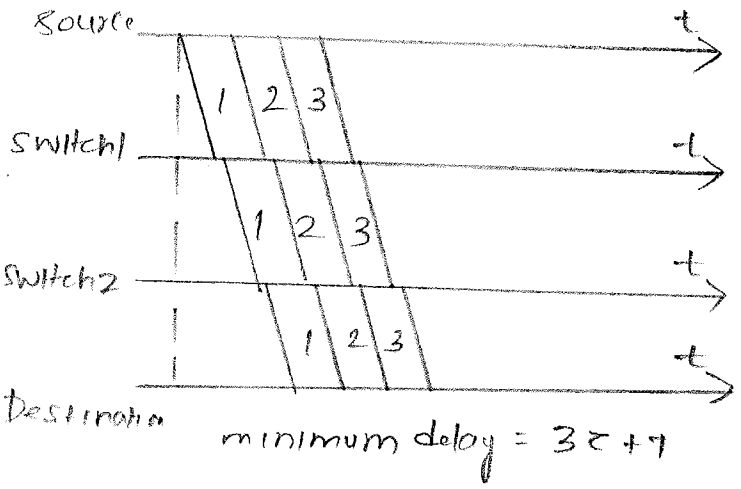
I/P VCI	O/P VCI	O/P VCI
12	13	44
15	15	23
27	13	16

* number of bits required for addressing reduces.
* use of abbreviated headers and h/w based table lookup allows fast processing and forwarding of packets.
* Virtual circuit packet switching can do a table lookup through direct indexing which is fast
* During call setup, some resources such as buffers may be ~~required~~ reserved for a virtual ckt at every switch along the path and certain amount of bandwidth can be allocated at each link in the path.

Disadvantages

* The switches in the network need to maintain information about the flows that pass the switches.
* When failure occurs in the network all affected connections must be setup again.

* modification of virtual circuit packet switching is called cut through packet switching can be adopted when retransmission are not used in the underlying datalink control.
* A packet is forwarded as soon as the header is received and the table look up is carried out.



* cut through packet switching may be used for speech transmission which has a delay requirement but can tolerate some errors.
* Will be appropriate when the transmission is virtually error free as in optical fibre transmission.
* Hop by hop error checking is unnecessary.

Structure of a packet switch

* Packet switch performs two functions
  - → Routing
  - → forwarding
* Routing function uses algorithms to find a path to each destination and stores the result in a routing table.
* Forwarding function processes each incoming packet from an i/p port and forwards the packet to the appropriate o/p port based on the information stored in routing table.

* Generic packet switch consists of i/p ports, o/p ports, an interconnection fabric, and a switch controller.
i/p port and o/p port are normally paired.

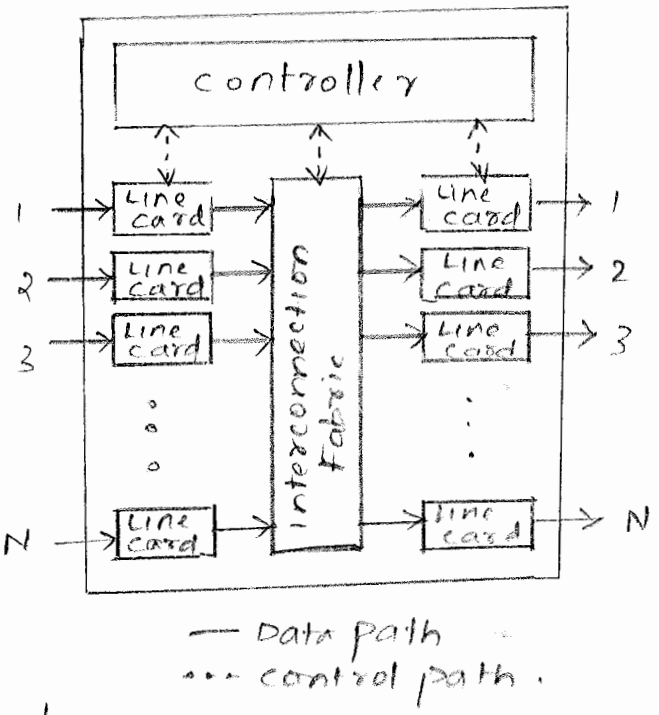


fig: components of generic packet switch

Line card

* A Line card often contains several i/p/o/p ports so that the capacity of the link connecting the line card to the interconnection fabric (typically of high speed) is fully utilized.

* Line card implements physical and datalink layer functions, as well as certain n/w Layer functions, like symbol timing, linecoding, framing, physical layer addressing, and error checking.

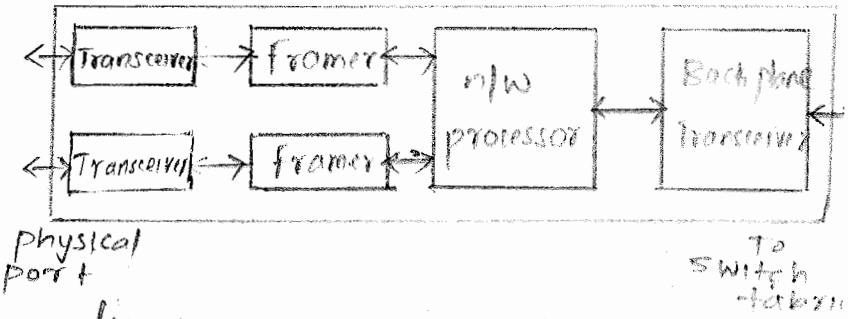


fig: organisation of a line card

* To handle broadcast ~~address~~ networks, the line card may also support a medium access control protocol.
* n/w layer routing tables may also reside in the line card
* Finally, Line card also contains buffers and associated scheduling algorithms.

Controller : controller in a

* packet switch contains general purpose processor to carry out a number of control and management functions depending on the type of packet switching.

For ex:

→ controller in connectionless mode, executes routing protocols

→ controller in connection oriented mode handles signalling messages.

* As a central coordinator, the controller also communicates with each line card and the interconnection fabric so that various internal parameters can be configured and maintained.

Interconnection fabric

* Its function is to transfer packets b/w the line cards.

* Interconnection fabric becomes bottleneck if there are many high speed line cards, since all the traffic from i/p line card have to go through interconnection fabric.

→ A bus type interconnection fabric (where packets are transferred serially from i/p to o/p ports) does not scale to large sizes since the speed of bus has to be N times faster than the port speed.

→ A cross bar interconnection fabric can transfer packets in parallel b/w i/p ports and o/p ports.

* For packet switching, buffers need to be added to the cross bar to accomodate packet contention. Buffers can be located at i/p ports or o/p ports as shown.

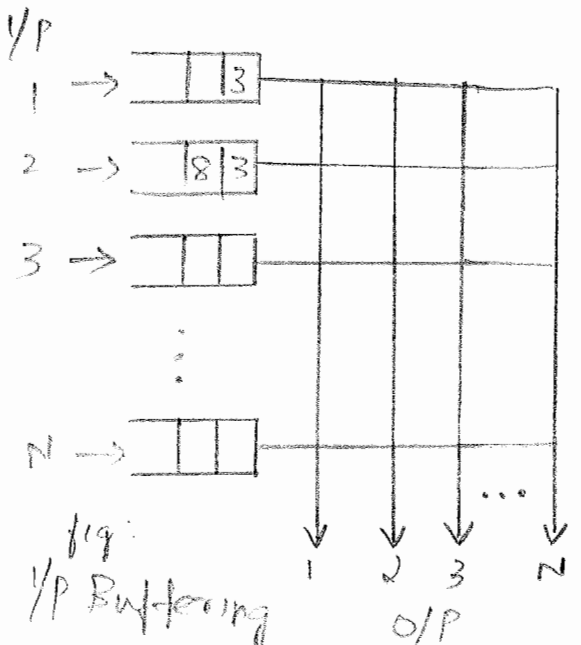


fig: I/P Buffering

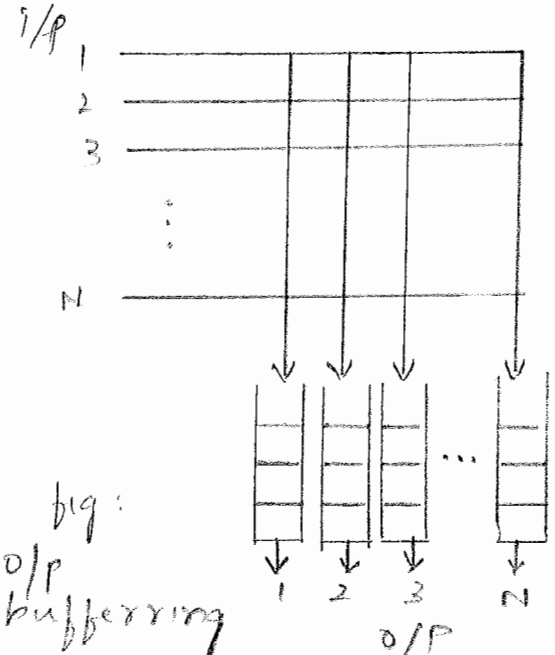


fig: O/P bufferring

* A crossbar with output buffering needs to run N times faster than the port speed since up to N-packets may simultaneously arrive at a particular output.
* Because only one packet is allowed to proceed to a particular o/p with the i/p buffering case, the crossbar does not need a speed up. However i/p buffering causes another problem called HOL blocking.
  Definition: problem of the first packet holding back other subsequent packets behind it is called Head-Of-Line (HOL) Blocking. (refer fig)
  HOL Blocking causes performance degradation of cross bar
* one way to eliminate HOL Blocking is to provide N-separate input buffers at each i/p port so that each i/p buffer is dedicated to a particular o/p.

Banyan Switch

* complexity of the crossbar is more (N^2) and is undesirable for building large switches.
  Multistage architectures have been considered as a sol^n to building a large switch. one such architecture for packet switching is called a Banyan switch, as shown.

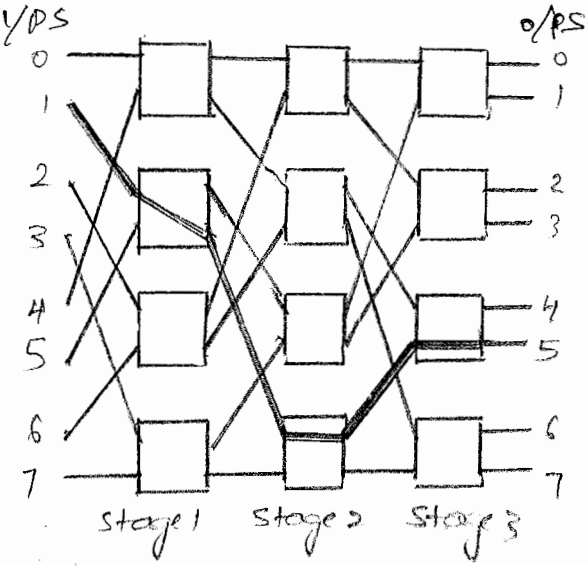


fig: 8x8 Banyan switch

* Banyan switch typically composed of 2x2 switching elements interconnected in a certain fashion such that exactly one path exists from each i/p to each o/p.
* Routing can be done in a distributed manner by appending the binary address of the o/p number to each packet and by having each switching element at

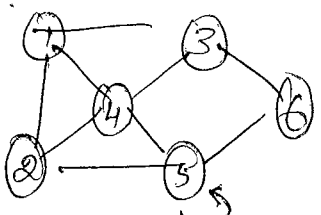
* If the bit is 0, switching element should steer a packet to its upper output; otherwise to lower output.
eg: If i/p 1 would like to send a packet destinated to o/p 5 with address 101 in binary.
The switching element at stage 1 looks at first bit of the addr & steers the packet to its lower o/p. The packet then arrives at bottom switching element at stage 2, which steers it to its upper o/p since second bit is 0. Finally switching element at stage 3 steers the packet to its lower o/p & sends the packet to o/p 5.

* If there is another packet from i/p 5 that would like to go to o/p 7 at the same time, this packet will contend for the same o/p of the switching element at stage 1. This behaviour causes the banyan switch to block packets even though some o/ps are idle.
* one way to prevent packets from being blocked is to provide buffering at each switching element so that contending packets may be temporarily stored at a local buffer. When the local buffer is full, the associated switching element can send a back pressure signal, notifying the upstream switching elements to stop sending packets.

ROUTING IN PACKET NETWORKS

* Routing is a major component of n/w Layer & is concerned with the problem of determining feasible paths (or routes) for packets to follow from each source to each destination.

*



node
(switch/router)

A packet could take one of several possible paths from node 1 to node 6

1-3-6
1-4-5-6
1-2-5-6.

Best path is selected based on
→ minimum hops
→ less delays
→ more bandwidth

goals of Routing Algorithm

1.) Rapid and accurate delivery of packets.
   → takes less time to reach dest^n
   → finds a path to correct dest^n if it exists.

2.) Adaptibility to change in n/w topology resulting from node or link failure.

3.) Adaptibility to varying source-destination traffic loads.
→ Traffic loads are quantities that are changing dynamically. An adaptive routing algorithm would be able to adjust the paths based on the current traffic loads.

4.) Adaptibility to route packets away from temporarily congested links.

5.) Ability to determine the connectivity of the n/w.
→ To find optimal paths, the routing system needs to know the connectivity or reachability information.

6.) Ability to avoid routing loops.

7.) low overhead..
   ↳ * inconsistent information in distributed computation may lead to routing tables that create routing loops
   ↳ * A routing system typically obtains the connectivity information by exchanging control messages with other routing systems. These messages represent an overhead on BW usage that should be minimised.

Routing Algorithm classification

1.) Based on their responsiveness
   a.) static routing
   b.) Dynamic (adaptive) routing

Static routing

* paths are precomputed based on the n/w topology, link capacities, and other information.
* Computation is typically performed offline by a dedicated host

* When the computation is completed, the paths are loaded to the routing tables and remain fixed for a relatively long period of time.

Adv: Static routing is best if
i.) network topology is fixed
ii.) n/w size is small
iii.) Traffic load does not change.

Disadv: i.) Becomes cumbersome if n/w size increases.
ii.) If traffic load changes, the precomputed paths may easily become sub optimal.
iii.) Inability to react rapidly for n/w failures.

Dynamic routing

* Each node continuously learns the state of the n/w by communication with its neighbours
* change in a n/w topology is propagated to all nodes.
* Based on the information collected, each node can compute the best paths to desired destination.

Disadv: Complexity in the node increases.

2.) Centralized routing / Distributed routing.

* In Centralized routing, A n/w control center computes all paths and then uploads this information to all the nodes in the n/w.
* In Distributed routing, nodes cooperate by means of messages exchanges and performs their own routing computations.

SHORTEST PATH ROUTING.

Bellman-Ford Algorithm.

Step 1: Initialization (destination node d is distance 0 from itself)

$D_i = \infty$ for $i \neq d$

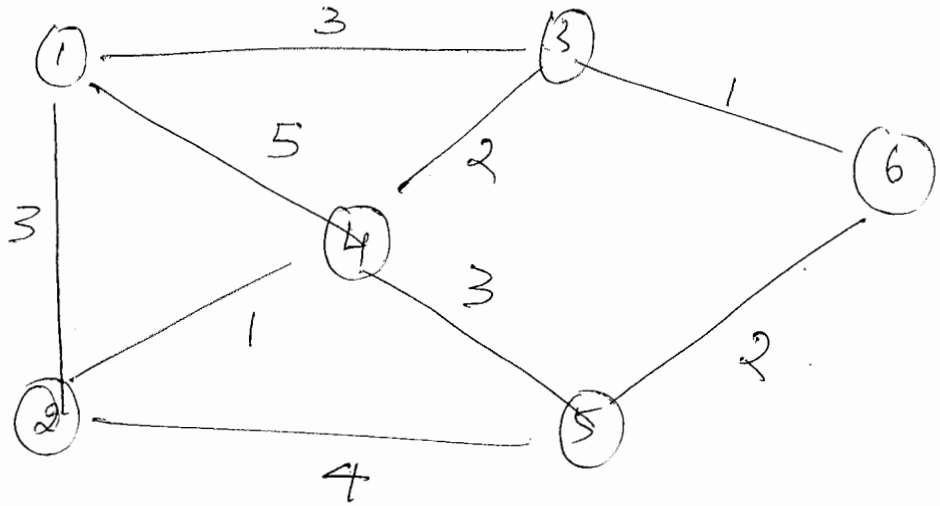
$D_d = 0$

Step 2: Updating (find minimum distance to destination through neighbours)

For each $i \neq d$,

$$D_i = \min_j \{C_{ij} + D_j\} \quad \text{for all } j \neq i$$

problem: Given fig below. Apply Bellman Ford Algorithm to find both the minimum cost from each node to the destination (node 6) and the next node along the shortest path.



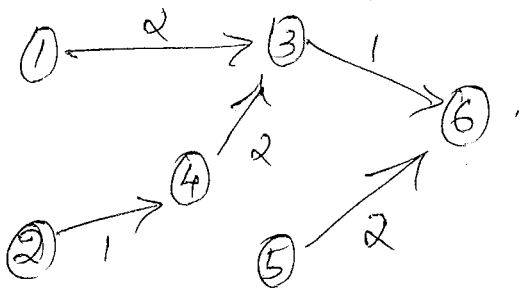
Each entry for node i represents the next node and the cost of current shortest path to dest^n 6.

iteration	node 1	node 2	node 3	node 4	node 5
initial	(1, ∞)	(-1, ∞)	(1, ∞)	(1, ∞)	(-1, ∞)
1	(-1, ∞)	(-1, ∞)	(6, 1)	(-1, ∞)	(6, 2)
2	(3, 3)	(5, 6)	(6, 1)	(3, 3)	(6, 2)
3	(3, 3)	(4, 4)	(6, 1)	(3, 3)	(6, 2)
4	(3, 3)	(4, 4)	(6, 1)	(3, 3)	(6, 2)

* initially, all nodes other than dest^n node 6 are at infinite cost (distance) to node 6. Node 6 informs the neighbours it is distance 0 from itself.
* (Iteration 1): node 3 finds that it is connected to node 6 with cost of 1. Node 5 finds that it is connected to node 6 with cost of 2. Node 2 & 5 updates their entries & inform their neighbo
* (Iteration 2): node 1 finds it can reach node 6 via node 3 with cost 3. Node 2 finds it can reach node 6 via node 5 with cost 6. Node 4 finds it has paths via nodes 3 & 5 with costs 3 & 7 respectively. Node 4 selects path via node 3. Nodes 1, 2, 4 update their entries & inform their neighbours.
* (Iteration 3): node 2 finds that it can reach node 6 via node 4 with distance 4. Node 2 changes its entry to (4, 4) and informs its neighbours
* (Iteration 4): Node 1, 4, and 5 process the new entry from node 2 but do not find any new shortest paths.

The algorithm has converged.

shortest path tree



what happens if link from node 3 to 6 break?

update	node1	node2	node3	node4	node5
Before break	(3,3)	(4,4)	(6,1)	(3,3)	(6,2)
1	(3,3)	(4,4)	(4,5)	(3,3)	(6,2)
2	(3,7)	(4,4)	(4,5)	(5,5)	(6,2)
3	(3,7)	(4,6)	(4,7)	(5,5)	(6,2)
4	(2,9)	(4,6)	(4,7)	(5,5)	(6,2)
5	(2,9)	(4,6)	(4,7)	(5,5)	(6,2)

note

link state Routing versus Distance-vector Routing

Distance vector routing	link state routing
* sends larger updates, about the complete n/w only to neighbouring routers	* send smaller updates, about the link state of neighbors to all routers.
* learns about n/w only from neighbours	* from all routers.
* Building routing table is based on inputs from only neighbours	* Based on complete database collected from all routers.
* Advertisement of updates is periodically (eg every 30s)	* Triggered updates, only when there is a change.
* convergence is slow	* fast
* Less CPU power & m/m is needed	* more
* cost : less	* more.
* scalability : less	* more.

ATM networks

* Asynchronous Transfer mode is a method for multiplexing & switching that supports a broad range of services.

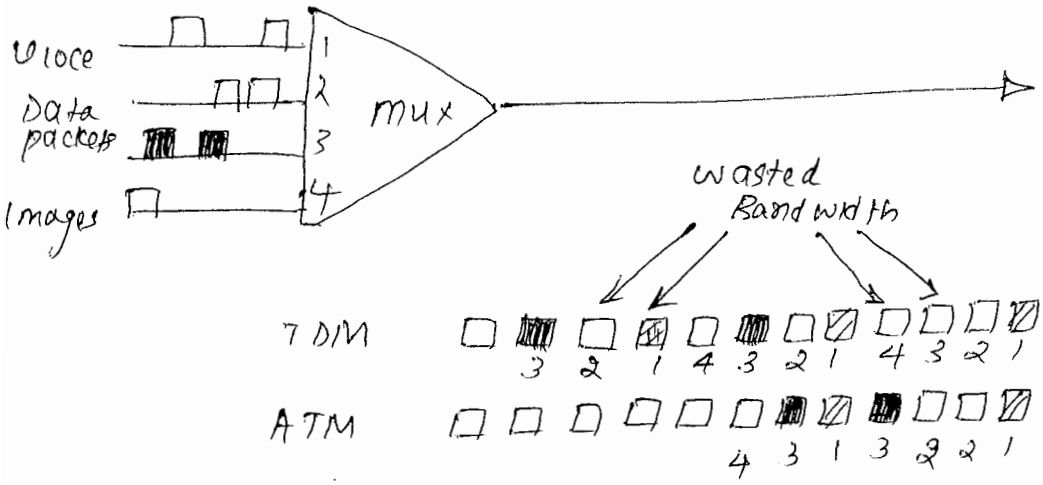


fig ATM multiplexing.

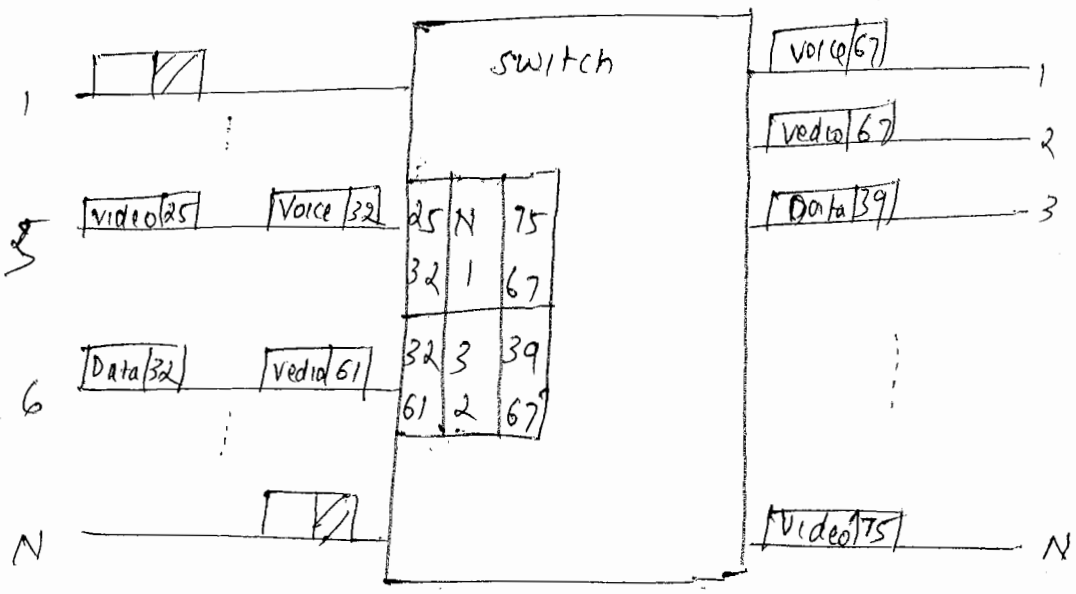


fig: ATM switching.

UNIT 2

PACKET - SWITCHING NETWORKS - 2.
TCP / IP - 1

Syllabus

* Traffic management at packet level
* Traffic management at flow level
* Traffic management at flow aggregate level
* TCP/IP architecture
* The internet protocol

– 6 Hours.

Ashok Kumar K
VIVEKANANDA INSTITUTE OF TECHNOLOGY

TRAFFIC MANAGEMENT AT THE PACKET LEVEL

* Traffic management is concerned with delivery of QoS to the end user and with efficient use of n/w resources.
* We can classify Traffic management into 3 levels
  - → packet level
  - → Flow level
  - → Flow aggregate level.
* Traffic management at packet level is concerned with packet queing and packet scheduling at switches, routers and multiplexers.

note:

* path traversed by a packet through a n/w can be modeled as sequence of queing system. as shown.

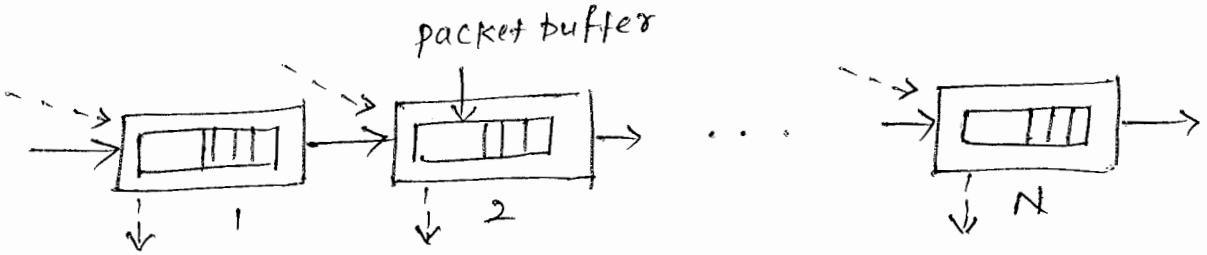


fig: end to end QoS of a packet.

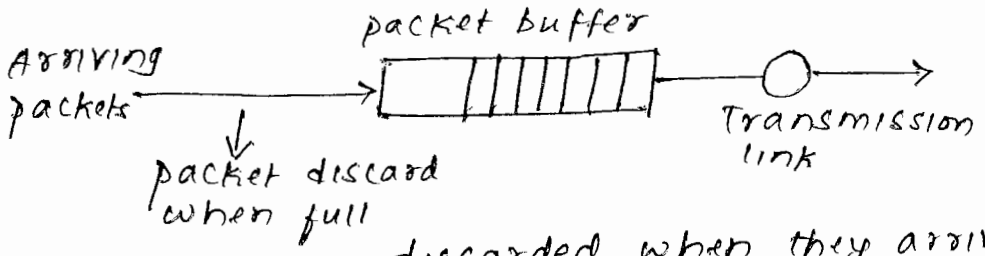
* performance experienced by a packet along the path is the accumulation of performance experienced at N queing systems. for eg: Total end-to-end delay is the sum of individual delays experienced at each system.
* Jitter measures the variability in packet delays
* packet loss occurs when packet arrives at queing system that has no more buffers available.
* queue scheduling → means implementing strategies for controlling the transmission bit rates
* Queue management → means managing how packets are placed in queing system.

FIFO and priority Queues

* Here we discuss three approaches to queue scheduling.

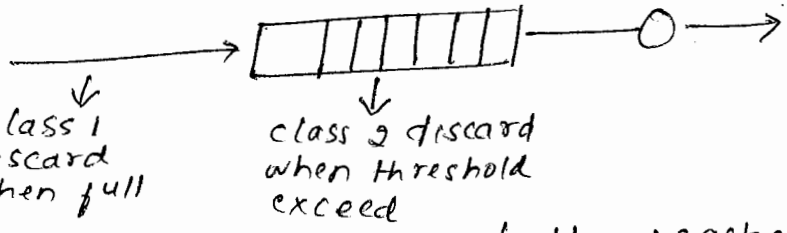
i.) First-in-First-Out (FIFO) queuing

* simplest approach, where packets are transmitted in the order of their arrival as shown.



* packets are discarded when they arrive at full buffer.
* Delay and loss experienced by packets depend on packet interarrival times & on the packet length.
* As interarrival time (or packet length) increases, the queue increases, and performance decreases.
* Since FIFO queing treats all packets in same manner, it is not possible to provide different QoS.
* This system is subject to hogging, which occurs when a user sends packets at high rate and fills the buffer, thus depriving other users of access to the buffer.

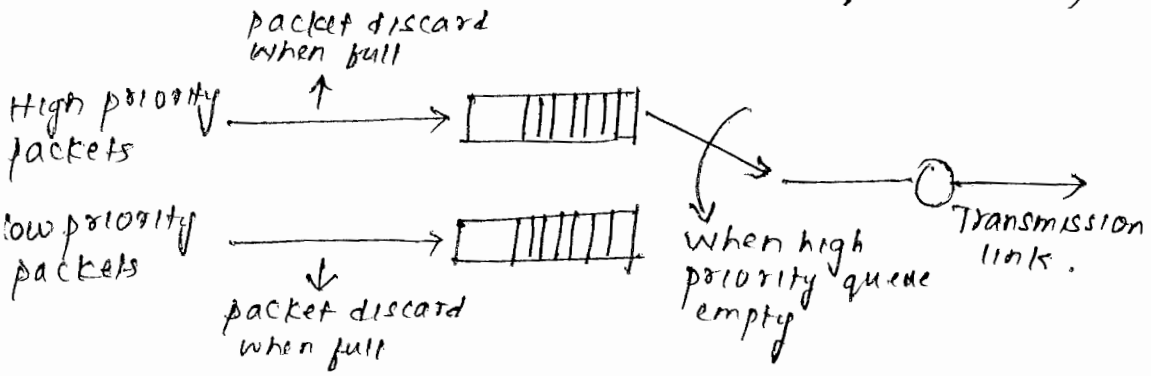
* Modified FIFO Queuing system ⇒ 2 classes of traffic.



* when no. of packets in a buffer reaches a certain threshold, arrivals of lower access priority (class 2) are not allowed into the system. Arrivals of higher access priority (class 1) are allowed as long as buffer is not full.

ii.) Head-of-line (HOL) priority queuing.

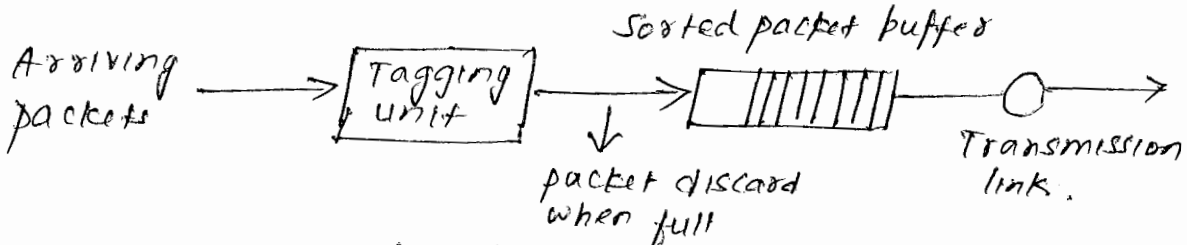
* Defines no. of priority classes.
* separate buffer is maintained for each priority class.



* Each time, the transmission link becomes available, the next packet for transmission is selected from the head of line of the highest priority queue that is not empty.

iii.) third approach.

* involves sorting packets in the buffer according to a priority tag that reflects the urgency with which each packet needs to be transmitted.



* priorities can be defined dynamically.
* priority tag consists of → priority class
  → arrival time.
* priority tag gives the due date
  if delay is less → earlier due date
  if delay is more → long due date.

Fair Queuing

* Fair queing attempts to provide equitable access to transmission bandwidth.
* In the ideal fluid flow situation, the transmission bandwidth is divided equally among all non empty buffers.

* Thus, if the total number of flows in the system is n (ie no. of non empty buffers is n), and the transmission capacity is C bits/second. Then each flow is guaranteed atleast C/n bits/second

* In practice, dividing the capacity exactly equally is not possible.

* one approach could be to service each non empty buffer one bit at a time in round-robin fashion as shown in above fig.

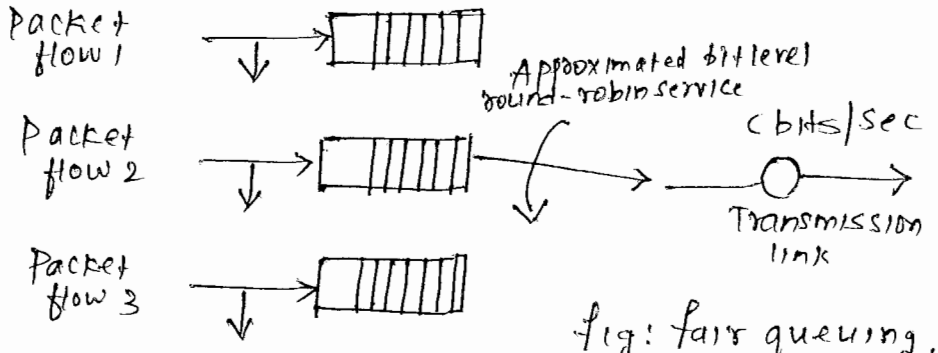
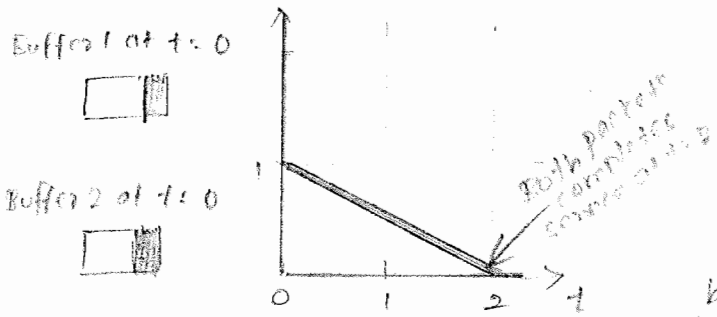


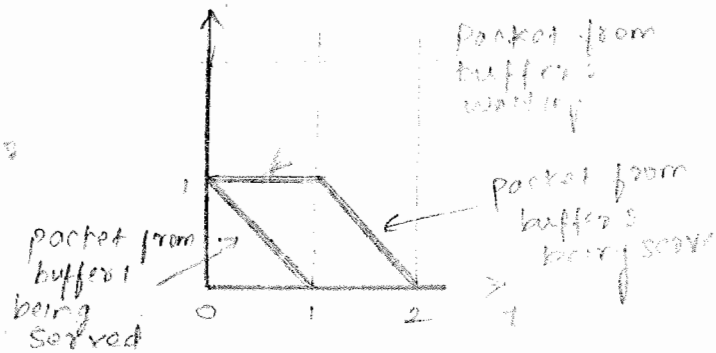
fig: fair queuing.

* Fair queing is used in ATM, since all packets are of same length here.

* Fair queuing can be either fluid-flow (ideal) or packet-by-packet. Diff b/w these two is illustrated below.



Fluid flow system: both packet served at rate 1/2



packet by packet system: Buffer 1 served first at rate 1, then buffer 2 served at rate 1

* Assumptions:
  - → Buffer 1 and buffer 2 each has single L-bit packet to transmi at $t=0$ & no subsequent packets arrive.
  - → Capacity is $C = L$ bits/second $=$ 1 packet/second.

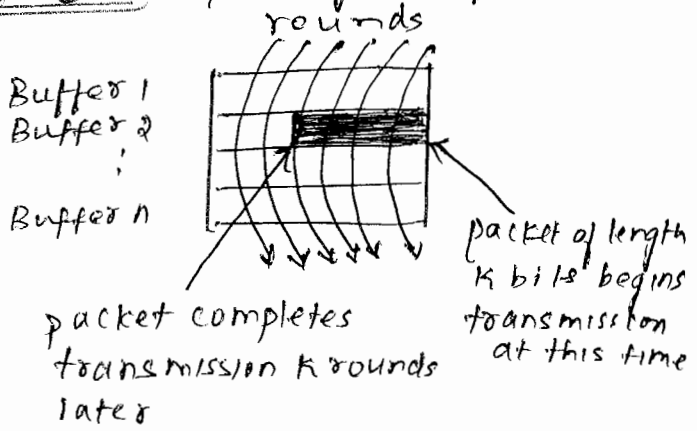
* Fluid flow system transmit each packet at a rate of 1/2 and therefore completes the transmission of both packets exactly at time $t=2$ seconds.

* packet-by-packet fair queuing system transmits the packet from buffer 1 first & then the packet from buffer 2, so packet completion times are 1 and 2 seconds respectively.

* if the packets have variable length, then servicing one packet at a time (from different user buffers) in round robin fashion will not necessarily obtain a fair allocation of transmission bandwidth.

Solution: Each time a packet arrives at a user buffer, the completion time of a packet is derived from a fluid-flow fair queuing system. This number is used as a finish tag for the packet.
Each time the transmission of a packet is completed, the next packet to be transmitted is the one with the smallest finish tag among all of the user buffers.
We refer to this system as packet-by-packet fair queuing system.

note: computing the finish tag.



Finish tag of the packet $F(i,k) = \max\{F(i,k-1), R(t_k^i)\} + P(i,k)$.

where $F(i,k-1)$ → Finish tag of prev. packet in its queue,
$R(t_k^i)$ → no. of rounds at time t for k^{th} packet from flow i.
$P(i,k)$ → length of the packet.

Example:

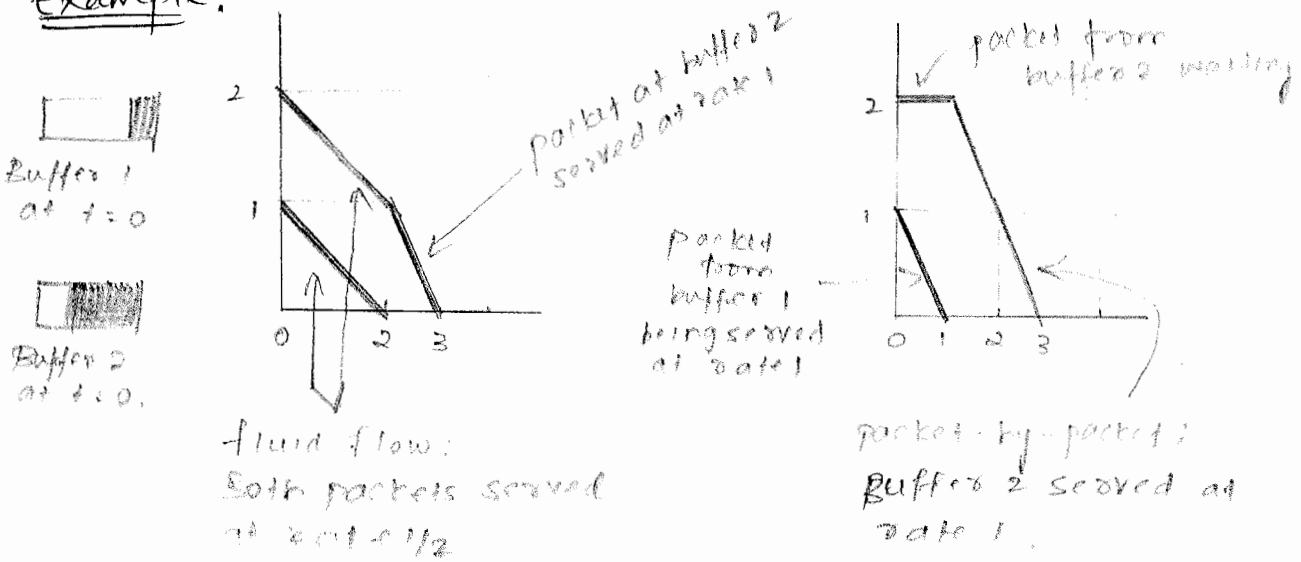


fig: Fluid flow & packet by packet fair queuing for two packets of different lengths.

* A fluid flow system services each buffer at rate 1/2 as long as both buffers remain non empty.
* In packet by packet system,
  finish tag of packet of buffer 1 $\} F(1,1) = R(0) + 1 = 1$.
  finish tag of packet from buffer 2 $\} F(2,1) = R(0) + 2 = 2$.
  ∴ system will service buffer 1 first.

Weighted Fair Queuing.

* weighted fair queuing addresses the situation in which different users have different requirements.
* Here, each user flow has its own buffer (as before), but it also has a **weight** that determines its relative share of the bandwidth.
* eg: If buffer 1 has weight 1 & buffer 2 has weight 3, then buffer 1 receives $1/(1+3) = 1/4$ of bandwidth & buffer 2 receives 3/4 of the bandwidth.
* Below fig shows weighted fair queuing for fluid flow and packet by packet system.

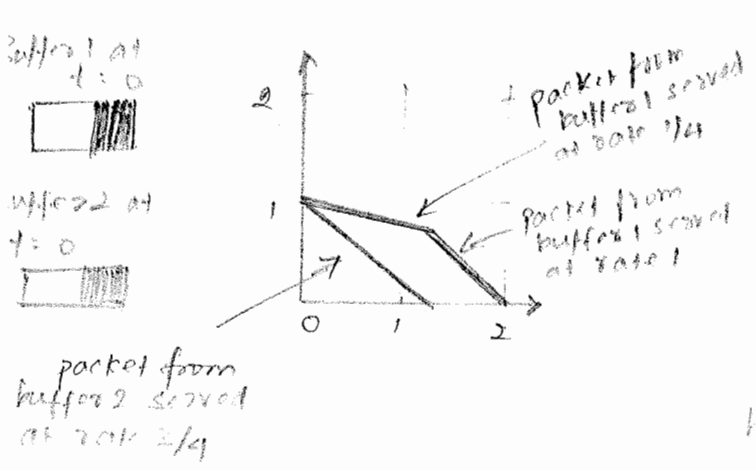
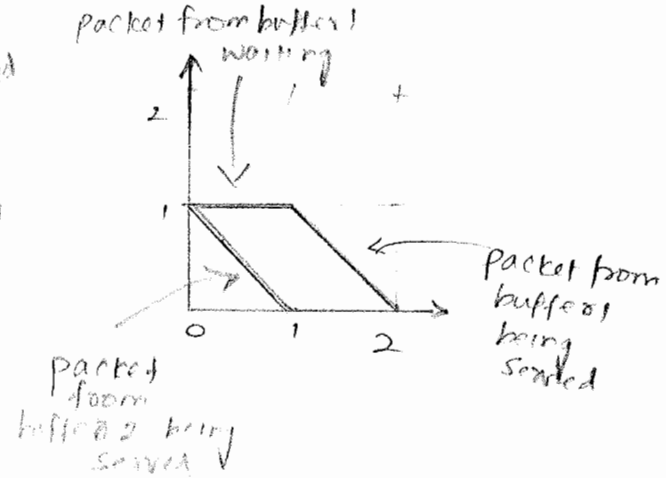


fig (a): fluid flow system



Here Buffer2 served first at rate 1, then buffer1 served at rate 1

fig (b): packet by packet system.

* weighted fair queuing is also easily approximated in ATM.

note:

* Suppose that there are n packet flows and that flow i has weight w_i, then, Packet-by-packet system calculates its finish tag as follows.

$$F(i,k) = \max\{ F(i,k-1), R(t_k^i) \} + P(i,k)/w_i$$

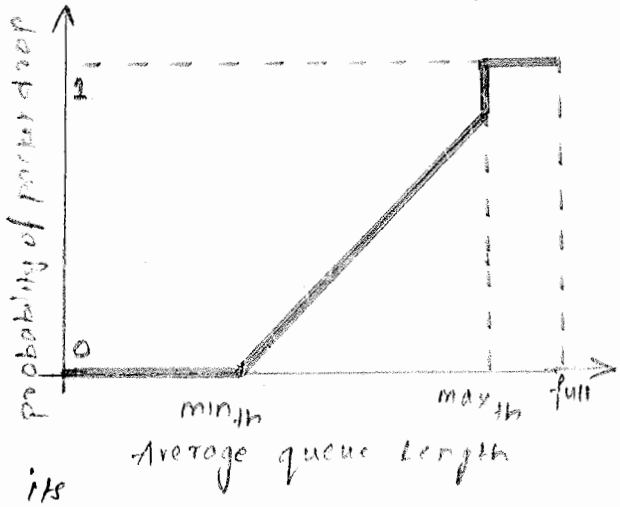
Random Early Detection (RED)

* RED is a buffer management technique that attempts to provide equitable access to a FIFO system by randomly dropping arriving packets before the buffer overflows.
  (A dropped packet provides feedback info to the source & inform the source to reduce its transmission rate)
* RED algorithm uses average queue length (not instantaneous queue length) to decide how to drop packets.
* Two thresholds are defined here.
  - → min_{th}
  - → max_{th}

→ When average queue length is below min_{th}, RED does not drop any arriving packets.

→ When average queue length is blw min_{th} & max_{th}, RED drops an arriving packet with an increasing probability as average queue length increases.

This method of "early" drop is used to notify the source to reduce its transmission rate before the buffer becomes full.



→ When average queue length exceeds max_{th}, RED drops any arriving packet (refer fig)

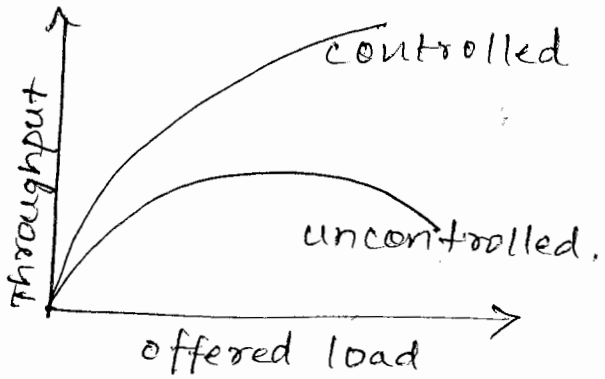
TRAFFIC MANAGEMENT AT THE FLOW LEVEL

* At the flow level, traffic management is concerned with managing the individual traffic flow to ensure that the QoS (eg: delay, jitter, loss) requested by the user is satisfied.

note:

* congestion arises when incoming rate exceeds outgoing rate.
* process of managing the traffic flow in order to control congestion is called congestion control.

In fig
controlled curve → throughput with congestion control
uncontrolled curve → throughput without congestion control



* purpose of traffic mgmt at the flow level is to control the flows of traffic and maintain performance (controlled curve) even in presence of traffic overload

* Two classification of congestion control algorithm are
  → open loop control
  → closed loop control

Open-Loop Control

* prevents congestion by making sure that the traffic flow generated by the source will not degrade the performance of the n/w to a level below the specified QoS.
  If QoS cannot be guaranteed, the n/w rejects the traffic flow before it enters the n/w.
* open loop control does not rely on feedback information to react to congestion. Instead, it is based on the principle that n/w performance is guaranteed to all traffic flows that have been admitted into the n/w.
* To guarantee n/w performance, open loop control relies on three mechanisms.
  → Admission control
  → Policing
  → Traffic shaping.

Admission control → (or network function)

* The function that makes the decision to accept or reject a new traffic flow into an n/w is called admission control. It makes this decision by computing the resources ~~bar~~ (typically bandwidth & buffers) requirements of a new flow.
* Thus, a source initiating a new flow must first obtain permission from an admission control entity.
  → If QoS of new flow can be satisfied without violating QoS of existing flows, the flow is accepted.
  → otherwise, the flow is rejected.

~~* To det~~ ~~mote:~~

* To determine whether the QoS of the flow can be satisfied, the admission control entity has to know the traffic parameters & QoS requirements of the flow.

* Traffic parameters typically include ;

→ peak rate (bits/second or bytes/second) : max rate that the source will generate its packets.

→ Average rate (bits or bytes/second) : ave rate that the source will generate its packets.

→ Maximum burst size (bits, bytes, or seconds) : max length of time the traffic can be generated at the peak rate.

* Based on traffic parameters and QOS requirement, the admission control entity calculates how much bandwidth it has to reserve for the ~~the~~ new flow.

* Amount of bandwidth generally lies b/w average rate & peak rate and is called the effective bandwidth of the flow.

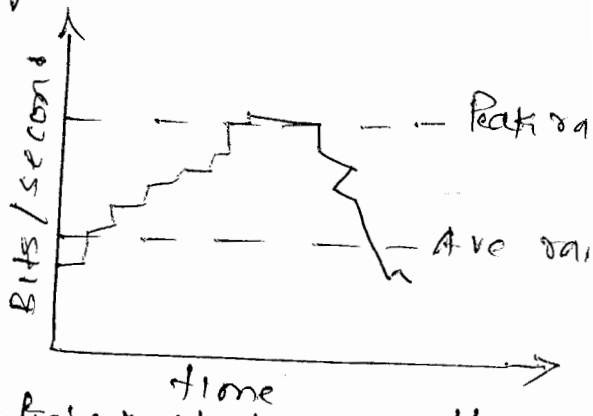


fig: ex of traffic flow

policing

* Once a flow is accepted by admission control entity, the QOS will be satisfied as long as the source obeys its negotiated traffic parameters during the lifetime of the flow.

* To prevent the source from violating its contract, the n/w may want to monitor the traffic flow continuously. The process of monitoring and enforcing the ~~$~~ traffic flow is called policing

* Most implementations of a policing device are based on the concept of leaky bucket.

* Below fig shows leaky bucket algorithm used for policing.

Explanation:

* At the arrival of first packet, the content of the bucket, X is set to zero and the LCT is set to the arrival time of first packet.

* At the arrival of k^{th} packet, the auxiliary variable x' records the difference b/w the bucket content at the arrival of last conforming packet & the interarrival time b/w the last conforming packet & the k^{th} packet.
* Auxiliary variable is non negative.
* If it is greater than L, the packet is considered nonconforming otherwise, conforming.
* Bucket content & the arrival time of the packet are then updated.

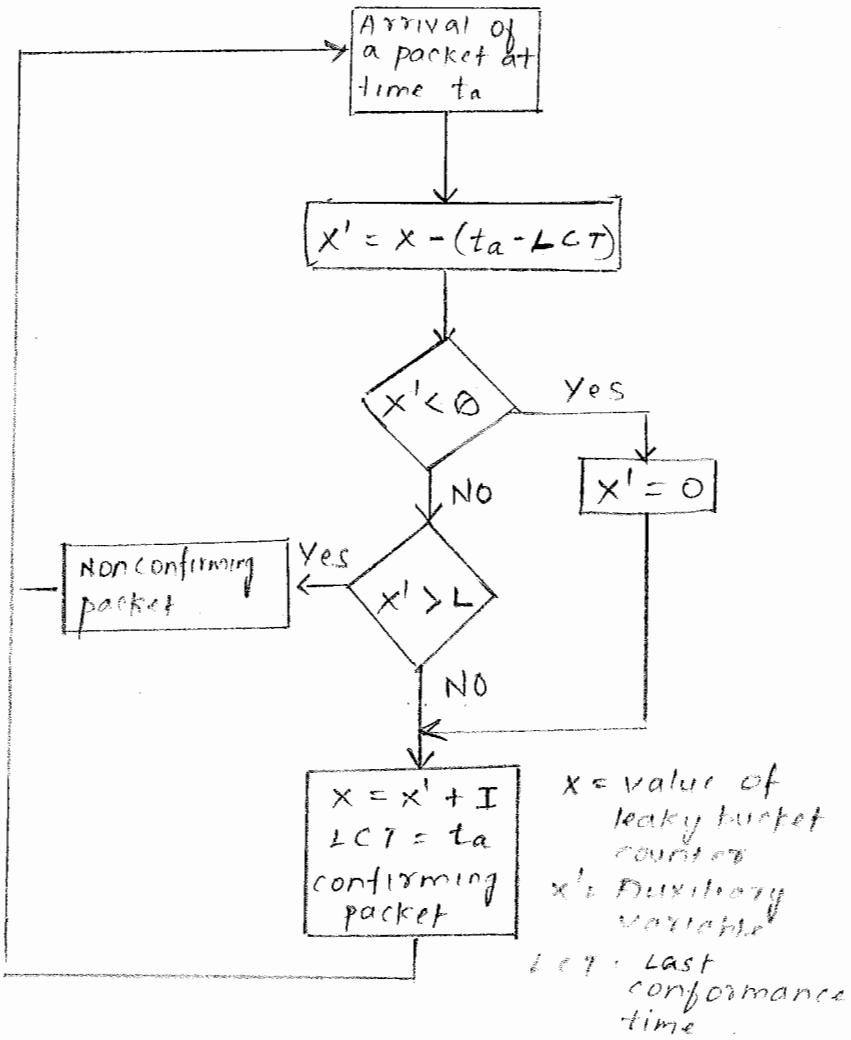
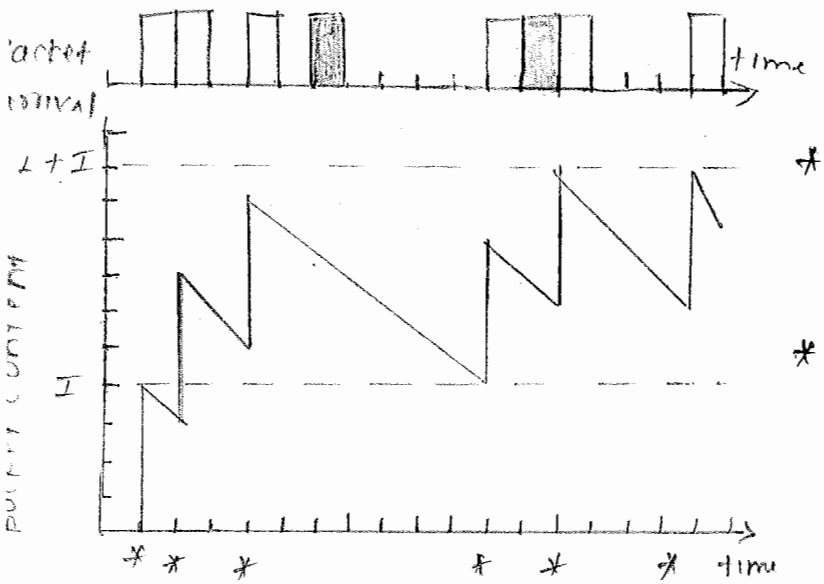


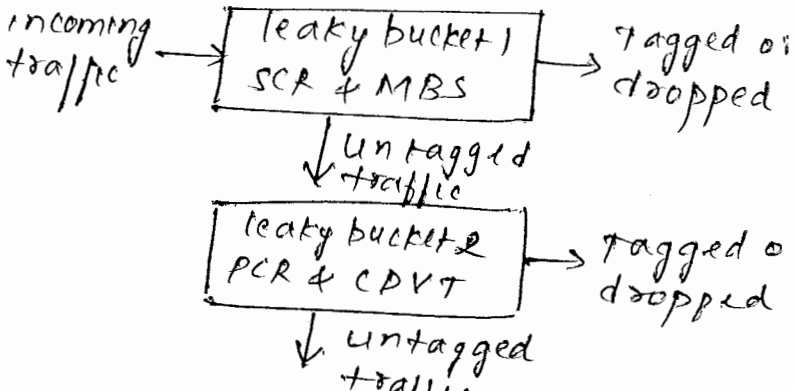
Illustration of operation of leaky bucket algorithm.



* Here value of I is four packet times, & the value of L is six packet times.
* Arrival of first packet increases the bucket content by four (packet times)
* At second arrival, the content had decreased to three, but four more are added resulting in total of seven.
* The fourth packet is declared as nonconforming since it would increase the content to 11, which would exceed L + I (10).
* packet 1, 2, 3, 5, 7, 8 are conforming, and the packets 4, 6 are nonconforming

note:

* maximum no. of packets that can be transmitted at the peak rate is called maximum burst size (MBS)
* inverse of I is often called the sustainable rate, which is the long-term average rate allowed for the conforming traffic.
* A combination of leaky buckets can be used to police multiple traffic parameters (eg PCR & SCR).
* In this situation, dual leaky buckets such as the one shown can be used.
* conforming → untagged
  nonconforming → tagged.



SCR = sustainable cell rate.
PCR = peak cell rate
CDVT = cell ~~tolerance~~ delay variation tolerance.

Traffic Shaping

* If the source wants to ensure that the traffic flow conforms to the parameters specified in leaky bucket policing device, it should first alter the traffic flow.
  Traffic shaping refers to the process of altering a traffic flow to ensure conformance.
* Typically, Traffic shaping device is located at the node just before the traffic flow leaves a n/w (egress node) while the policing device is located at the node that receives the the traffic flow from another n/w (ingress node)

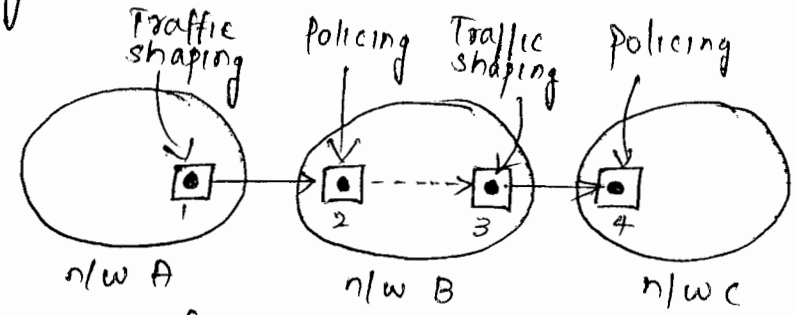
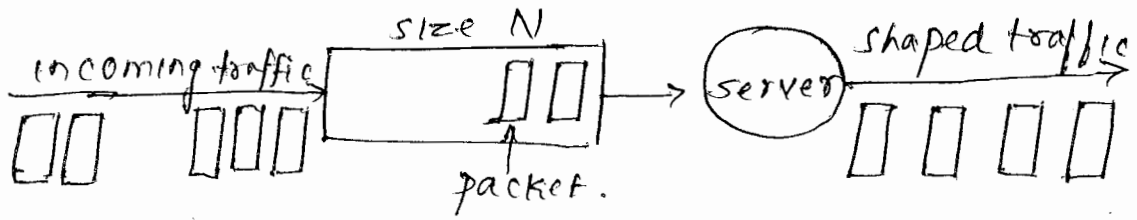


fig: Typical locations of policing & traffic shaping devices.

* Realising traffic shaping in 2 ways
  → Leaky bucket traffic shaper. (simple)
  → Token bucket traffic shaper. (more realistic)
    (simple extension of Leaky bucket)

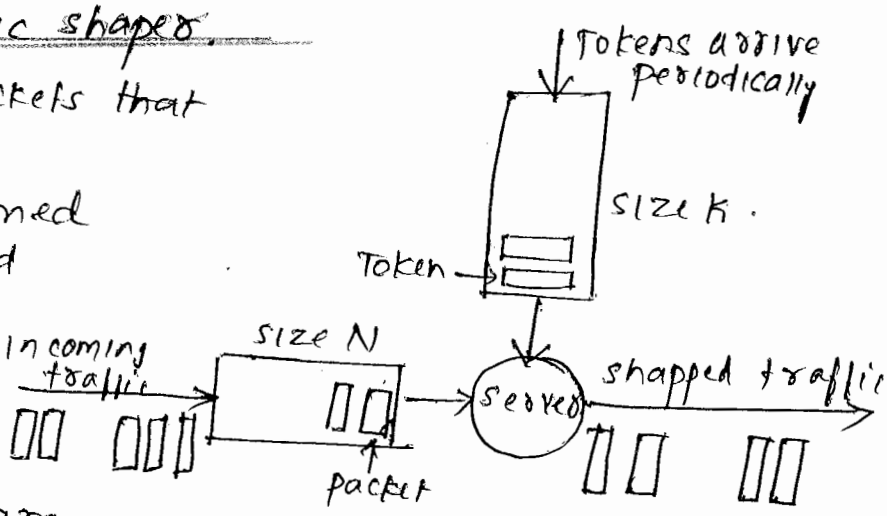
Leaky bucket traffic shaper.



* Incoming packets are first stored in a buffer
* packets (assumed to be of fixed length) are served periodically so that the stream of packets at the o/p is smooth
* Buffer is used to store momentary bursts of packets. The buffer size defines the maximum bursts that can be accomodated, and incoming packets are discarded when the buffer is full.
* ~~A traffic shaping device needs to introduce~~.
* A policing device checks & passes each packet on the fly. A traffic shaping device needs to introduce certain delays for packets that arrive earlier than their scheduled departures & requires a buffer to store these packets.

Disadv: o/p rate is constant when buffer is not empty. Many applications produces variable rate traffic.

Token bucket Traffic shaper.



* Regulates only the packets that are not conforming
* Packets that are deemed conforming are passed through without further delay.
* Tokens are generated periodically at a constant rate and are stored in token bucket.
* If token bucket is full, arriving tokens are discarded
* A packet from the buffer can be taken out only if a token in the token bucket can be drawn.

* If the token bucket is empty, arriving packets have to wait ~~the~~ in the packet buffer.
Thus, we can think of a token as a permit to send a packe

note: In the case where the buffer has a backlogs of packets when the token bucket is empty, the behaviour of token bucket shaper is very similar to that of the leaky bucket shaper.

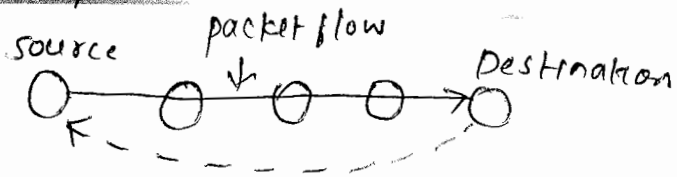
closed-Loop control

* 2nd algorithm for congestion control
* closed loop control mechanism relies ~~relies~~ on feedback information to regulate a packet flow rate according to feedback information about the state of the n/w, which may be based on buffer content, link utilization, or other relevent congestion information.

note: In TCP/IP environment, control is implemented at the transport layer.
In ATM environment, control is implemented at ATM layer corresponding to n/w layer.

End-to-end versus hop-by-hop

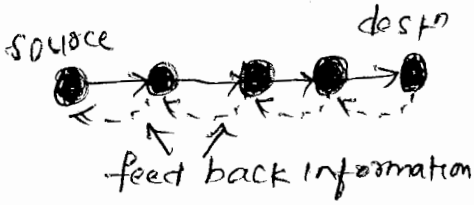
* With end-to-end closed loop control, the feedback information about the state of the n/w is propagated back to the source that can regulate the packet flow rate.



(a) end-to-end.

The feedback information may be forwarded directly by a node that detects congestion, or it may be forwarded to the dest" first which then relays the information to the source as shown

Disadv:
→ Transmission of feedback information introduces certain propogation delay.
→ ∴ information may not be accurate when source receives such information.



(b) Hop-by-hop.

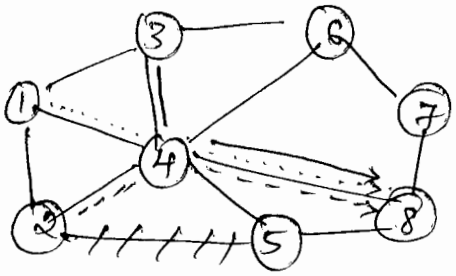
* Hop-by-hop control typically can react much faster, due to shorter propagation delay.
* State of the n/w is propogated to the upstream node as shown.
* When a node detects congestion on its outgoing link, it can tell its upstream neighbour to slow down its transmission rate.
  As a result, the upstream neighbour may also experience congestion some time later if the incoming rate exceeds the outgoing rate.
* This "back-pressure" process from one downstream node to another node upstream may continue all the way to the source.

Implicit versus explicit feedback.

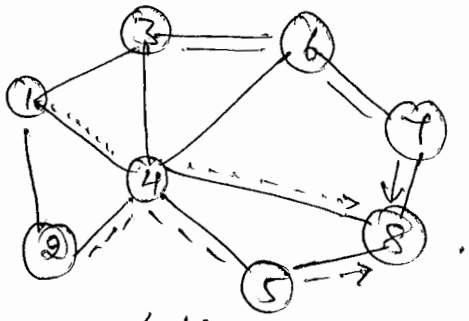
* Feedback information can be implicit or explicit.
* With explicit feedback, the node detecting congestion initiates an explicit message that eventually arrives at the source notifying congestion in the network.
  The explicit message can be transmitted as a separate packet (often called the choke packet) or piggybacked on a data packet.
  eg: closed loop control in ATM n/w. - Here explicit feedback info. is record in a bit called EFCI bit.
* With implicit feedback, no such explicit message is forwarded. Instead, the source has to rely on some surrogate information to deduce congestion.
  one ex. is to use a time-out based on missing acknowledgements from a destn to decide whether congestion has been encountered in the n/w.
  eg: TCP congestion control - here it is derived from missing acknowledgements.

TRAFFIC MANAGEMENT AT THE FLOW AGGREGATE LEVEL

* Deals with multiplicity of flows.
* often called Traffic engineering.
* objective : map aggregated flows onto the n/w so that the resources are efficiently utilized.
* mapping the traffic according to shortest paths may not result in overall n/w efficiency <refer fig (a)>
* Fig (b) shows an ex of better traffic mapping where the traffic is distributed across the n/w.



(a)



(b)

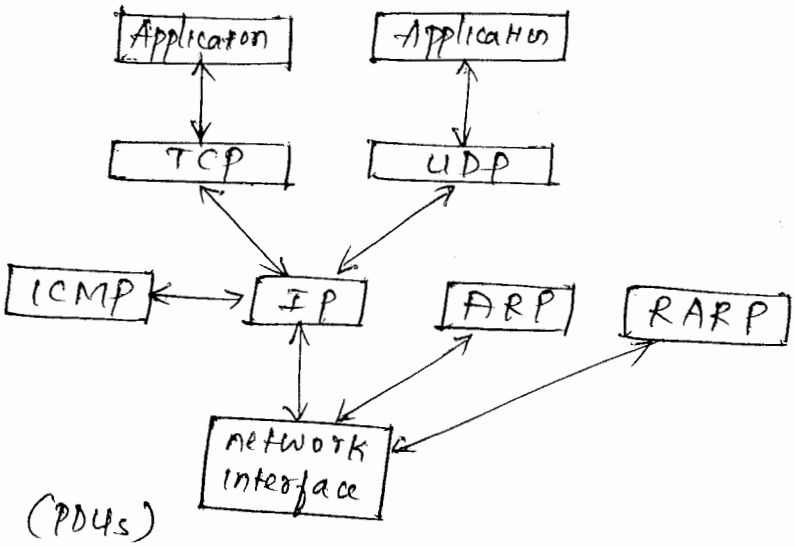
* simple traffic engineering technique is ~~called~~ constrained shortest path routing, which is suitable for connection-oriented packet switching n/ws.

TCP/IP

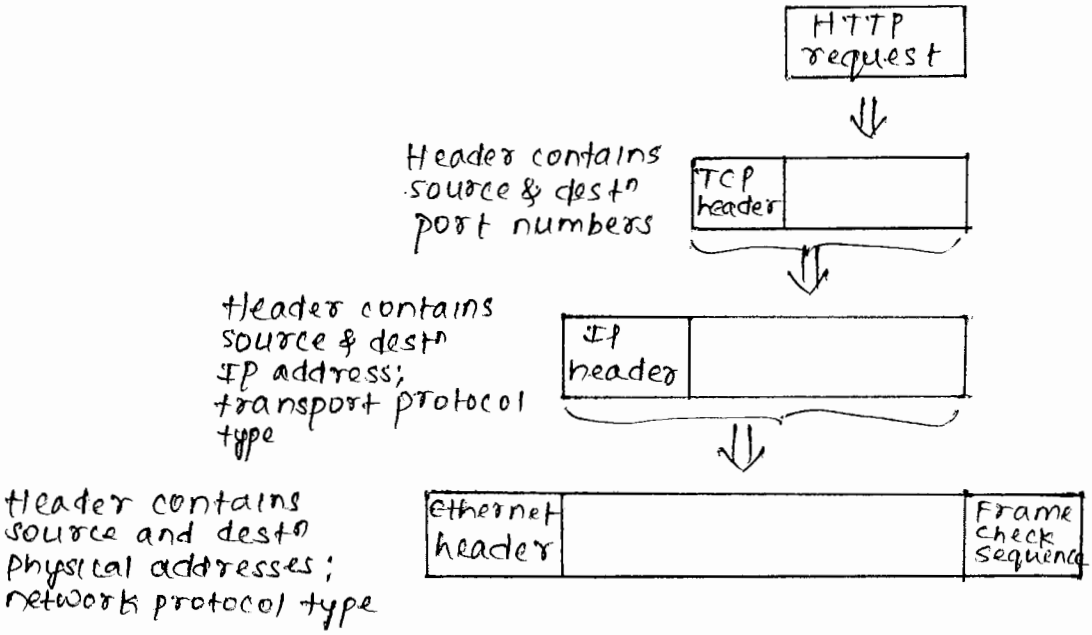
THE TCP/IP ARCHITECTURE

* Basic structure of the TCP/IP protocol suite is as shown.

* Application layer protocols such as FTP and HTTP send messages using TCP.
  Application layer protocols such as SNMP and DNS send their messages using UDP.

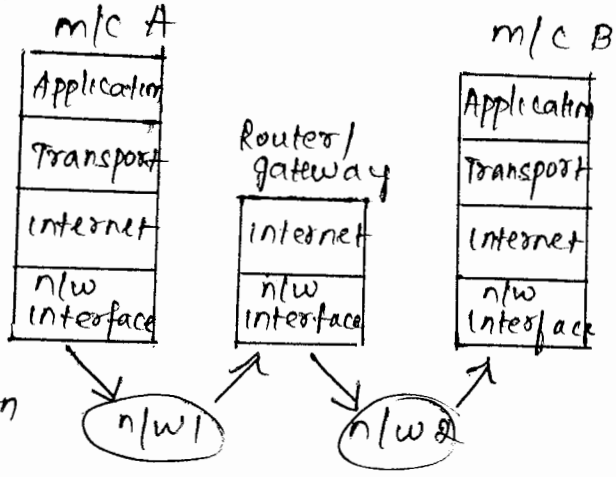


* Protocol Data Units (PDUs) exchanged by the peer TCP protocols are called TCP segments or segments, while those exchanged by UDP protocols are called UDP datagrams or datagrams.
* IP multiplexes TCP segments and UDP datagrams and performs fragmentation, if necessary.
* PDUs exchanged by IP protocols are called IP packets or packets.
* IP packets are sent to n/w interface for delivery across the physical network.
* At the receiver, packets passed up by the n/w interface are demultiplexed to the appropriate protocol (IP, ARP, or RARP) Receiving IP entity determines whether a packet should be sent to TCP or UDP.
  Finally TCP (or UDP) sends each segment (or datagrams) to the appropriate application based on port number.
* The PDU of a given layer is encapsulated in a PDU of the layer below as shown:



* For ex; HTTP GET command is passed to the TCP layer which encapsulates the message into a TCP segment. TCP segment in turn is passed to the IP layer where it is encapsulated in an IP packet. IP packet is then passed through the n/w interface and encapsulated into a PDU of the underlying n/w. In above fig; the IP packet is encapsulated into an ~~PDU of the~~ ethernet frame.

note: logical IP addresses need to be converted into specific physical address to carry out the transfer of bits from one device to the other. this conversion is done by Address Resolution protocol.



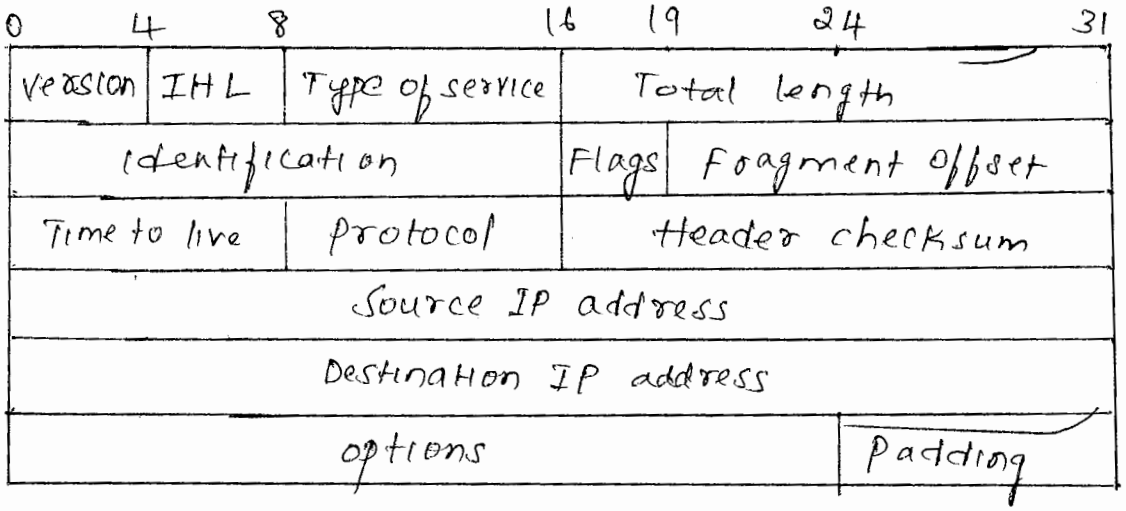
* Each host in the internet is identified by a globally unique IP address, which divided into two parts.
  - network ID & - host ID.
* Internet layer provides for the transfer of information across multiple networks through the use of routers as shown in above figure.
* IP packets are exchanged b/w routers without a connection setup; they are routed independently & may

* The routers that interconnect the intermediate n/ws may discard packets when they encounter congestion.
* The responsibility for recovery from these losses is passed on to transport layer.
* The n/w interface layer is particularly concerned with the protocols that are used to access the intermediate n/ws At each router, this layer is used to encapsulate the IP packet into a packet or frame of the underlying n/w or link

THE INTERNET PROTOCOL

* IP is the heart of TCP/IP protocol suite. IP corresponds to the n/w layer in OSI model and provides a connectionless best-effort delivery service to the transport layer. ↳⇒ IP will try its best to forward packets to destn & does not guarantee that packet will be delivered to destn,

IP packet (IPv4 header format)



1.) Version: indicates version number used by the IP packet so that revisions can be distinguished from each other. current IP version is 4.
2.) Internet Header length: specifies length of the header in 32 bit words. If no options are present, IHL will have a value of 5 length of option field can be determined from IHL

3.) Type of service (TOS): Traditionally specifies priority of the packet based on delay, throughput, reliability, and cost requirements.
Three bits are assigned for priority levels (called "precedence") and four bits for specific requirements (delay, throughput, reliability, cost)

4.) Total length: specifies no. of bytes of the IP packet including header & data. 16 bits assigned. Hence max. packet length is 2^{16} (65535) bytes.

5.) Identification, Flags, and Fragment offset: used for fragmentation & reassembly.

6.) Time to live (TTL): indicate amount of time in seconds the packet is allowed to remain in the n/w. However, most routers uses this field to indicate the no. of hops the packet is allowed to traverse in the n/w.

7.) protocol: specifies upper layer protocol that is to receive the IP data at the destination host. eg of protocols include TCP (protocol=6), UDP (protocol=17), & ICMP (protocol=1)

8.) Header checksum: verifies the integrity of the header of the IP packet. Data part is not verified here & is left to upper layer protocols. If verification process fails, the packet is simply discarded.

9.) Source and destination IP address: contains the addresses of source & destn hosts.

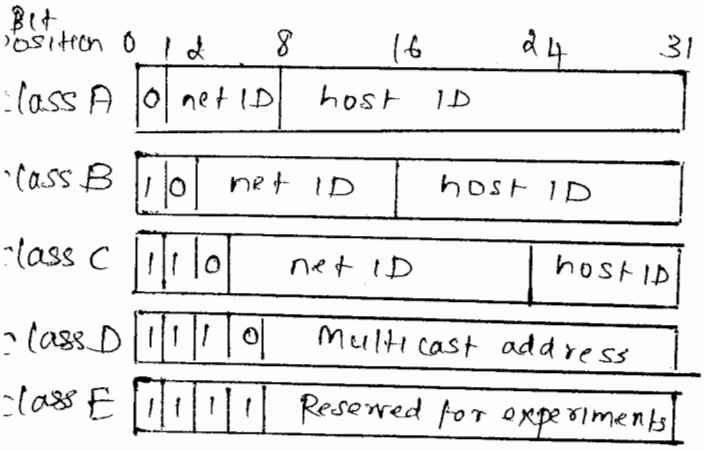
10.) options: variable length, allows the packet to request special features such as security level, route to be taken by the packet, & timestamp at each router.

11.) padding: used to make the header a multiple of 32-bit words.

IP addressing

- An IP address has a fixed length of 32 bits.
The address structure was originally defined to have a two-level hierarchy
  - → network ID: identifies the n/w the host is connected to

* Host ID is assigned by n/w administrator at the local site.
* network ID for an organisation may be assigned by the ISP. ISP inturn may request the n/w ID from its regional internet registry: American registry for Internet numbers (ARIN)
* The IP address structure is divided into five address classes, Class A, B, C, D, E identified by MSBs of the address as shown



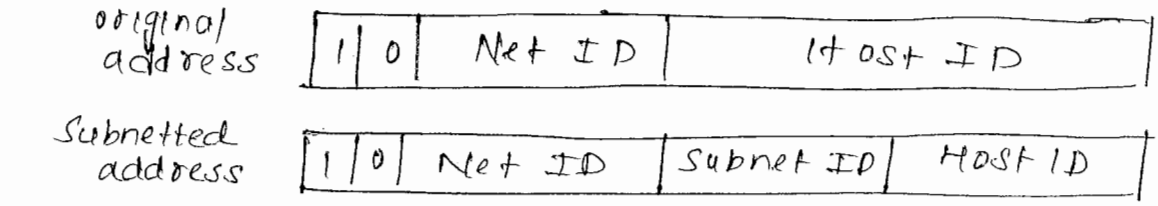
* class A addr have 7 bits for net IDs & 24 bits for host IDs allowing upto 126 n/ws & about 16 million hosts per n/w.
* class D addr. are used for multicast services that allow a host to send info. to a group of hosts simultaneously
* class E addr. are reserved for experiments.

* IP addr. are usually written in dotted-decimal notation.
  eg: IP addr of
  10000000 10000111 01000100 00000101
  is written as
  128.135.68.5 in dotted-decimal notation.
* A range of addresses has been defined for each IP class
  range 1: 10.0.0.0 to 10.255.255.255
  range 2: 172.16.0.0 to 172.31.255.255
  range 3: 192.168.0.0 to 192.168.255.255. (used in home LANs)
* NAT is used to connect to the global Internet

note:
* An ID that contains all 1s ⇒ the packet is broadcasted on the local n/w.
* An ID that contains all 0s ⇒ refers to the n/w specified by the n/w ID, rather to a host.

Subnet Addressing.

* Basic idea of subnetting is to add another hierarchial level called the "subnet" as shown



* To find the subnet number, the router ~~has~~ needs to store an additional quantity called a subnet mask, which consists of binary 1s for every bit position of the address except in the host ID field where binary 0s are used.

eg: IP addr: 150.100.12.176,
for this, the subnet mask is
11111111 11111111 11111111 10000000 which corresponds to 255.255.255.128 in dotted-decimal notation.

* The router can determine the subnet number by performing a binary AND b/w subnet mask and the IP addr.

eg: IP addr: 10010110 01100100 00001100 10110000
subnet mask: 11111111 11111111 11111111 10000000
subnet number: 10010110 01100100 00001100 10000000
≡ 150.100.12.128

note: IP address 150.100.12.128 is used to identify the subnetwork and, ~~the~~
IP address 150.100.12.255 is used to broadcast packets inside the subnetwork.
Thus the host connected to this subnetwork must have an IP addr in the range 150.100.12.129 to 150.100.12.254

problems.

1. Identify the address class of the following IP addresses.

a.) 200.58.20.165
b) 128.167.23.20.
c.) 16.196.128.50.
d) 150.156.10.10
e.) 250.10.24.96.

Also convert into their binary representation
↳ <left as assignment>

Soln:

a.)
2 | 200
2 | 100 - 0
2 | 50 - 0
2 | 25 - 0
2 | 12 - 1
2 | 6 - 0
2 | 3 - 0
1 - 1

11001000.
↳ class C

b.) 128
= 10000000
↳ class B

c.) 16
= 00010000
↳ class A

d.) 150
= 10010110
↳ class B

e.) 250
= 11111010
↳ class E

2. An Host in an organisation has an IP addr. 150.32.64.34 and an subnet mask 255.255.240.0 what is the address of this subnet? What is the range of IP addresses that a host can have on this subnet.

Soln ~~IP addres: 150.32.64.34.~~
~~Subnet mask 255.255.240.0~~

Subnet

IP Address : 10010110 00100000 01000000 00100010
subnet mask : 11111111 11111111 11110000 00000000
Subnet : 10010110 00100000 01000000 00000000

Host

from : 10010110 00100000 01000000 00000001
To : 10010110 00100000 01001111 11111110

Classless Interdomain Routing (CIDR)

* CIDR is a mechanism introduced to slow the growth of routing tables on routers ~~across~~ across the internet and to help prevent wastage of IP addresses by allocating a subset of a class A, B, or C network to ISPs and organisations.
* Using a CIDR notation, a prefix 205.100.0.0 of length 22 is written as 205.100.0.0/22. The /22 notation indicates that the network mask is 22 bits, or 255.255.252.
* entries in CIDR routing table contain a 32 bit IP address and a 32 bit mask. CIDR enables a technique called supernetting to allow a single routing entry to cover a block of classful addresses
for ex: instead of having four entries for contiguous set of class C addresses (eg 205.100.0.0, 205.100.1.0, 205.100.2.0, and 205.100.3.0), CIDR allows a single routing entry 205.100.0.0/22 which includes all IP addresses from 205.100.0.0 to 205.100.3.255.
* To see the route aggregation process in detail, we note that the original four class C entries,

class C address 205.100.0.0 = 11001101 01100100 00000000 00000000
class C address 205.100.1.0 = 11001101 01100100 00000001 00000000
class C address 205.100.2.0 = 11001101 01100100 00000010 00000000
class C address 205.100.3.0 = 11001101 01100100 00000011 00000000
become

mask 255.255.252.0 = 11111111 11111111 11111100 00000000
supernet address 205.100.0.0 = 11001101 01100100 00000000 00000000

problem.

1. perform CIDR aggregation on following /24 IP addresses: 128.56.24.0/24, 128.56.25.0/24, 128.56.26.0/24, 128.56.27.0/24.

Soln:

128.56.24.0/24 = 10000000 00111000 00011000 00000000
128.56.25.0/22 = 10000000 00111000 00011001 00000000
128.56.26.0/22 = 10000000 00111000 00011010 00000000
128.56.27.0/22 = 10000000 00111000 00011011 00000000
mask = 11111111 11111111 11111100 00000000
resulting prefix is = 128.56.24.0/22

2. perform CIDR aggregation on following /24 IP addresses 200.96.86.0/24, 200.96.87.0/24, 200.96.88.0/24, 200.96.89.0/24.

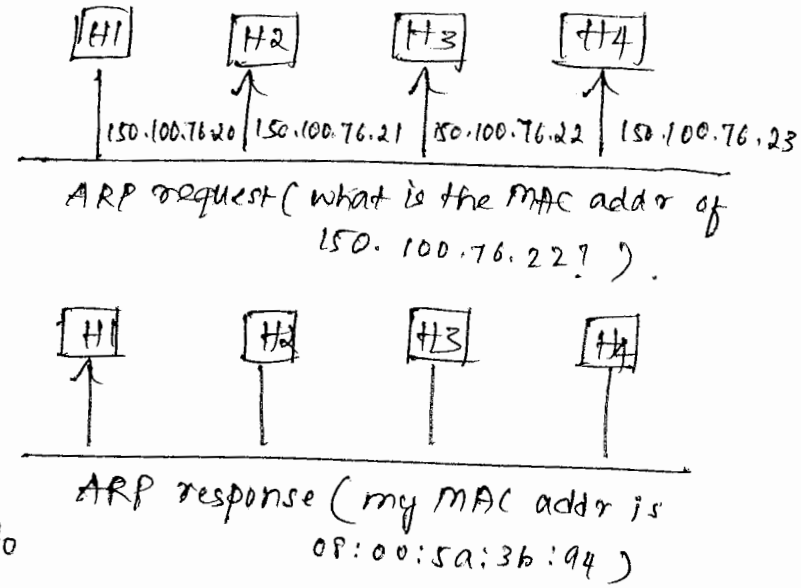
Soln:

200.96.86.0/20 = 11001000 01100000 01010110 0000 0000
200.96.87.0/20 = 11001000 01100000 01010111 0000 0000
200.96.88.0/20 = 11001000 01100000 01011000 0000 0000
200.96.89.0/20 = 11001000 01100000 01011001 00000000
mask = 11111111 11111111 11110000 00000000
resulting prefix is = 200.96.86.0/20

Address Resolution.

* Address Resolution protocol is the method for finding a host's link layer (hardware) address when only its IP or some other network layer address is known.
ie ARP helps the host map the IP address to MAC address.

* The main idea is illustrated in the figure.



* Suppose H1 wants to send IP packet to H3 but does not know the MAC addr of H3.
H1 first broadcasts an ARP request packet asking the dest^n host to reply.
All hosts in the n/w receives the packet, but only the intended host (H3) responds to H1. ARP response packet contains H3's MAC and IP addresses.

Reverse Address Resolution.

* Reverse address resolution protocol (RARP) is used to get an IP address ~~when~~ from an MAC address.
* works in fashion similar to ARP.
* To obtain IP addr, the host first broadcasts an RARP request packet containing its MAC addr. on the n/w.
All hosts on the n/w receives the packet, but only the server replies to the host by sending an RARP response packet containing the host's MAC and IP addresses.

Left Topics.

- Fragmentation and Reassembly
- ICMP: error and control messages.

UNIT 5 : NETWORK MANAGEMENT, SECURITY

CHAPTER 5A: NETWORK MANAGEMENT

* Network Management Overview
* SNMP
* Structure of Management information.
* MIB
* Remote Network Monitoring

CHAPTER 5B: SECURITY PROTOCOLS

* Security and cryptographic algorithms
* Security protocols
* Cryptographic algorithms

-7 Hours.

Ashok Kumar K,
VIVEKANANDA INSTITUTE OF TECHNOLOGY

NETWORK MANAGEMENT

* Network management involves configuring, monitoring, and possibly reconfiguring components in a network with the goal of providing optimal performance, minimal downtime, proper security, accountability, and flexibility.

* The functions performed by network management system can be categorized into following five areas:

1. Fault management
   - refers to the detection, isolation, and resolution of network problems.
   - Fault mgmt provides a means for improving reliability of the n/w.
   - Ex: detecting a fault in transmission link or n/w component, reconfiguring the n/w during the fault to maintain service level, & restoring the n/w when the fault is repaired.

2. Configuration management
   - refers to the process of initially configuring a n/w and then adjusting it in response to changing n/w requirements.
   - important area of n/w mgmt
   - ex: configuration of various parameters on n/w interface.

3. Accounting management
   - involves tracking the usage of n/w resources.
   - For ex, one might monitor user load to determine how to better allocate resources. Alternatively, one might examine the type of traffic or level of traffic that passes through a particular port.
   - Accounting management also includes activities such as password administration and charging.

4. Performance management

- involves monitoring n/w utilization, end-to-end response time, and other performance measures at various points in a network.
- results of the monitoring can be used to improve the performance of the network.
- Ex: tracking ethernet utilization on all switched interfaces & reconfiguring new switched interfaces if performance is deemed to be below some specific level.

5. Security Management

- refers to process of making the network secure.
- This process, offcourse, involves managing the security services that pertain to access control, authentication, confidentiality, integrety, & non repudiation.

Note:

* Figure shows a portion of a departmental n/w to illustrate how the n/w mgmt concepts might apply.

- An agent is part of n/w mgmt system & resides in a managed device. (routers, bridges, switches, hosts)
An agent's tasks are to provide management information about the managed device and to accept instructions for configuring the device.
It is also possible too an agent to not reside in managed device; such an agent is called proxy agent.

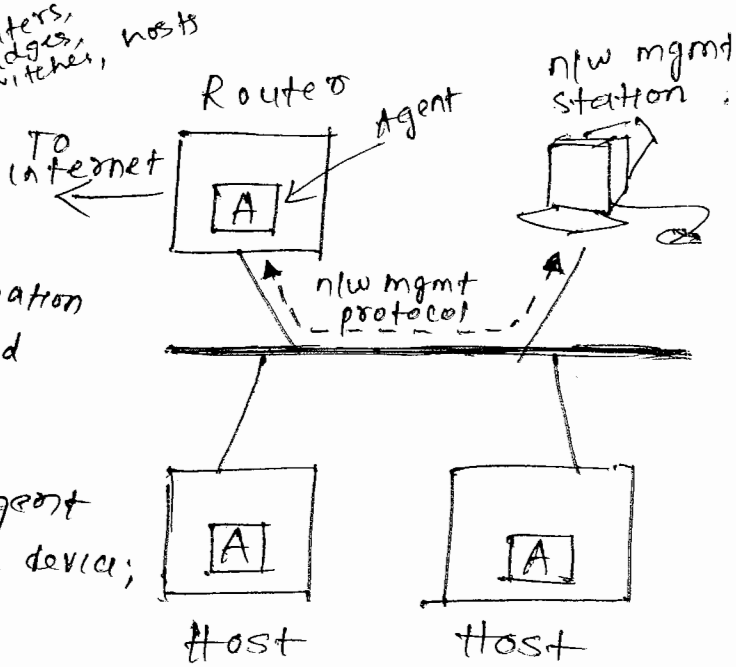


fig: A managed n/w.

- An n/w mgmt station provides a text or graphical view of entire network. The manager exchanges management

SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

* In early days of the internet, the internet Activities board recognized the need for a mgmt framework by which to manage TCP/IP implementations. The framework consists of three components.

1. SMI (structure of mgmt information) - A conceptual framework that defines the rules for describing mgmt information.
2. MIB (management information Base) - A virtual database containing information about the managed device
3. SNMP - A protocol for communication between an manager and an agent of managed device.

* SNMP is an application layer protocol that is used to read and write variables in an agent's MIB.
* SNMP is based on an asynchronous request-response protocol enhanced with trap directed polling.
  - asynchronous refers to the fact that the protocol need not wait for a response before sending other messages.
  - Trap directed polling refers to the fact that manager polls in response to a trap message being sent by an agent <Transport mode in SNMP is UDP (connectionless manner)>

note: SNMP manager sends messages to an agent via UDP destination port 161, while an agent send trap messages to a mgr via UDP dest^n port 162.

* The messages (PDUs) exchanged via SNMP consists of a header & data part.
  Header part contains a version field, a community name field, & a PDU type field.
  ↳ transmits a clear text password blw a mgr & an agent, so this field can serve as a limited form of authentication.

* SNMP provides three ways to access management information.

1. Request/Response interaction in which a manager sends an request to an agent and an agent responds to the request.
   The request is usually to retrieve or modify mgmt information associated with n/w device in question.
   Specific information is requested by manager, using one of the following requests.

- GetRequest-PDU for requesting info on specific variables
- GetNextRequest-PDU for requesting the next set of info (usually from a mgmt info. table)
- GetBulkRequest-PDU for requesting bulk info. retrieval This request was introduced in SNMPv2 to allow the retrieval of as much info. as possible in a packet.
- SetRequest-PDU for creating or modifying mgmt info.

The agent must always reply using a Response-PDU

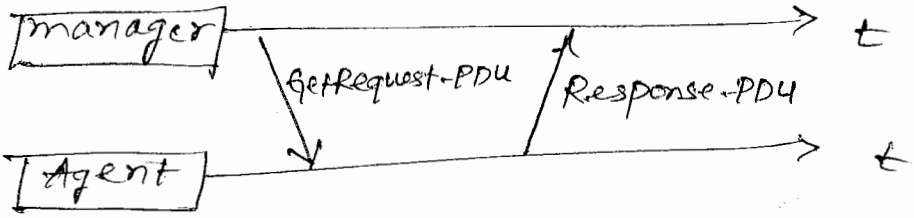
2. Request/Response interaction in which a mgr sends a request to another mgr & the latter responds to the requ
   The request is usually to notify a mgr of management info associated with the mgr, using InformRequest-PDU.
3. Unconfirmed interaction in which an agent sends an unsolicitated Trap-PDU to a manager. This request is usually to notify the manager of an exceptional situation that has resulted in changes to the management information associated with the network device.

* Below Table shows another way of looking at the mgr/agent roles & the functions they can perform.

Table: Request/Response interaction by role

Role	Generate	Receive
Agent	Response-PDU, Trap-PDU	GetRequest-PDU, GetNextRequest-PDU GetBulkRequest-PDU, SetRequest-PDU InformRequest-PDU
manager	GetRequest-PDU, GetNextRequest-PDU, GetBulkRequest-PDU SetRequest-PDU, InformRequest-PDU	Response-PDU Trap-PDU.

* Below fig shows Typical request/response interaction.



STRUCTURE OF MANAGEMENT INFORMATION (SMI)

* SMI defines the rules for describing managed objects.

(or) SMI Language is used to define the rules for naming objects and to encode objects in a managed n/w center.

(or) SMI is a language by which a specific instance of the data in a managed n/w center is defined.

for ex Integer32 means 32-bit integer with a value b/w $+2^0$ and $+2^{31}-1$

* collections of related objects are defined in MIB modules.

* The modules are written using a subset of Abstract Syntax Notation one (ASN.1), which describes the data structures in a m/c dependent language.

* SNMP uses the Basic Encoding Rule (BER) to transmit the data structures across the n/w unambiguously.
* Table below shows some of the datatypes permitted in SMI

INTEGER - A 32 bit integer.

OCTET STRING - A string of zero or more bytes with each byte having value b/w 0-255.

Display STRING - A string of zero or more bytes with each byte being a char. from the NVT ASCII set.

NULL - A variable with no value.

OBJECT IDENTIFIER - An authoritatively defined datatype described below.

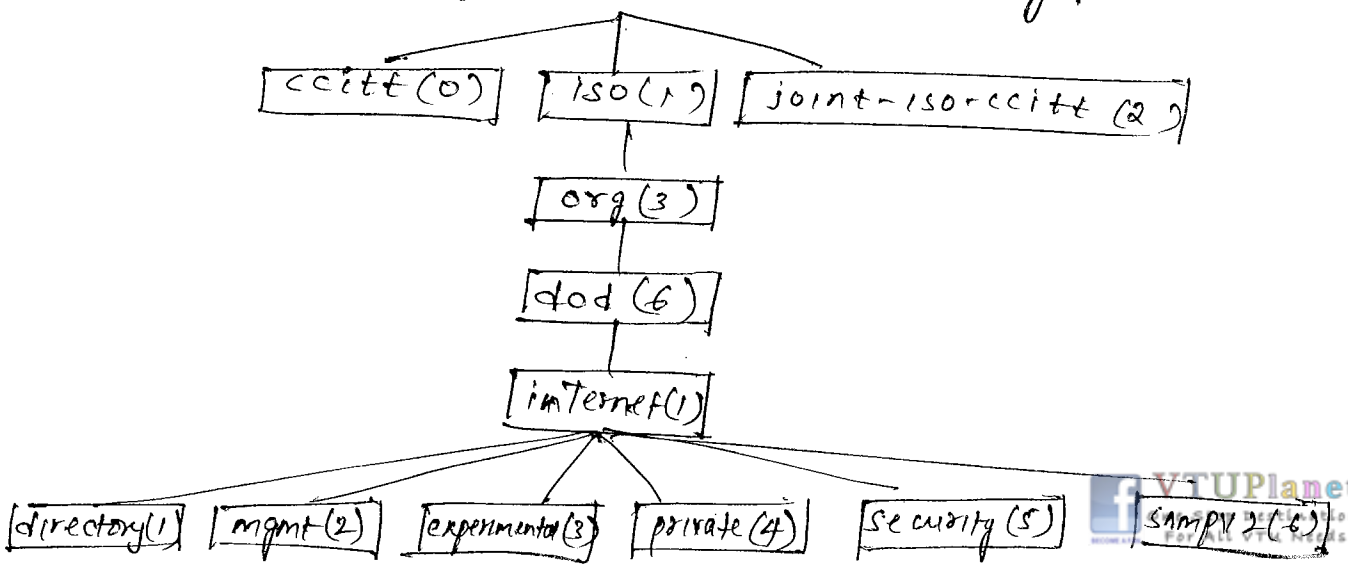
IpAddress - A 32-bit Internet address represented as an octet string of length 4.

Counter - A non-ve integer that increases from 0 to 2^{32} & then wraps back to 0.

Gauge - An non-ve integer that can increase/decrease, but which latches at a max. value.

TimeTicks - A non-ve integer that counts the time in hundredths of a second since some epoch.

Opaque - An opaquely encoded data string.



MANAGEMENT INFORMATION BASE (MIB)

* MIB is a virtual database used to define the functional and operational aspects of n/w devices. The database should contain an object for each functional aspect of a device that needs to be managed. These objects are usually grouped into different information modules.
* Each defn of a particular object contains the following info. about the object.
  - its name
  - the datatype
  - a human-readable description
  - type of access (read/write)
  - object identifier.
* objects are organised in an hierarchial manner & are identified by ASN.1 object defn language.
* Ex: IP(4) subtree refers to an ip group that contains a no. of object defn pertaining to common ip objects. One such object is

```
ipInHdrErrors   OBJECTTYPE
    SYNTAX      counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        " The no. of input datagrams discarded due to errors in their IP headers, including bad checksums, version no mismatch, other format errors, TTL exceeded, errors discovered in processing their IP options etc "
    ::= { IP4 }
```

* we see that a mgr can use this managed info. to obtain statistics of the no. of IP packets that are discarded because of various errors.

CHAPTER 5B

SECURITY PROTOCOLS

Security protocol provides a secure communication service that prevents eaves-droppers from reading or ~~alternat~~ altering the contents of messages and prevents imposters from impersonating legitimate users.

SECURITY AND CRYPTOGRAPHIC ALGORITHMS

* Fig shows several threats that can ~~arrive~~ arise in a network setting

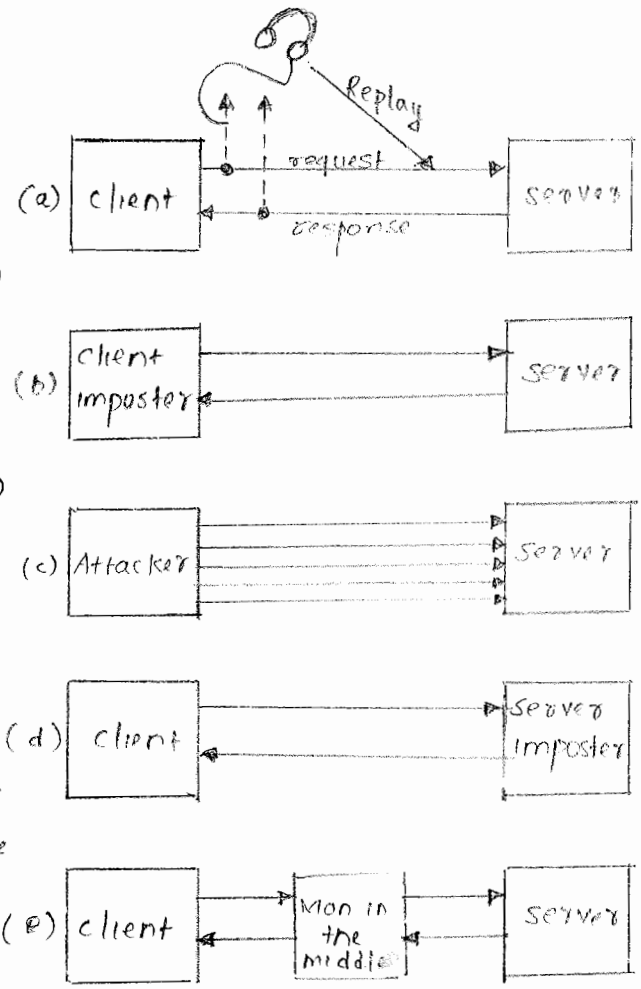
a:) eaves dropping:
Information transmitted over the nlw is not secure & can be observed and recorded by eavesdroppers. This information can be replayed in attempts to access the server.

b:) client imposter:
Imposters can attempt to gain unauthorized access to server. for ex: bank account

c:) Denial of service:
Attacker can also flood a server with requests, overloading the server resources and resulting in denial of service to legitimate clients.

d:) server imposters:
imposters can impersonate a legitimate server & gain sensitive information from clients eg: bank acc no. & associated user password.

e:) Man-in-the middle:
An imposter manages to place itself as the man-in-the-middle, convincing the server that it is the legitimate client, and the



* These threats gives rise to one or more of the following security requirements for information that is transmitted over a n/w.

1.) privacy or confidentiality : Information should be readable only by the intended recipient.

2.) Integrity : Recipient of the information should be able to confirm that a message has not been altered during transmission

3.) Authentication : It should be possible to verify that the sender or receiver is who he or she claims to be.

4.) Non repudiation : The sender ~~has to~~ cannot deny having sent a given message.

Applications of cryptography to security.

* Terminologies :

1.) Cryptography : science and art of manipulating messages to make them secure.

2.) Plain text : original message to be transformed.

3.) cipher text : Resulting message after transformation.

4.) Encryption : process of converting the plain text into cipher text.

5.) Decryption : Reverse process of encryption

5.) cipher : Algorithm used for encryption and decryption.

7.) secret key : A user can recover the original message only by decrypting the ciphertext using secret key.

* Some ex for cipher are

-> substitution ciphers : Each letter of the alphabet is mapped into another letter.
for ex: a b c d .... z
↳ z y x w .... a

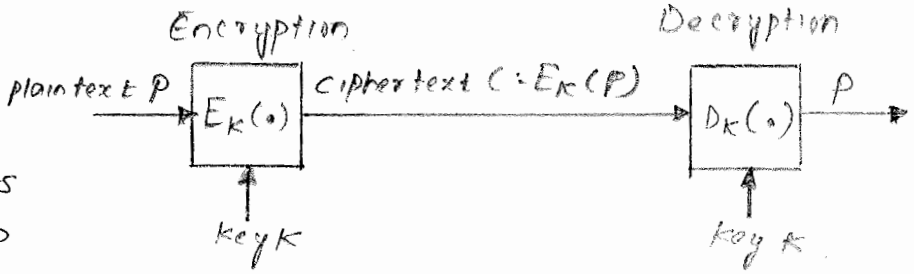
-> Transposition ciphers : order in which the letters of the message appear is altered.
for ex: letters may be written into array in one order & read out in different order.

note: these techniques can easily be broken.

* modern encryption algorithms depend on the use of mathematical problems that are easy to solve when a key is known and that becomes extremely difficult to solve without the key.
There are several types of encryption algorithms. Two of them are discussed here
  - Secret key cryptography
  - Public key cryptography.

Secret key cryptography (symmetric key cryptography)

* Fig depicts secret key cryptographic system, where a sender converts the plain text P into cipher text $C = E_K(P)$ before transmitting the original message over an insecure channel.



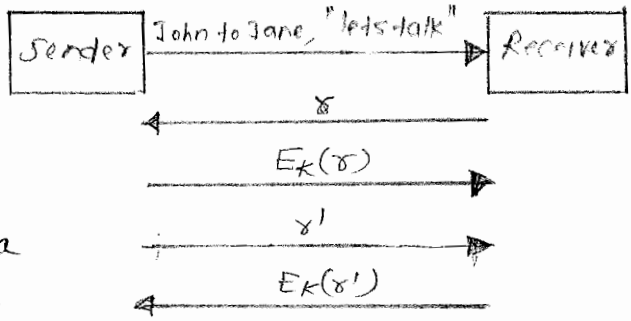
* The sender uses secret key K for encryption.
When the receiver receives the ciphertext C, it recovers the plaintext by performing decryption $D_K(C)$, using the same key K.
It is the sharing of the secret, ie the key, that enables the transmitter & receiver to communicate.
* Symbollically we can write $P = D_K(E_K(P))$
* E_K: DES

note: Show how secret key cryptography meets the following security requirements ?
  - privacy or confidentiality
  - Authentication.
  - Integrity.

1. Security

- A message that needs to be kept confidential is encrypted prior to transmission, and any eavesdropper that manages to gain access to the ciphertext will be unable to access the contents of plain text message.

2. Authentication



- Figshows the secret key authentication.
- Transmitter sends a message identifying itself. The receiver replies with a message that contains a random number r. This process is called challenge.

The transmitter then sends an response with an encrypted version of the random number. The receiver applies the shared key to decrypt the number. If the decrypted number is r, then the receiver knows that it is communicating with the given transmitter.

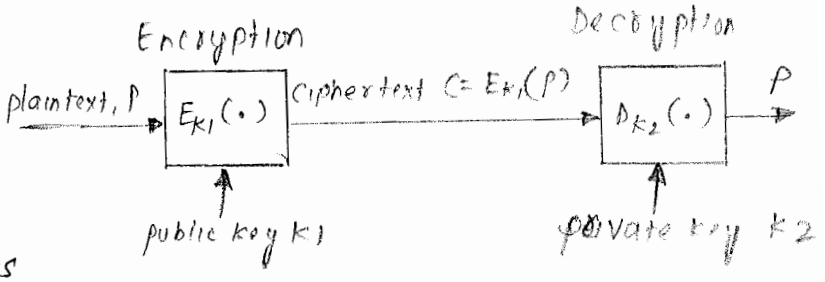
- Transmitter may also wish to authenticate the receiver by issuing a challenge by sending its own random number.

3. Integrity

- Usual approach to providing integrity is to transmit a cryptographic checksum or hash along with the unencrypted message.
- To ascertain integrity, the receiver calculates the checksum of the received message and compares it to the received checksum. If the checksum agree, the message is accepted.
- Ex. for hash algorithm are
  - message digest 5 (MD5) algorithm
  - Keyed MD5 algorithm
  - secure hash algorithm 1 (SHA-1) algorithm.

public key cryptography (asymmetric cryptography)

* Figure depicts public key cryptography



* Unlike secret key cryptography, keys are not shared b/w senders and receivers in public key cryptography.
* This was invented in 1975 by Diffie and Hellman.
* It relies on two different keys
  - public key
  - private key.
* A sender encrypts the plain text by using a public key, and a receiver decrypts the ciphertext by using a private key as shown in the figure.
* Symbolically, a public key cryptographic system can be expressed as $P = D_{K2}(E_{K1}(P))$ where
  K1 → public key and K2 → private key.
  In some systems, encryption & decryption process can be applied in reverse order such as $P = E_{K1}(D_{K2}(P))$
* One imp requirement for public key cryptography is that it must not be possible to determine K2 from K1.
  In general public key is small and private key is large.
* Ex for public key cryptography is RSA.

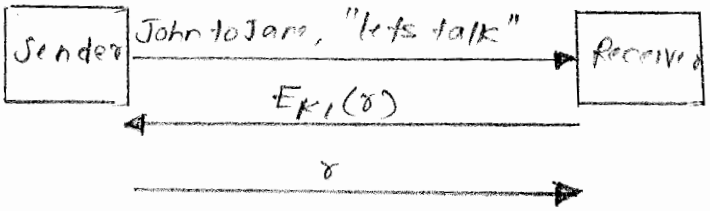
note1: show how public key cryptography meets the following security requirements?

- privacy
- Authentication.
- Non repudiation.

1. Privacy.

- Transmitter uses the public key K_1 to encrypt its message P and then transmits the corresponding ciphertext $E_{K_1}(P)$ to the receiver.
- only the holder of the private key K_2 can decrypt the message $P = D_{K_2}(E_{K_1}(P))$. Therefore privacy of the message is assured.
- Similarly the messages from receiver to transmitter are kept private by encrypting ~~with~~ them with transmitter's public key.

2. Authentication



- Fig shows public key authentication.
- Transmitter begins by identifying itself. The receiver picks a nonce r, (term nonce is derived from number once, & it describes the desired random numbers) encrypts it by using the transmitter's public key, & issues a challenge. The transmitter uses its private key to determine the nonce & responds with the nonce r.

3. Non Repudiation.

- public key cryptography provide nonrepudiation by producing a digital structure.
- To sign a message, the transmitter first produces a noncryptographic checksum or hash of the message. The transmitter then encrypts the checksum or hash using its private key to produce the signature. No one else can create such a signature. The transmitter then sends the message & the signature to the receiver. Then the receiver confirms the signature as follows
  - receiver first applies publickey encryption algo. to the signature to obtain a checksum
  - receiver then computes the checksum directly from the message.
  - If two checksum agree, then only the given transmitter could have issued the message.

note 2: In public key cryptography, integrity is not assured since an intruder can intercept the message from the transmitter & inserts a new message using the public key K_1.

soln ? = Have the transmitter encrypt the message with its private key K_2' & transmit $E_{K_1}(P')$, where $P = D_{K_2'}(P)$. No intruder can successfully alter this cipher text, and the receiver can decrypt it by applying D_{K_2} to $E_{K_1}(P')$ to recover P', followed by $E_{K_1'}(P') = E_{K_1'}(D_{K_2'}(P))$ to recover P.

note 3:

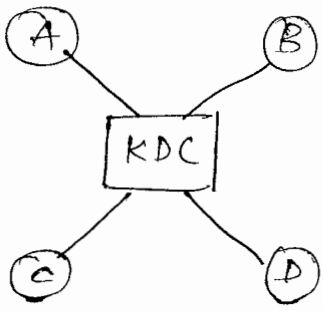
secret key cryptography	public key cryptography
* less powerful	* more powerful (∵ provides digital signatures)
* much faster	* much slower.

Key Distribution

Key Distribution in case of secret key systems.

* General approach: every pair of users share a separate key.
  disadv: no of keys can grow as the square of number of users.
  soln: KDC (key distribution center)

* Fig shows KDC.



* Here every user has a shared secret key with the KDC.
* If user A wants to communicate with user B, user A contacts the KDC to request a key for use with user B.
* KDC authenticates user A, selects a key K_{AB} & encrypts it by using the key

with A and B to produce $E_{KA}(K_{AB})$ and $E_{KB}(K_{AB})$.
KDC sends both versions of encrypted key to A.
Finally, user A can contact user B and provide a ticket in the form of $E_{KB}(K_{AB})$ that allows them to communicate securely.

Key Distribution in case of public key syskms.

* requires only one pair of keys per user. But they still face the problem of how the public keys are to be distributed. (public keys ~~are~~ must be certified some how)

Soln: Establish a certification authority (CA).

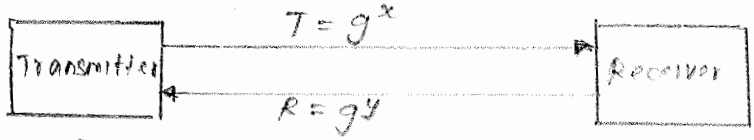
* The function of a CA is to issue certificates that consists of a signed message, stating the name of a given user, his/her public key, a serial number identifying the certificate, & an expiration date.
The certificates can be stored anywhere. They can be accessed through directory service.
* Each user is initially configured to have the public key of CA.
To communicate with user B, user A contacts a server to obtain a certificate for B. The signature in the certificate authenticates the message & its integrity.

Key Generation: Diffie-Hellman exchange.

* Its an alternative to key distribution using KDC or CA.
* The procedure assumes that the transmitter & receiver have agreed on the use of a large prime number p ~~are~~ (about 1000 bits long), and a generator number g that is less than p.
* The Transmitter picks a random number x and calculates $T = g^x \text{ modulo } p$.
Illy the receiver picks a random number y and calculates $R = g^y \text{ modulo } p$.

* The transmitter sends T to the receiver, and the receiver sends R to the transmitter.
* At this point, the transmitter & receiver both have T and R, so they can compute the following numbers.
- Transmitter calculates R^x modulo p
$= (g^y)^x$ modulo p
$= g^{xy}$ modulo p
$= K$.
- The receiver calculates T^y modulo p
$= (g^x)^y$ modulo p
$= g^{xy}$ modulo p
$= K$.

* Therefore, both transmitter & receiver arrive at the same number, K. as shown in the fig.



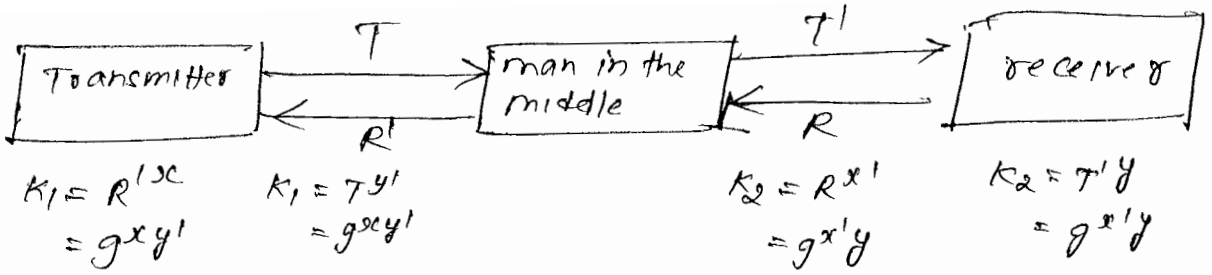
$K = R^x \bmod p$ $= g^{xy} \bmod p$
$K = T^y \bmod p$ $= g^{xy} \bmod p$

Fig: Diffie-Hellman exchange.

* An eavesdropper would have p, g, T & R available, but neither x nor y. To obtain this, $x = \log_g(T)$, $y = \log_g(R)$ which is very difficult to do for large numbers.
* Thus the transmitter & receiver jointly develop a shared secret key K, which they can use in subsequent security operations.

Disadvantages

* The algorithm is computationally intensive. ie it needs many multiplications for large prime no. p. This results in flooding attack, & leads to denial of service for legitimate clients.
* The algorithm is also susceptible to a man-in-the-middle attack. (refer fig)
- Suppose the intruder is able to intercept T and R.
- Then it keeps R & sends $R1 = g^{y'}$ modulo p to the transmitter. At this point, a transmitter & intruder have established



- similarly, the intruder can establish a shared secret key K_2 with the receiver based on x' and y.
- From now on, the intruder is privy to all communication blw transmitter & receiver, and the transmitter & receiver can do nothing to detect the intruder.

SECURITY PROTOCOLS

* Security protocols refers to a set of rules governing the interaction between peer processes to provide a certain type of security service.

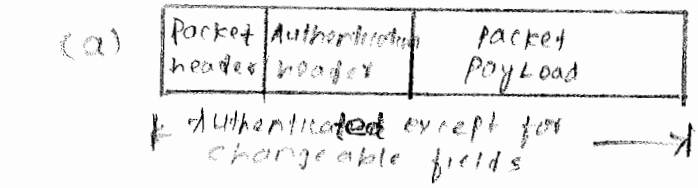
Types of service.

* Here we are concerned with three communication security requirements
  - integrity
  - Authentication
  - privacy.

ie we discuss how a communication service can provide integrity, Authentication and privacy.

Integrity and Authentication service

* Fig below shows typical packet structure for providing integrity and authentication service.



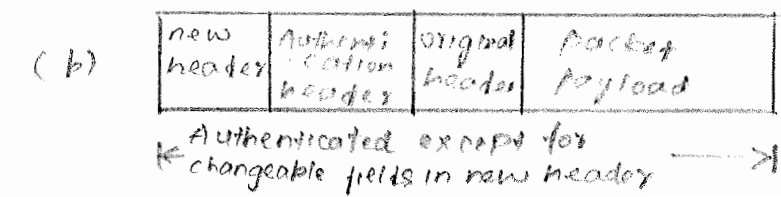
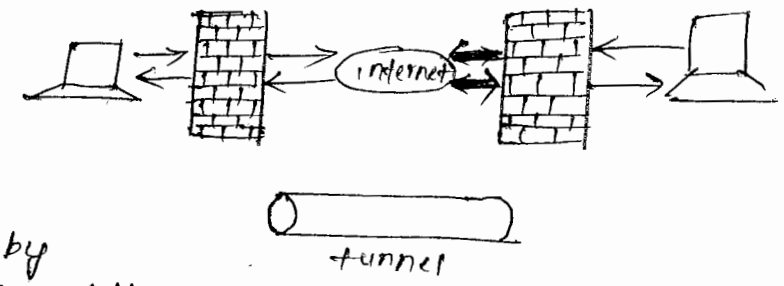


fig: packet structure for authentication and integrity service.

* An authentication header is sandwiched in between the normal packet header and its payload (SDU)
The normal header includes a field set to indicate the presence of the authentication header.
Authentication header itself contains a
  - field that identifies the security association
  - field for a sequence number (provide protection against replay attacks)
  - field for cryptographic checksum.
* Ideally, the cryptographic checksum should cover the entire packet. However some fields in the packet header need to be changed while the packet traverses the n/w, and so these needs to be excluded.
* The cryptographic checksum algorithm is applied to fig (a) with the values in the changeable fields and the cryptograph checksum field set to zero. The resulting checksum is then inserted in the authentication header and the packet is transmitte
* To verify integrity and authentication, the receiver recalculates the cryptographic checksum based on the received packet and the shared secret key, with the appropriate fields set to zero.
* If the recalculated checksum and the received checksum do not agree, the packet is discarded and an event is recorded in an audit log.

~~Privacy service~~

note: A tunnel can be established to provide security b/w two gateways (ref fig)



The tunnel is established by encapsulating a packet inside another packet as shown in fig(b) page 18.

privacy service

* note that integrity and authentication service do not prevent eavesdroppers from reading the information in each packet. Encryption is needed to provide privacy in the transmission of packet information.

* fig(a) shows a typical packet structure where an encryption header is ~~waste~~ inserted after the normal header.

* the normal header contains a field ~~to~~ set to indicate the presence of an encryption header.

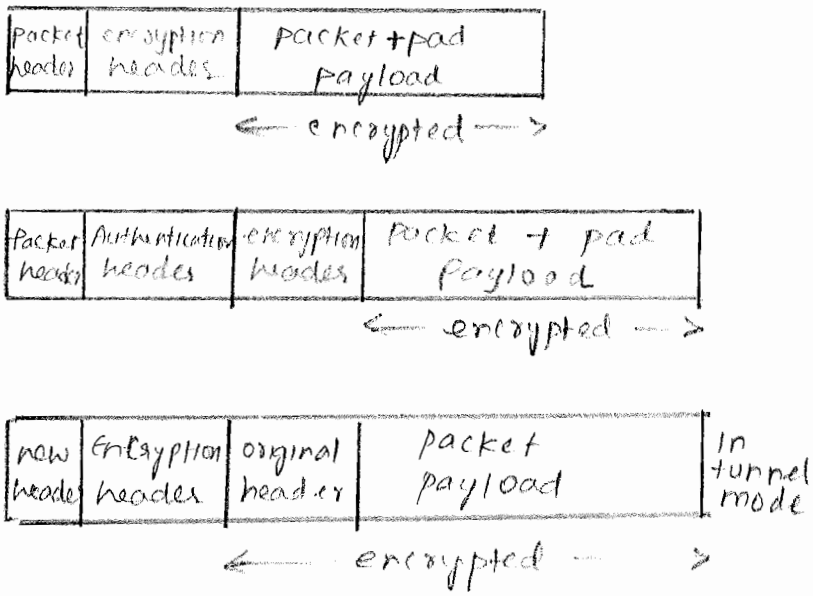


fig: packet structure for privacy

The encryption header contains fields to identify the security association and to provide a sequence number.

* This header is followed by encrypted version of the payload

* Note that packet header and encryption header are not ~~encryption~~ encrypted, so privacy is provided only by the upper layer protocols encapsulated in the payload.

* The receiver takes each packet and decrypts the payload. If the payload portion has been altered during transmission, then the decrypted message will differ from original message such changes may be detectable by the higher layer protocol if the decrypted message does not make sense.

* fig(b) shows packet structure that can be used to provide privacy for the payload, and, integrity & authentication for the overall packet.

* These packet structures allow eavesdropper to read the information in the packet header.
A stronger form of privacy is obtained by using encryption in a tunnel mode as shown in fig (c).

Setting up a Security Association

* Security Association is the establishment of shared security information b/w two network entities to support secure communication.
* Internet key exchange (IKE) : It is the protocol used to setup security association in the IPsec protocol suite. IKE allows two hosts to establish a security association and a common secret key independently of other hosts or servers. It does this by using Diffie hellman exchange.
* IKE overcomes disadv of Diffie hellman exchange by adding an authentication ~~setup~~ step after the Diffie-Hellman exchange that can detect the presence of man-in-the-middle and by using "cookies" to thwart denial-of-service attacks.

*

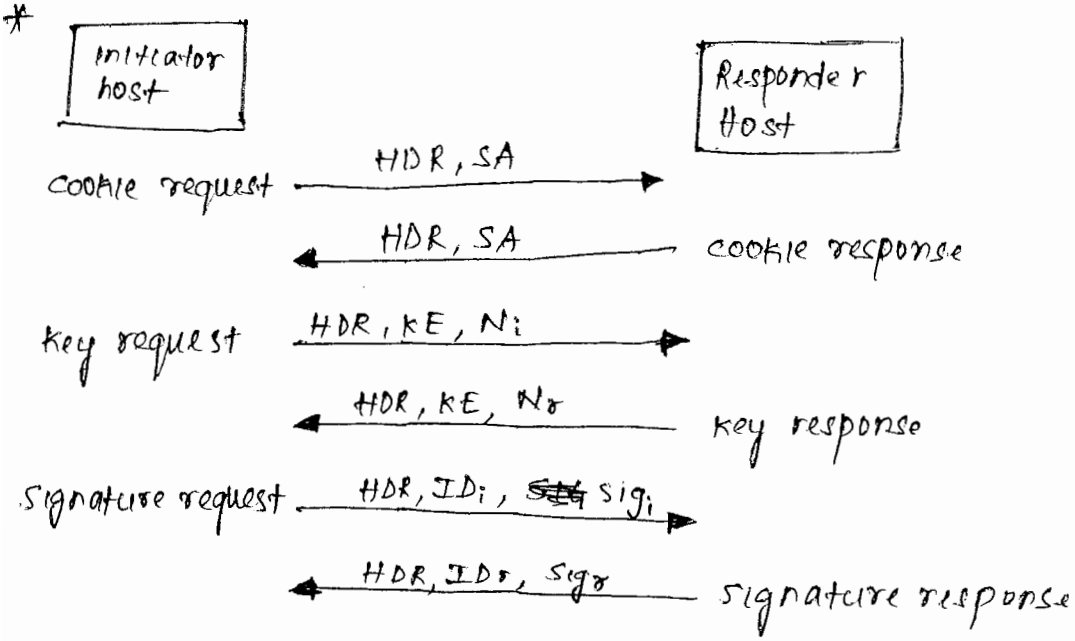


fig: Establishing a security association.

* Fig shows a series of message exchanges b/w two hosts to establish a security association (SA)

* Note the following about the exchanges of ~~the~~ messages.

1.)
- Initiator generates a initiator cookie C_i (unique pseudo random number, of say, 64 bits) ~~asso~~ and associates this value with the expected address of the responder, and sends a cookie request message to the responder requesting a security association.
- HDR (header) contains initiator cookie C_i. SA field in the message offers a set of choices regarding encryption algo, hash algo, authentication method etc

2.)
- Responder checks to see whether the initiator cookie is not already in use. If not, it generates responder cookie C_r and associates this value with the expected address of the initiator, and replies with cookie response message in which it selects one of the offered choices in the initiators SA's field.
- HDR includes both C_i & C_r.

3.)
- Upon receiving the response, initiator checks the address & initiator cookie.
- From now, initiator will identify ~~HDR~~ the SA by the pair (C_i, C_r)
- At this point, it records the association as "unauthenticated".
- Next the initiator sends a key request message including its public Diffie hellman value $T = g^a$ modulo p & a nonce N_i

4.) - Responder checks responder cookie in the arriving message.
- If cookie is valid, SA will henceforth be identified by pair (C_i, C_r)
- At this point, the association is recorded as "unauthenticated".
- It sends a key respond message with its public value $R = g^b$ modulo p and an nonce N_i.

5.)
- At this point, Both generates secret const $K = g^{xy}$ modulo p.
- Both parties, also ~~generates~~ computes a secret string of bits SKEYID known only to them:
(for ex: they may obtain a hash of concatenation of N_i, N_r, & K
SKEYID might be 128 bits long)

6.)
- initiator now prepares a signature stating what it knows: namely, SKEYID, T, R, C_i, C_r; contents of SA field; and initiator identification. (for eg: initiator can obtain a hash of the concatenation of binary repres^n of all these information)
- The initiator identification and the signature are then encrypted & sends this information in a signature response message.

7.)
- Responder decrypts the message & recalculates the hash of its version of the shared information. If the recalculated hash agrees with received hash, then the initiator & Diffie-hellman public values have been authenticated.
- SA and keys are recorded as authenticated.
- A man-in-the middle would have been detected at this point, since it could not have knowledge of SKEYID which is derived from K

8.)
- Responder now prepares its signature stating what it knows. namely SKEYID, R, T, C_i, C_r; the contents of SA field, and responder identification.
- Responder identification & the signature are then encrypted, & the signature response message is sent to the initiator.

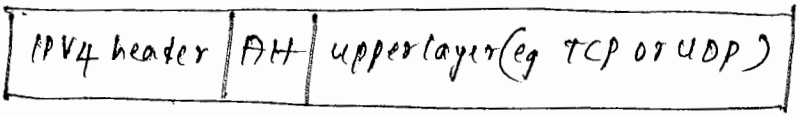
9.)
- initiator recalculates the hash of its version of shared inform to authenticate the responder as well as ~~both~~ the values of Diffie-hellman public values.
- At this point, the SA is established.
- SA and keys are recorded as authenticated.

IPsec (IP Security)

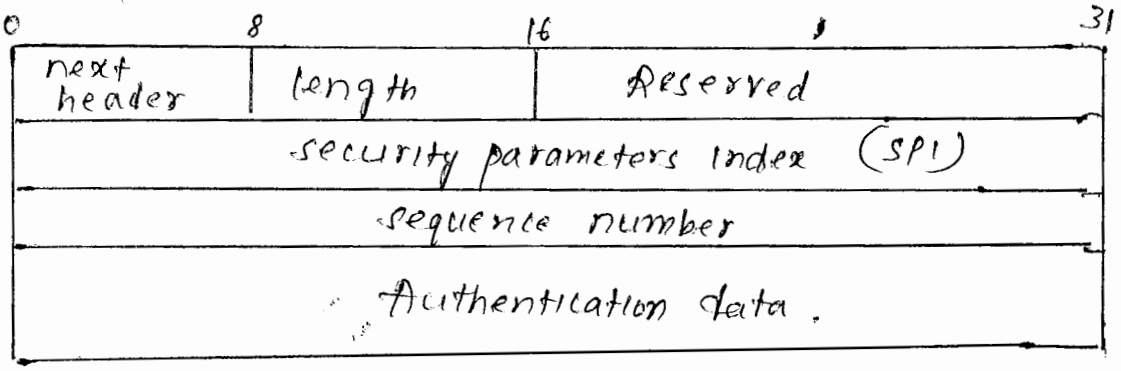
* The goal of IPsec is to provide a set of facilities that support security services (like authentication, integrity, confidentiality, & access control) at the IP layer.
* IPsec uses two protocols to provide traffic security
  - Authentication header
  - Encapsulating security payload.
* Each protocol can operate in either Transport mode or tunnel mode.

Authentication header. (AH)

* provides Authentication and integrity to an IP packet.
* location of AH is after the headers that are examined at each hop and before any other headers that are not examined at intermediate hop.
* with IPV4, the AH immediately follows IPV4 header as shown



* protocol value in IP header is set to 51 to identify the presence of an AH following the IP header.
* Format of AH is shown below.



next header - used to identify next payload after the AH.
length - indicates length of authentication data in multiples of four octets.
SPI - identifies the SA for this packet.
sequence number - its value is incremented by one for each packet sent purpose is to protect against replay attacks.
Authentication data - Result of authentication algo. is placed in this field.

Transport layer Security (TLS)

* TLS and its predecessor secure sockets layer (SSL) are cryptographic protocols that provides security and data integrity for comm over n/w such as internet. (or) TLS & SSL provides secure HTTP connections.

* As shown in the fig, TLS protocol consists of protocols that operate at two layers: the TLS record protocol & TLS handshake protocol, along with the change cipher spec protocol & Alert protocol.

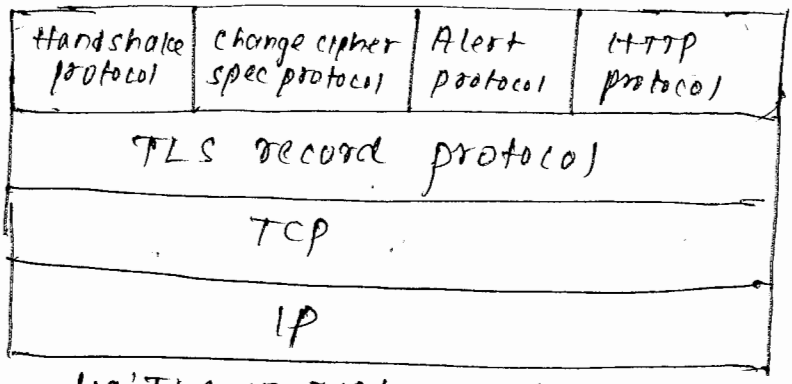
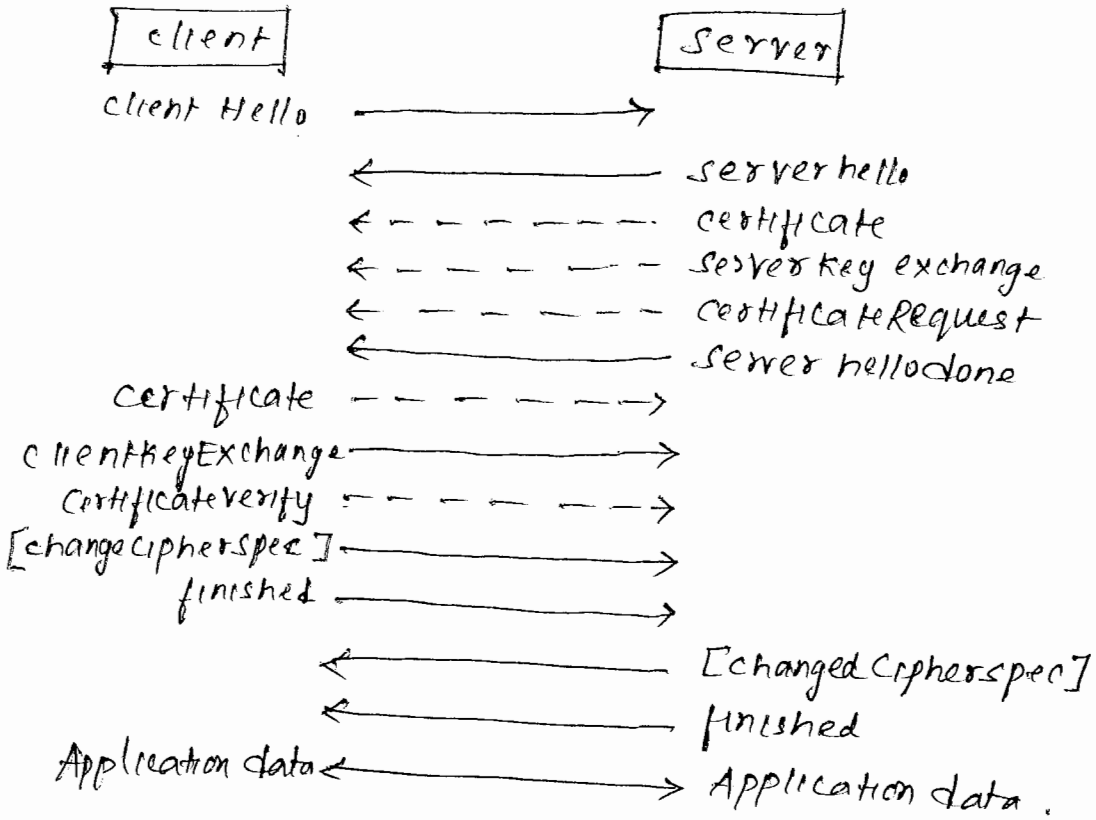


fig: TLS in TCP/IP protocol stack.

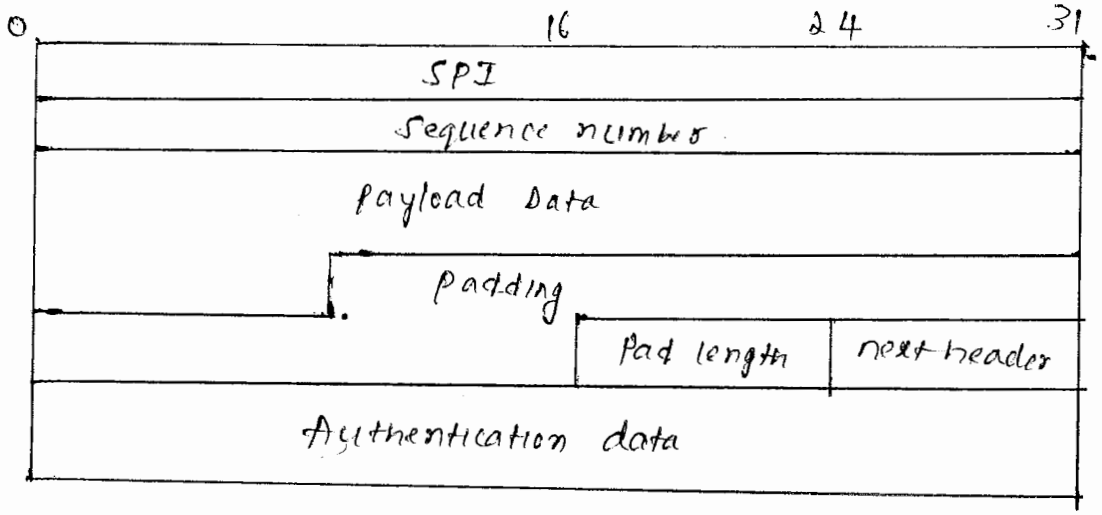
TLS Handshake protocol.

* Fig shows TLS handshake process.



Encapsulating security payload. (ESP)

* ESP provides confidentiality, authentication, and data integrity.
* ESP can be applied alone or in combination with an AH
* Format of ESP is shown here.



* Authenticated coverage starts from SPI field until next header field. and the Encrypted coverage starts from payload data field until next header field.
* protocol number immediately preceeding ESP is 50.
* SPI
  Sequence no } same as before.
  payload data - variable length data whose contents are specified in next header field.
  padding - optional, purpose is to make this format multiple of four octects.
  pad length - indicates length of padding field in octets.
  Authentication data - optional, purpose is to provide authentication serve.

note

* Dashed lines indicates optional or situation - dependent messag that are not always sent.
* The ChangeCipherSpec is not an TLS handshake message

step 1 :

client and server exchange hello messages to negotiate algorithms, exchange random values, and initiate or resume the session.

step 2 :

client and server exchange cryptographic parameters to allow them to agree on a premaster secret.
If necessary, they exchange certificates and cryptographic information to authenticate each other.
They can generate a master secret from the premaster secret and exchange random values.

step 3 :

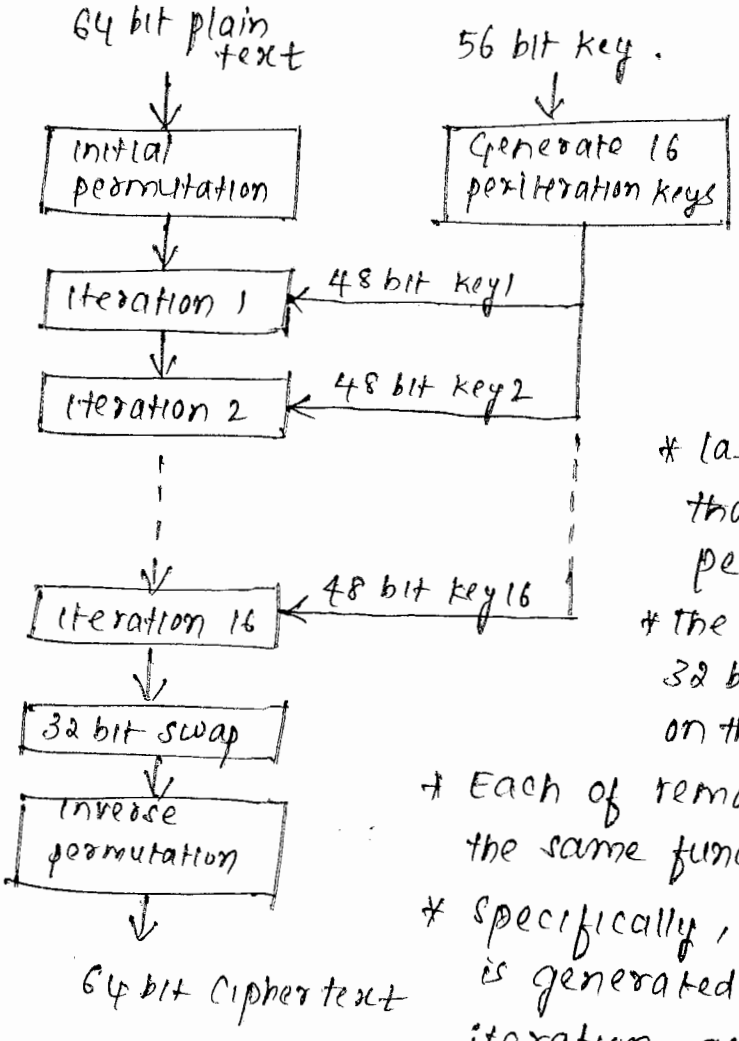
The client and server provide their record layer with the security parameters. The client and server verify that their peer has calculated the same security parameters and that the handshake occurred without tampering by an attacker.

CRYPTOGRAPHIC ALGORITHMS

- DES (Data Encryption Standard)
- RSA (Rivest, Shamir, and Adleman)

DES

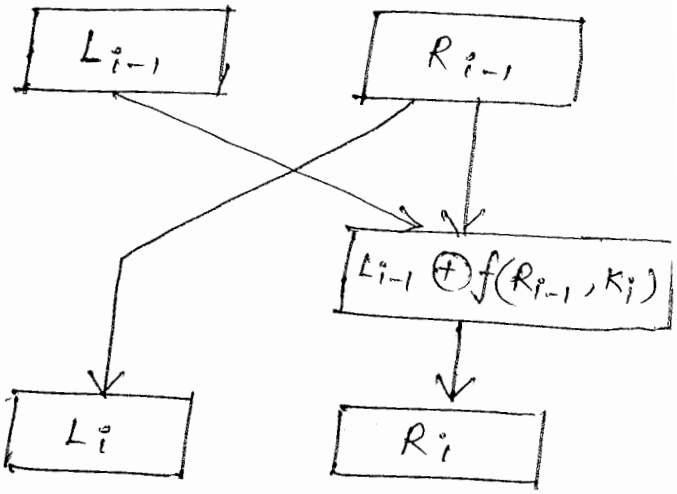
* Developed by IBM in early 1970s.
* most widely used shared key cryptographic system.
* in encryption process, DES first divides the original message into blocks of 64 bits. Each block of 64 bit plaintext is separately encrypted into a block of 64 bit ciphertext.
* DES uses 56 bit secret key. (generally agreed that 56 bits are too small to be secure).
* DES encryption algorithm which has 19 steps, is shown below. Decryption runs the algorithm in reverse order.



* Each step takes 64 bit I/p from the preceeding step & produces 64 bit o/p for next step.
* First step performs initial permutation of 64 bit plaintext that is independent of key.
* last step performs final permutation that is the inverse of initial permutation.
* the next-to-last stage swaps the 32 bits on the left with the 32 bits on the right.
* Each of remaining 16 iteration performs the same function but uses a different key.
* Specifically, key at each ~~step~~ iteration is generated from the key at preceeding iteration as follows -

- First a 56 bit permutation is applied to the key.
- Then the result is partitioned into two 28 bit blocks, each of which is independently rotated left by some number of bits
- combined result undergoes another permutation.
- Finally, a subset of 48 bits is used for the key at the given iteration.

* The operation at each iteration is as shown in the figure.

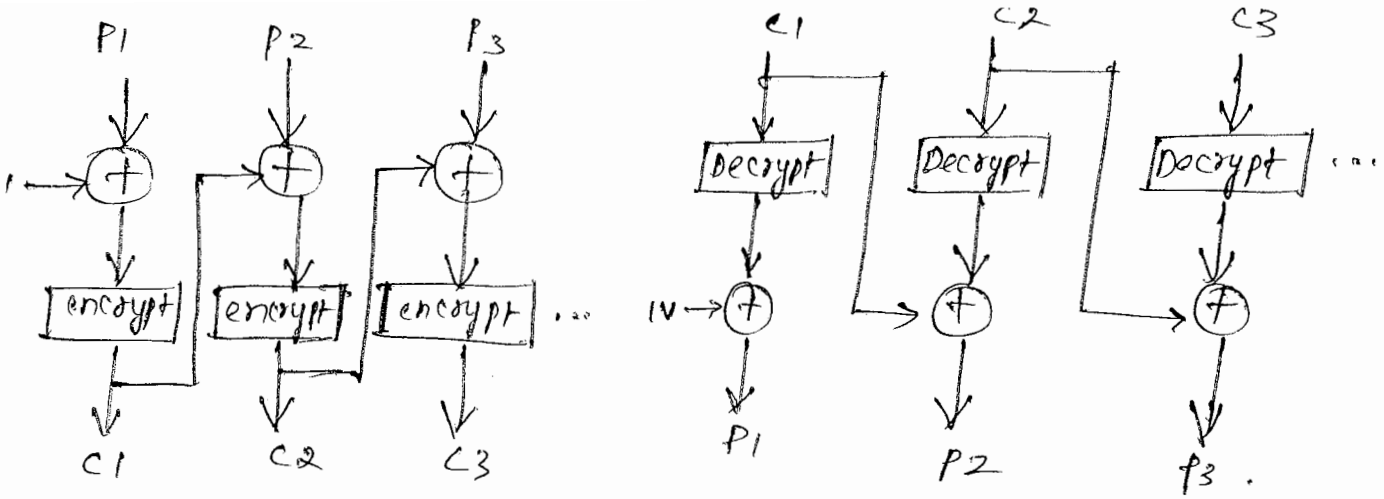


- 64 bit i/p is devided into equal portions denoted by L_{i-1} and R_{i-1}
- o/p generates two 32 bit blocks denoted by L_i and R_i
- Left part of the o/p is simply equal to right part of the input. Right part of the o/p is derived from bitwise XOR of the left part of the input and a function of the right part of i/p and the key at the given iteration.

note:

* The preceeding algorithm simply breaks a long message into 64 bit blocks, each of which is independently encrypted using the same key. In this scheme DES is said to be operating in Electronic codebook (ECB) mode. This mode may not be secure when the structure of the message is known to the attacker.

* Soln: introduce dependency among the blocks.
A simple way to introduce dependency is to XOR the current plain text block with the preceeding cipher text block. Such a scheme is called Cipher Block chaining (CBC) it is shown below.



(a) encryption. (b) Decryption.

IV → Initialization vector.

* Using 56 bit key makes DES to vulnerable to brute force attack.
* Soln: triple DES.
  ↳ uses two keys.

$$C = E_{K1}(D_{K2}(E_{K1}(P)))$$

$$P = D_{K1}(E_{K2}(D_{K1}(C)))$$

RSA

* widely accepted scheme for public key cryptography.
* utilizes modular arithmetic and factorization of large numbers
* public & private keys are generated based on following rules.

. choose two large prime no.s p and q such that the product is equal to n. The plaintext P, represented by a no. must be less than n. In practise, n is a few hundred bits long.

1. Find a number e that is relatively prime to (p-1)(q-1). Two nos are said to be relatively prime if they have no common factors except 1. The public key consists of {e, n}
3. Find a number d such that $de = 1 \bmod ((p-1)(q-1))$. In other words d and e are multiplicative inverses of each other modulo ((p-1)(q-1)). Private key consists of {d, n}

* For any integer $p < n$ the following key property holds.

$$p^{de} \pmod n = p \pmod n$$

* Suppose P is an integer that corresponds to a block of plaintext. RSA encrypts P as follows

$$\boxed{C = P^e \pmod n}$$

= integer b/w 0 and n.

* To decrypt the ciphertext C, RSA algorithm uses.

$$\boxed{P = C^d \pmod n}$$

$$= (P^e)^d \pmod n$$

$$= p^{de} \pmod n$$

$$= p \pmod n$$

problems.

1. Using RSA algorithm, encrypt the following

(a) $p=5$, $q=11$, $e=7$, $P=18$.

(b) $p=5$, $q=11$, $e=7$, $P=19$.

(c) $p=5$, $q=11$, $e=7$, $P=1$. Also decrypt them

Soln: $n = p*q = 5*11 \Rightarrow \boxed{n=55}$

$$C = P^e \pmod n$$

(a) $C = 18^7 \pmod{55} = \boxed{17}$

(b) $C = 19^7 \pmod{55} = \boxed{24}$

(c) $C = 1^7 \pmod{55} = \boxed{1}$

} encryption.

(a) $17^{23} \bmod 55 = \boxed{18}$

(b) $24^{23} \bmod 55 = \boxed{19}$

(c) $1^{23} \bmod 55 = \boxed{1}$

Decryption $\boxed{P = C^d \bmod n}$ for this we should calculate d.

multiplicative inverse of 7 modulo 40 yields $d = 23$.

2. Using RSA, encrypt the following

(a) $p=3$, $q=11$, $e=7$, $P=12$.

(b) $p=7$, $q=11$, $e=17$, $P=25$.

Find the corresponding d's and decrypt the ciphertext.

Ashwath Kumar K,
VIVEKANANDA INSTITUTE OF TECHNOLOGY.

UNIT 6 :

QOS, RESOURCE ALLOCATION, VPNS, TUNNELING, OVERLAYING NETWORKS

Syllabus

chapter 6A: QOS and Resource Allocation.

* overview of QOS
* Integrated services QOS
* Differentiated services QOS
* Resource Allocation.

chapter 6B: VPNS, Tunnelling, overlaying networks

* Virtual private Networks.
* Multi protocol label switching.
* overlaying n/w.

-7 Hours.

Ashok Kumar. K
VIVEKANANDA INSTITUTE OF TECHNOLOGY

~~CYE81~~

CHAPTER 6A : QOS AND RESOURCE ALLOCATION

OVERVIEW OF QOS

* motivation of QOS is to control access to available bandwidth and to regulate traffic (to avoid congestion)
* QOS is typically based on the best effort model, whereby a n/w provides no guarantee on the delivery of packets but makes its best effort ~~do~~ to do so.
* Approaches to providing quality support can be divided into
  - integrated services
  - Differentiated services.

INTEGRATED SERVICES QOS → provide QOS to individual applications & flow records.

* The integrated services approach consists of two service classes.
  - guaranteed service class, defined for applications that cannot tolerate a delay beyond a particular value. eg. voice or video communications.
  - controlled load service class, is used for applications that can tolerate some delay and loss.

* Fig shows four common categories of processes providing QOS.

Note: Here, providing QOS requires certain features to be maintained in switching nodes.

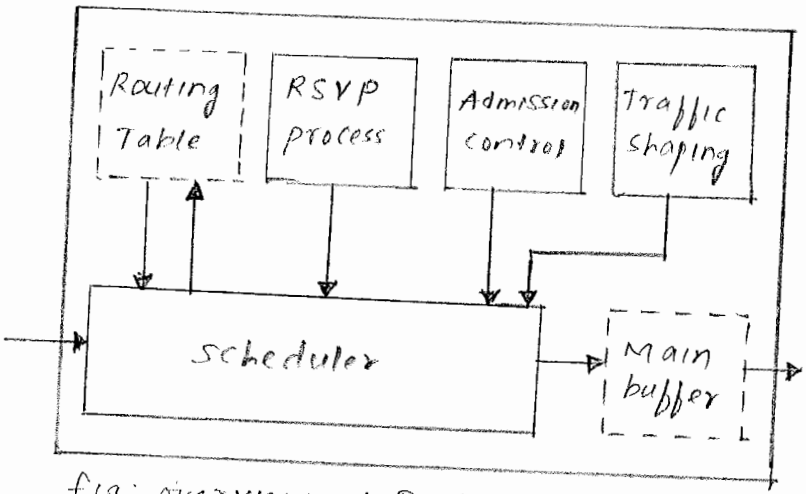
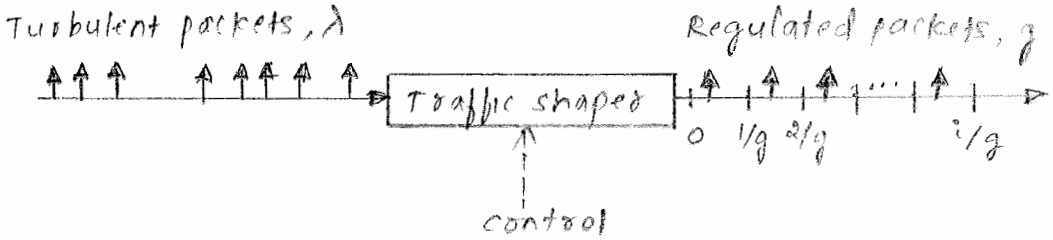


fig: overview of QOS methods in integrated service.

1. Traffic shaping - regulates turbulent traffic
2. Admission control - governs whether the n/w, given inform^n about an applicns flow, can admit or reject the flow
3. Resource allocation - lets n/w users reserve BW on neighbouring routers.
4. Packet scheduling - sets the time table for the transmission of packet flows.

Traffic shaping

* Realistically, spacing blw incoming packets has an irregular pattern, which may cause congestion. The goal of traffic shaping is to control access to available BW to regulate incoming data to avoid congestion and to control the delay incurred by the packets (ref. fig)



Turbulent packets at rate λ & with irregular arrival patterns are regulated in a traffic shaper over equal-sized 1/g intervals.

* Two traffic shaping algorithms
  - leaky Bucket
  - Token Bucket

Leaky Bucket Traffic shaping.

* input: any turbulent incoming traffic.
  output: smooth, regular stream of packets.
* Fig (a) shows how this algorithm works.

* This algo is used for nlws with variable-length packets and also equal sized packet protocols, such as ATM.

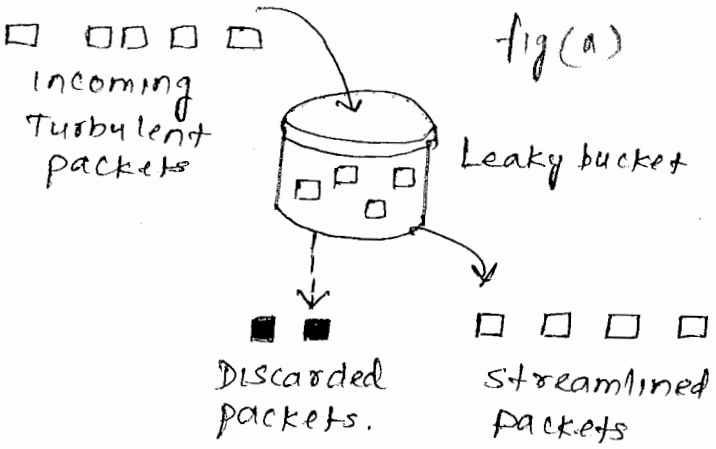


fig (a)

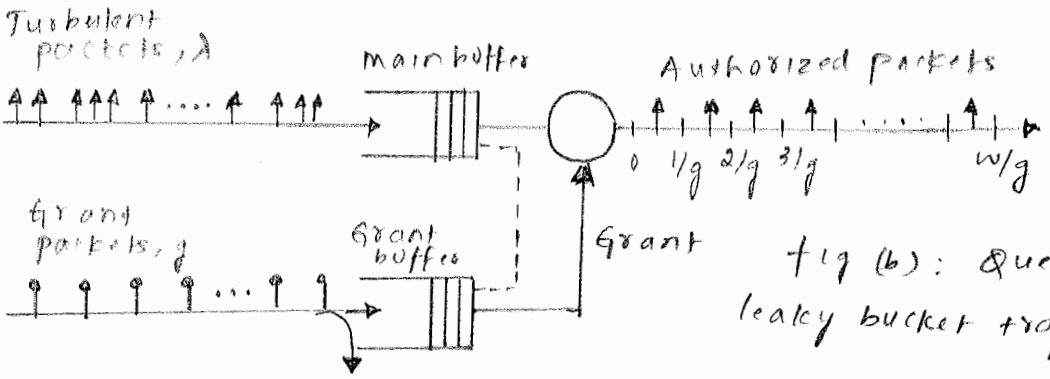


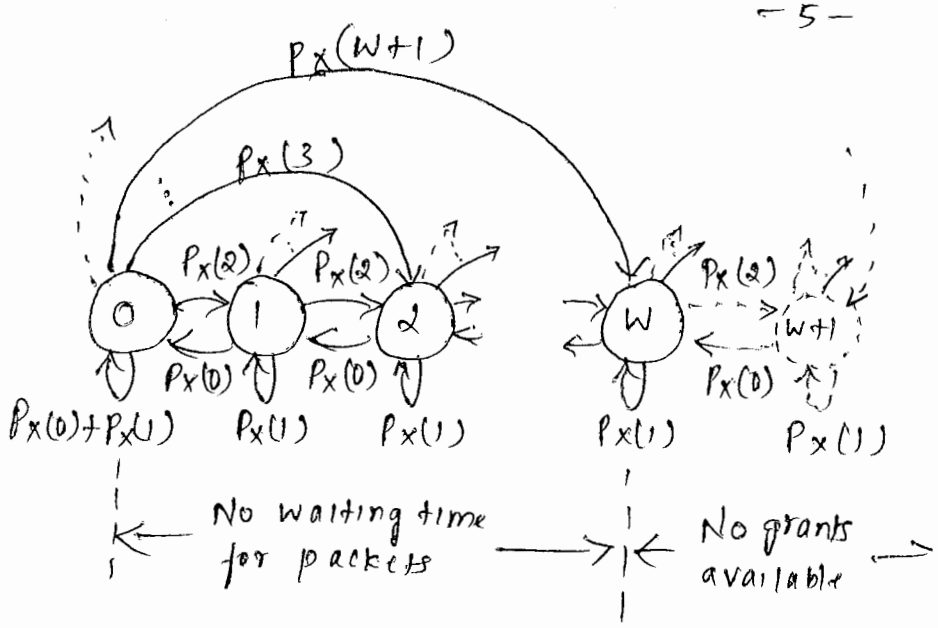
fig (b): Queing model of leaky bucket traffic shaping algo

* The leaky bucket scheme is modeled by two main buffers (as shown in fig b). One buffer forms a queue of incoming packets, & the other one receives authorizations.

* leaky bucket traffic shaper algorithm is summarized below

1. Define for the Algorithm
   λ = rate at which packets with irregular rate arrive at main buffer
   g = rate at which authorization grants arrive at grant buffer
   w = size of grant buffer & can be dynamically adjusted.
2. Every $1/g$ seconds, a grant arrives.
3. over each period of $1/g$ sec, i grants can be assigned to the first i incoming packets where $i \leq w$, & packets exit from the queue one at a time every $1/g$ sec, totaling i/g sec.
4. If more than w packets are in the main buffer, only first w packets are assigned grants at each window time of $1/g$, and the rest remain in the main queue to be examined in the next $1/g$ interval.
5. If no grant is in the grant buffer, packets start to be queued.

* fig shows the markov chain state diagram depicting the activity of grant generation in time.



Delay analysis:

variables -

- $P_x(x)$: probability of x packet arrivals in $1/g$ sec.
- P_i : probability that a grant has arrived or that the markov chain is in state i.
- P_{ji} : transition probability from any state j to a state i on the markov chain.

Transition probability at state 0 has two components

$$P_{00} = P_x(0) + P_x(1).$$

As long as $i \geq 1$, State 0 can be connected to any state $i \leq W$, including the following property of the queuing system in fig.

$$P_{0i} = P_x(i+1) \quad \text{for } i \geq 1$$

the remaining transition probabilities are derived from

$$P_{ji} = \begin{cases} P_x(i-j+1) & \text{for } 1 \leq j \leq i+1 \\ 0 & \text{for } j > i+1 \end{cases}$$

Now the global balance eqns can be formed. We are particularly interested in the probability of any state i denoted by P_i.

The probability that the chain is in state 0, P_0 (implying that no grant has arrived) is the sum of incoming transitions:

$$P_0 = P_x(0) P_1 + [P_x(0) + P_x(1)] P_0$$

For P_i, we can write

$$P_i = P_x(2) P_0 + P_x(1) P_i + P_x(0) P_2$$

Generally

$$P_i = \sum_{j=0}^{i+1} P_x(i-j+1) P_j \quad \text{for } i \geq 1.$$

This can be recursively solved, & we can use little's formula to estimate average waiting period to obtain a grant for a packet, $E(T)$

i.e. $$\text{average waiting period } E(T) = \frac{\sum_{i=w+1}^{\infty} (i-w) P_i}{g}$$

Token Bucket Traffic shaping

* For most applications, traffic flow varies. Hence BW varies with time for these applications.
* Two parameters together can describe the operation of the token bucket traffic shaper
  - token arrival rate, v – normally set to average traffic rate of the source.
  - Bucket depth, b – measure of max amount of traffic that a sender can send in a burst.

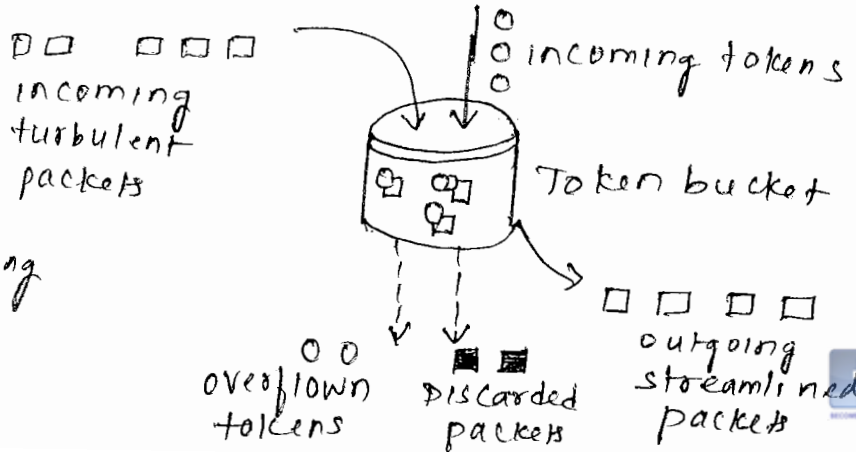


fig: Token bucket Traffic shaping method.

Working:

- Token bucket shaper consists of a buffer (like a water bucket) that accepts fixed size tokens of data generated by a token generator at a constant rate every clock cycle
- packets with any unpredictable rate must pass through this bucket unit.
- Acc to this protocol, a sufficient no. of tokens are attached to each incoming packet (depending on its size) to enter the n/w.
- If the bucket is full of tokens, additional tokens are discarded. If the bucket is empty, incoming packets are delayed (buffered) until a sufficient no. of tokens are generated.
- If the packet size is too big, such that there are not enough tokens to accomodate it, a delay of i/p packets is carried over.

* operates well when no. of tokens is larger than no. of incoming packet sizes.

By changing the clock frequency and varying the token-generation rate, we can adjust the i/p traffic to an expected rate.

Token Bucket algo.	Leaky bucket algo.
* Enforces more flexible o/p pattern at the average rate, no matter how irregular the incoming traffic is.	* Enforces a more rigid pattern.
* greater System complexity.	* No virtual tokens are seen, greatly increasing the speed of operation & enhancing the performance of the

Admission control

* This process decides whether to accept the traffic flow by looking at two factors.
  1. r_s = type of service requested
  2. t_s = required BW information about the flow.

note: for controlled load services, no additional parameters are required. For guaranteed services, the max. amount of delay must also be specified. In any router (or host) capable of admission control, if currently available resources can provide service to the flow without affecting the service to other (already admitted flows), the flow is admitted; else, rejected. Admission control scheme is aided by policing scheme once the flow is admitted, policing make sure that flow confirms to the specified t_s.

Resource Reservation Protocol (RRP)

* used to provide real-time services over connectionless n/w.
* It is soft-state protocol, can handle link failure efficiently
* Required to provide desired QoS for an applicn.

Working.

* In order for a receiver to make a reservation at intermediate routers, the sender initially sends the t_s message, which passes thru' each intermediate router before reaching receiver.
* This way, the receiver becomes aware of the information on the flow & the path & makes a reservation at each router. This msg is sent periodically to maintain the reservation. The router can accept/deny the reservation, based on its available resources.
* Also t_s is periodically refreshed to adapt to link failures. If a link fails, the receiver receives this msg over a different path.

* The receiver can they use this new path to establish a new reservation, consequently the n/w operation continues as usual.

packet scheduling

* once the resources have been reserved, packet scheduling mechanism has to be in place to provide requested QoS.
* packet scheduling involves managing packets in queues to provide the QoS associated with the packet, as shown.
* packet classifier performs on the basis of the header information in the packet.
* packet classifying involves identifying each packet with its reservation & ensuring that the packet is handled correctly.

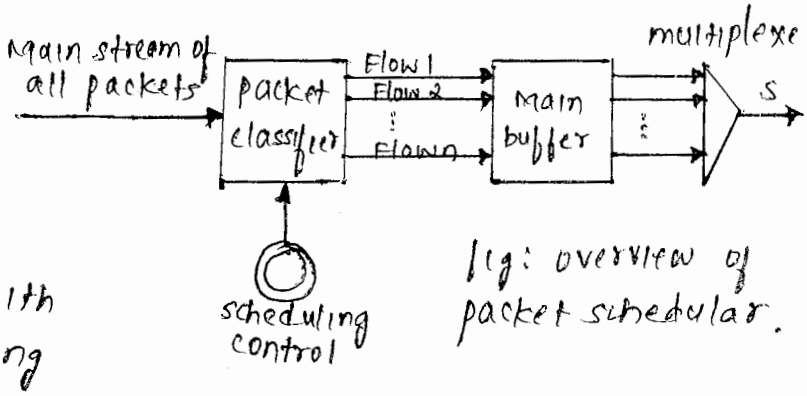


fig: overview of packet schedular.

* some of the scheduling algorithms that can be implemented in the input port processors (IPPs) & output port processors (OPPs) of routers or main host are discussed here.

- First-in First-out schedular.
- Priority Queuing schedular
- Fair Queuing schedular
- Weighted fair queuing schedular.
- Deficit round-robin schedular.
- Earliest deadline first schedular.

First In First Out Scheduler.

* Simple.
* Incoming packets are served in the order in which they arrive.
* Provides no fair treatment to packets.
  - An higher speed user can take up more space in buffer & consume more than his fair share of BW.
* Simple to implement.

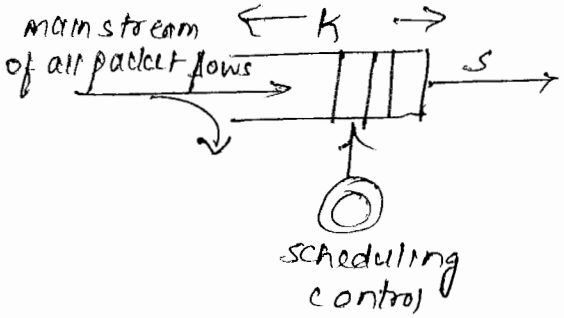


fig: A typical FIFO queueing scheduler.

Delay bound of this scheduler $T_q \leq \frac{K}{S}$

K → max. buffer size
S → outgoing link speed.

Priority Queueing scheduler

* simple + ability to provide service classes.
* With various priority queuing, packets are classified on the priority of service indicated in the packet headers.

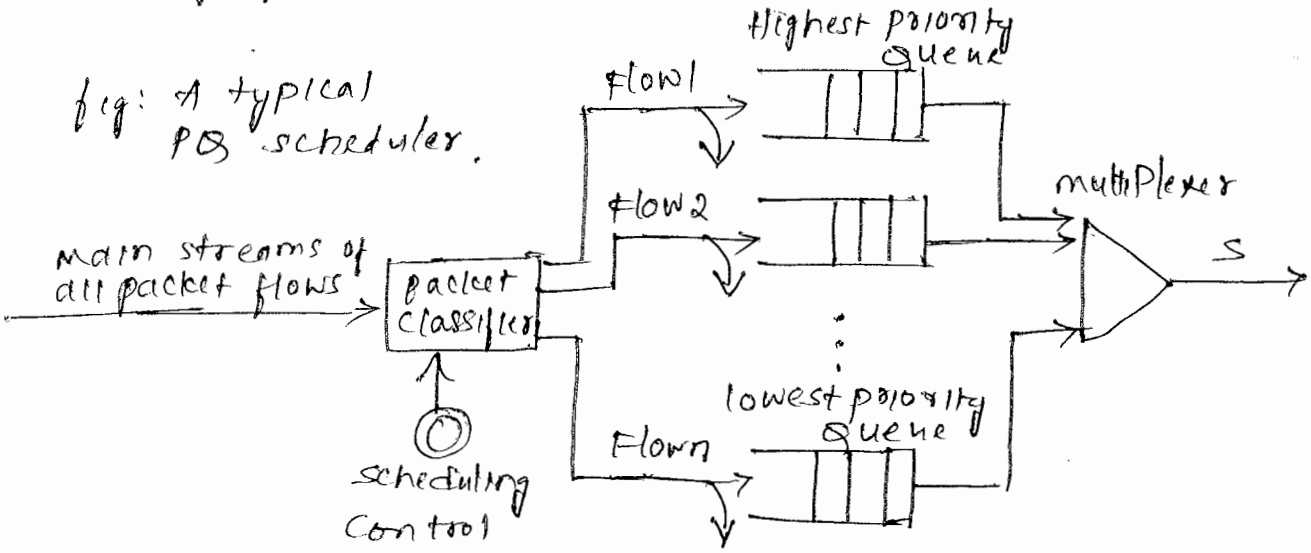


fig: A typical PQ scheduler.

* lower priority queues are serviced only after all packets from higher priority queues are serviced.
* only a queue with highest priority has a delay bound similar to that with FIFO. lower priority queues have a delay bound that includes the delays incurred by higher priority queues.

As a result, queues with lower priorities are subject to bandwidth starvation if the traffic rates for higher-priorities are not controlled.

* priority queuing is either non preemptive or preemptive

note: For a queue with flow i (class i queue), let

$\lambda_i \rightarrow$ arrival rate

$1/\mu_i \rightarrow$ mean service rate

then, mean offered load (utilization) $\rho_i = \frac{\lambda_i}{\mu_i}$

Non preemptive priority queues.

* In non preemptive priority queueing schemes, the service in a lower priority packet cannot be interrupted under any condition.
Everytime an in-service packet terminates from its queue, the waiting packet from the highest priority queue enters service.

* let $E[T_i]$ be the mean waiting time for a packet in flow i queue. This total queueing delay has following 3 components.

1. W_x - mean waiting time for any class i packet until an inservice class j packet among n classes terminates.
2. $E[T_{q,i}]_2$ - mean time until a packet from a class i or lower (higher-priority) waiting ahead is serviced.
3. $E[T_{q,i}]_3$ - mean time owing to higher priority packets arriving while the current packet is waiting & being serviced before the current packet.

Hence,

$$E[T_{q,i}] = W_x + E[T_{q,i}]_2 + E[T_{q,i}]_3 \quad \text{---(1)}$$

$$W_x = \sum_{j=1}^{n} \rho_j r_j$$

r_j - mean residual service time.

using little's formula, we can derive $E[T_{q,i}]_2$ & $E[T_{q,i}]_3$

$$E[T_{q,i}]_2 = \sum_{j=1}^{i} \rho_j E[T_{q,j}]$$

$$E[T_{q,i}]_3 = E[T_{q,i}] \sum_{j=1}^{i-1} \rho_j$$

where $E[T_{q,i}]$ - mean no. of waiting packets for class i packets

$E[T_{q,j}]$ - mean no. of waiting packets for class j packets.

$\therefore$ eq (1) $\Rightarrow$

$$E[T_{q,i}] = \sum_{j=1}^{n} \rho_j r_j + \sum_{j=1}^{i} \rho_j E[T_{q,j}] + E[T_{q,i}] \sum_{j=1}^{i-1} \rho_j$$

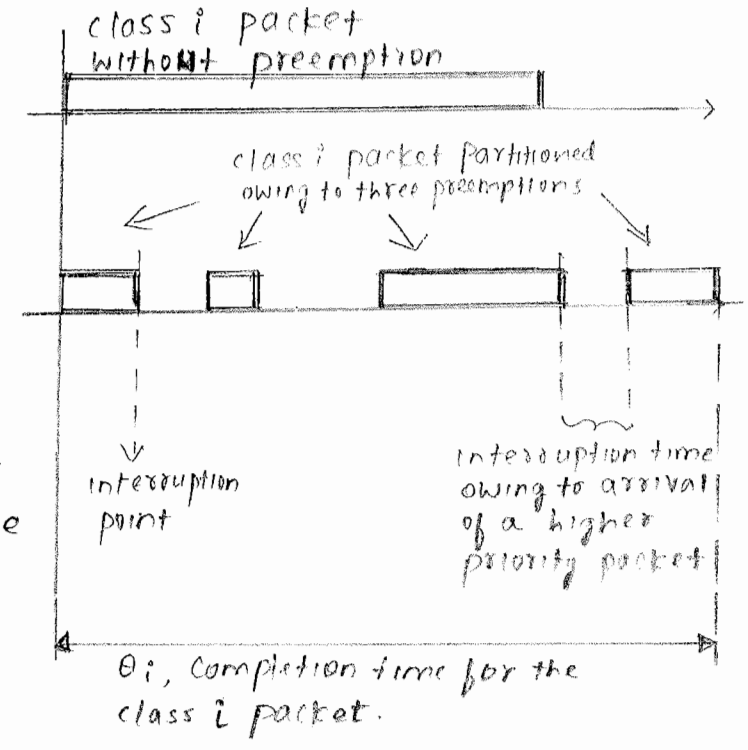
$\therefore$ Total system delay for a packet passing queue i & the server (multiplexer)

$$\boxed{E[T_i] = E[T_{q,i}] + \frac{1}{\mu_i}}$$

Preemptive priority Queues.

* Here, service in a lower priority packet can be interrupted by an incoming higher-priority packet.
* let $E[T_i]$ be the mean waiting time for a packet in flow i (class i) queue.
* Fig. below shows an ex. in which a class i packet has been interrupted three times by higher priority packets.

* This total queueing delay has four components. The first three are identical to those for the non preemptive case. fourth is the θ_i - total mean completion time for the current class i packet when it is preempted in the server by higher priority packets (classes i-1 to 1)



Total system delay for a packet passing queue i & the server

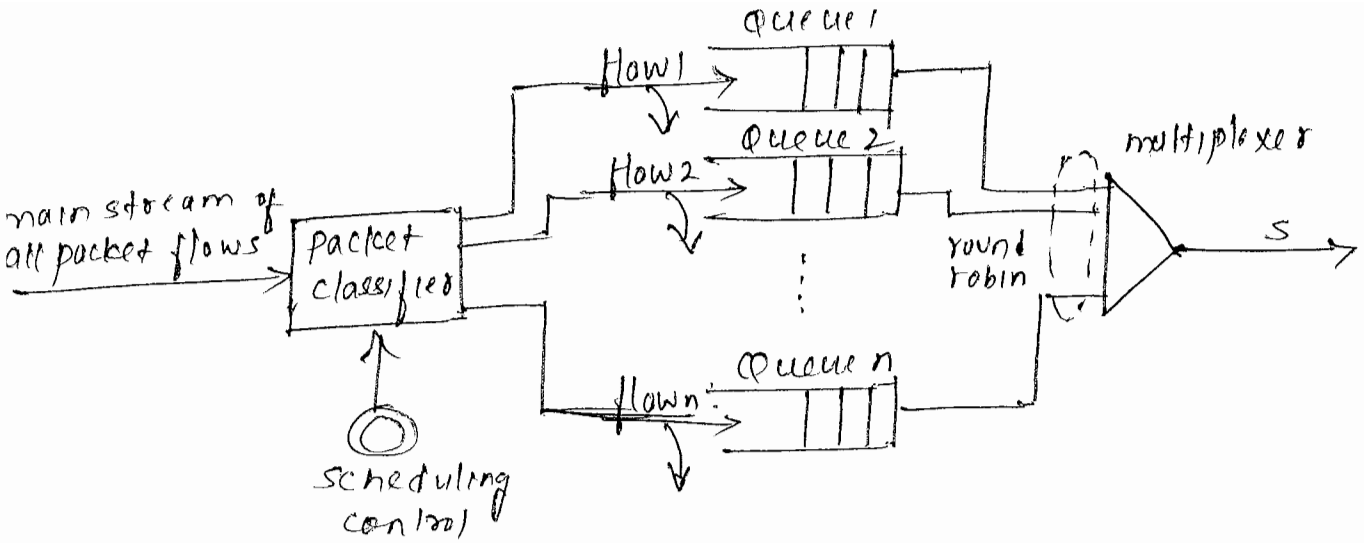
$$E[T_i] = E[T_{q,i}] + \left(\frac{1}{\mu_i} + \theta_i\right)$$

$$\theta_i = \frac{1}{\mu_i\left(1 - \sum_{j=1}^{i-1} \rho_j\right)}$$

$$\therefore \boxed{E[T_i] = E[T_{q,i}] + \frac{1}{\mu_i} + \frac{1}{\mu_i\left(1 - \sum_{j=1}^{i-1} \rho_j\right)}}$$

Fair queueing scheduler.

* Fig below shows fair queuing scheduler.
* Each flow i is guaranteed a minimum ~~share~~ fair share of s/n bits per second on the o/p.
  where s - transmission BW
  n - no. of flows (no. of queues)
* Eliminates the process of packet priority sorting
  $\Rightarrow$ performance & speed of scheduler improved.
* In practise, since all i/ps may not necessarily have packets at the same time the individual queue's final share will be higher than s/n.



* Assume that a_j - arriving time of packet j of a flow. let s_j & f_j be the start & ending times respectively for transmission of packet.

Then

Virtual clock count needed to transmit the packet } $c_j = f_j - s_j$

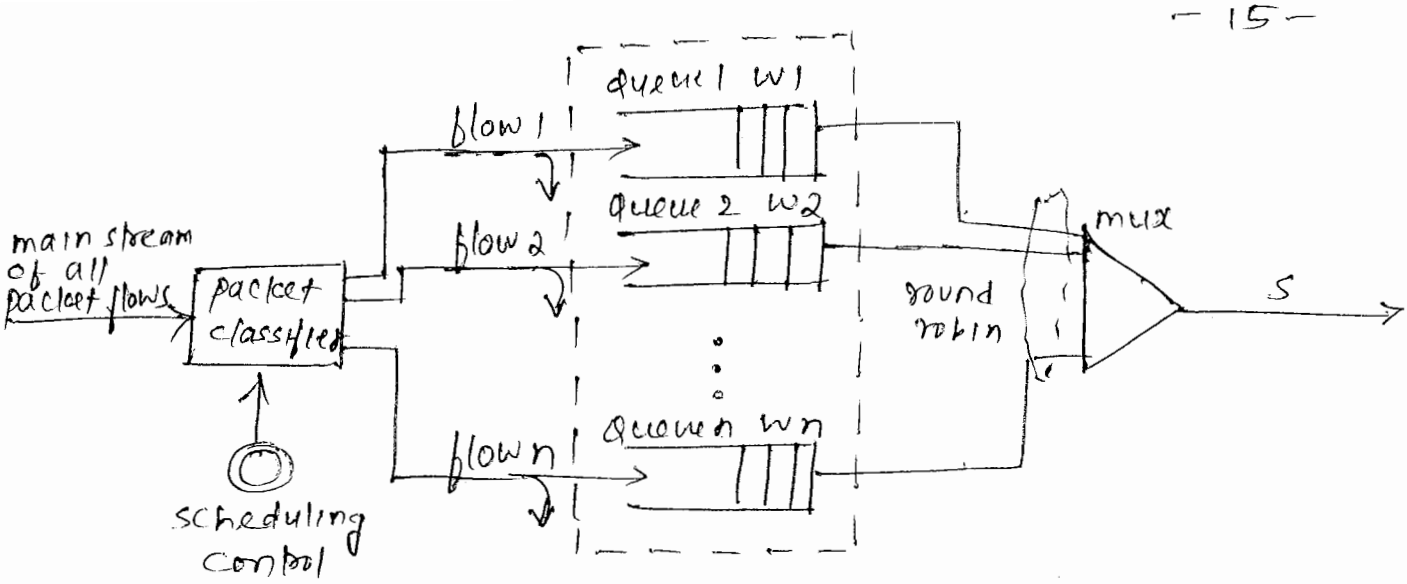
$$c_j = f_j - \max(f_{j-1}, a_j)$$

Weighted-fair-Queueing scheduler.

* improved.
* For n queue system, queue $i \in \{1 \ldots n\}$ is assigned a weight w_i.
* Outgoing link capacity s is shared among the flows w.r.t their allocated weights. thus, each flow i is guaranteed to have a servica rate of at least

$$r_i = \frac{w_i}{\left(\sum_{j=1}^{n} w_j\right)} s \text{ bits per second.}$$

* WFQ provides effective soln for servicing real time & non real time packets, owing to its fairness and ability to satisfy real time constraints
* fig below shows overview of weighted fair queueing scheduler.



* Any unused BW from inactive queues can be distributed among the active queues w.r.t their allocated weights, again with the same fair share of

$$r_i = \left(\frac{w_i}{\sum_{j \in b(t)} w_j} \right) S$$

$b(t)$ - set of active queues at any time t.

* In this scheme, delay bound is independent of max no. of connections, n. ∴ WFQ is one of the best queueing schemes for providing tight delay bounds.
* Disadv - Implementation is complex.
* version of this scheduler, so called weighted round robin (WRR) scheduler was proposed for Asynchronous transfer mode. Here each flow is served in a round robin fashion wrt weight assigned for each flow i, without considering packet length.

Deficit Round-Robin scheduler

* Here, each flow i is allocated b_i bits in each round of service, and bm is defined as ~~maximum~~ minimum bit-allocation value among all flows.
∴ we have, $bm = \min\{b_i\}$

* In every cycle time, active flows are placed in an active list and serviced in round-robin order. If a packet cannot be completely serviced in a round of service without exceeding b_i, the packet is kept active until the next service round, and the unused portion of b_i is added to the next round.

Disadv
- lacks reasonable delay bound
- delay bound for a flow with a small share of BW can be very large.

Adv.
- Throughput.

Earliest deadline first scheduler.

* EDF scheduler computes the departure deadline for incoming packets and forms a sorted deadline list of packets to ensure the required transmission rate and maximum delay guarantees.
* Deadline for a packet can be defined as

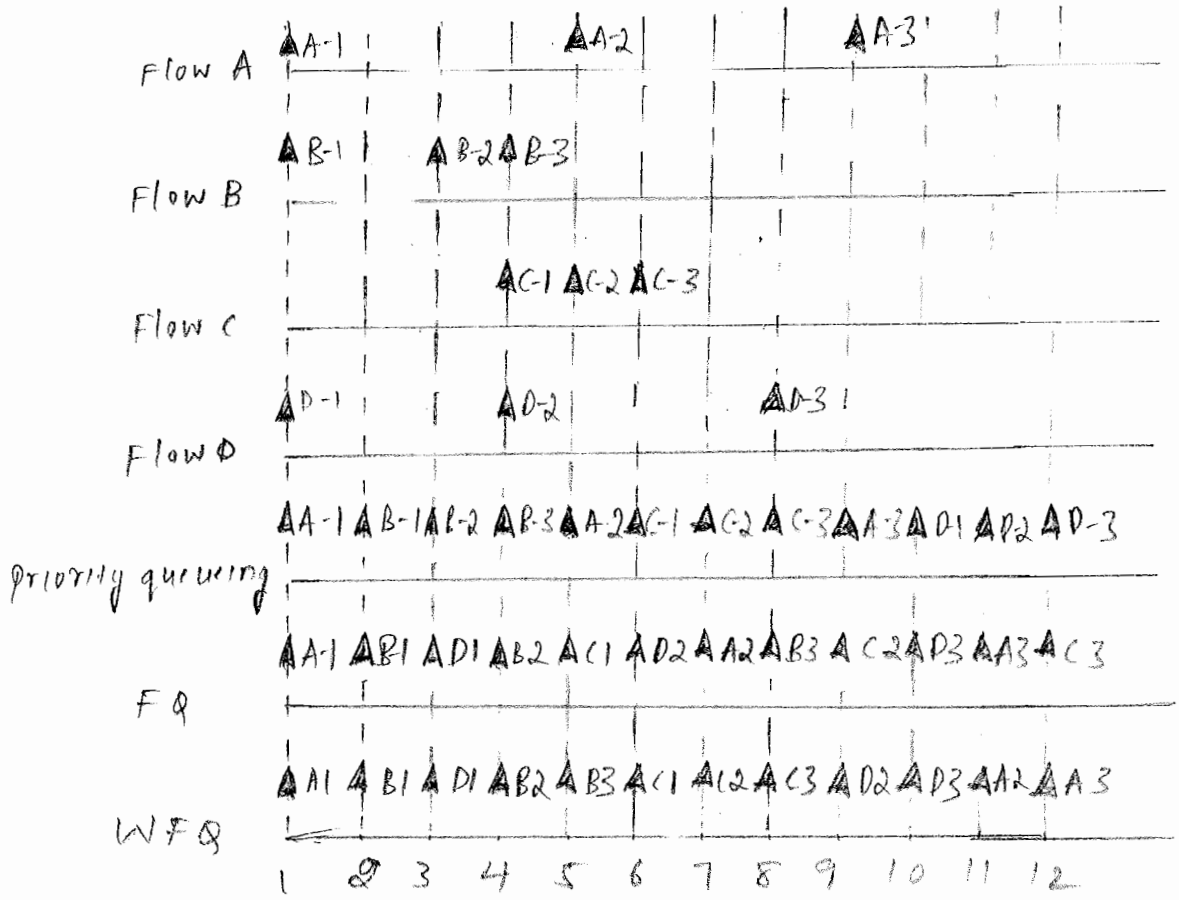
$$D = t_a + T_s.$$

t_a - expected arrival time of a packet at the server.
T_s - delay guarantee of the server associated with the queue that the packet arrives from.

* packet with minimum deadline is served first.

problem:
Given the set of four flows - A, B, C & D to be processed at one of the o/p of a router. Each packet's arrival time & label are indicated. Compare three schedulers - priority queueing, fair queueing, WFQ.

With priority queueing - priorities decrease from line A to D
With FQ - If arrival times of two packets are same, select smaller flow no.
using WFQ - capacity of A, B, C & D are 20%, 10%, 40% and 20% of the o/p capacity respectively

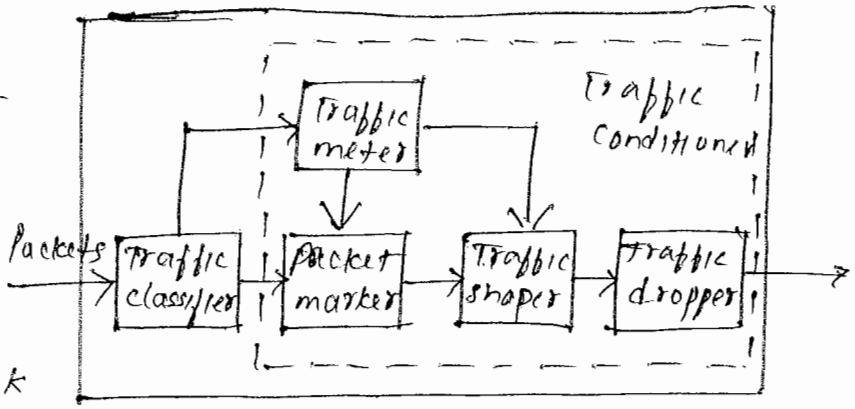


DIFFENTIATED SERVICES QOS

~~* provide QoS support to broad class of applications~~

* Differentiated services (Diffserv) approach is based on providing QoS support to a broad class of applications
* provides a simpler & more scalable QoS.
* D.S minimizes the amount of storage needed in a router by processing traffic flows in an aggregate manner moving all the complex procedures from the core to the edge of the n/w.
* Fig shows overview of Diffserv operation. A traffic conditioner is one of the main features of a Diffserv node to protect the Diffserv domain. It includes four major components
  - meter
  - marker

- meter measures the traffic to make sure that packets do not exceed their traffic profiles
- marker marks or unmarks packets in order to keep track of their situations in the DS node.



- shaper delays any packet that is not compliant with the traffic profile.
- Dropper discards any packet that violates its traffic profile

note:

* In order to allocate and control the available bandwidth within the DS domain (to ensure requested QOS), a Bandwidth broker is needed to manage the traffic. It operates in its own DS domain & maintain contact with other BW brokers at neighbouring domains.
* In order to process traffic flows in aggregate manner, a packet must go through a service-level agreement (SLA) that includes a traffic-conditioning agreement (TCA) An SLA indicates the type of forwarding service, & a TCA presents all the detailed parameters that a customer receives.

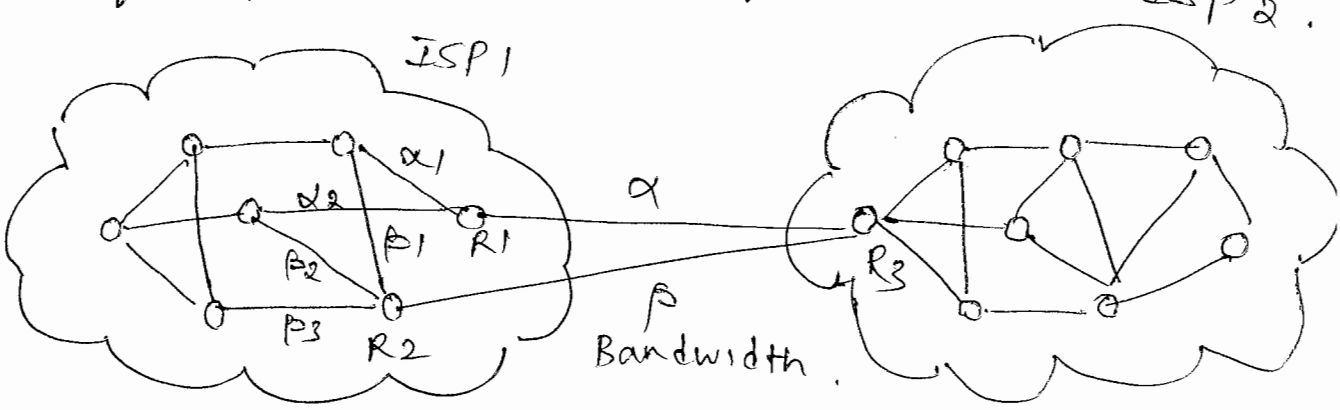
Per Hop Behaviour (PHB)

* we define two PHBs
  - expedited forwarding, provides low-loss, low-latency, low-jitter, ensured-BW, & end-to-end services.
  - Assured (or ensured) forwarding, does not provide low-latency & low-jitter application. The ensured forwarding PHB group can be classified into three service types: good, average, & poor.

RESOURCE ALLOCATION

* Buffer management is required to provide rate guarantees for a network.

need for n/w resource allocation.



* Congestion occurs ~~when out~~ at router R1 when outgoing link capacity (α) is less than the sum of incoming ($\alpha_1 + \alpha_2$) traffic. Similarly congestion occurs at R2 when β is less than $\beta_1 + \beta_2 + \beta_3$.
* Another important situation is that router R3 located in different domain must be able to handle a great volume of traffic $\alpha + \beta$.

classification of Resource allocation schemes

1. Router Based versus Host Based

- Router based
  * Here router sets up required resources
  * Routers have primary responsibility for congestion control.
  * A router selectively forwards packets or drops them, if required, to manage the allocation of existing resources.
- Host Based
  * Host sets up required resources.
  * End hosts have primary responsibility for congestion ~~cong~~ control
  * Hosts observe the traffic conditions, such as throughput, delay, & packet losses, & adjusts the rate at which they generate & send packets accordingly.

2. Fixed versus Adaptive.

- Fixed reservation scheme.
  * Here end hosts request resources at the router level before a flow begins.
  * The router then allocates enough resources (BW, buffers etc) for the flow based on its available resources. A router also ensures that new reservations does not affect the QoS provided to the existing reservations.
  * This is router based, as router is responsible for allocating sufficient resources for the flow.

- Adaptive reservation scheme
  * Here, end hosts send packets without reserving resources at the router and then adjust their sending rates, based on observable traffic conditions or the response from the router.
  * The router may also send messages to end hosts to slow down their sending rate.
  * This can be either router based or host based. If end hosts adjust rates of transmitted packets based on observable traffic conditions, the scheme is typically host based.

3. Window Based versus Rate based

- Window Based resource allocation.
  * Here a receiver chooses a window size based on buffer space available to the receiver.
  * The receiver then sends this window size to the sender. The sender transmits packets in accordance with the window size advertised by the receiver.

- Rate Based resource allocation.
  * Here the receiver specifies a maximum rate (bits per second) it can handle. A sender sends traffic in compliance with the rate advertised by the receiver.

CHAPTER 6B: VPNs, TUNNELING and OVERLAY NETWORKS.

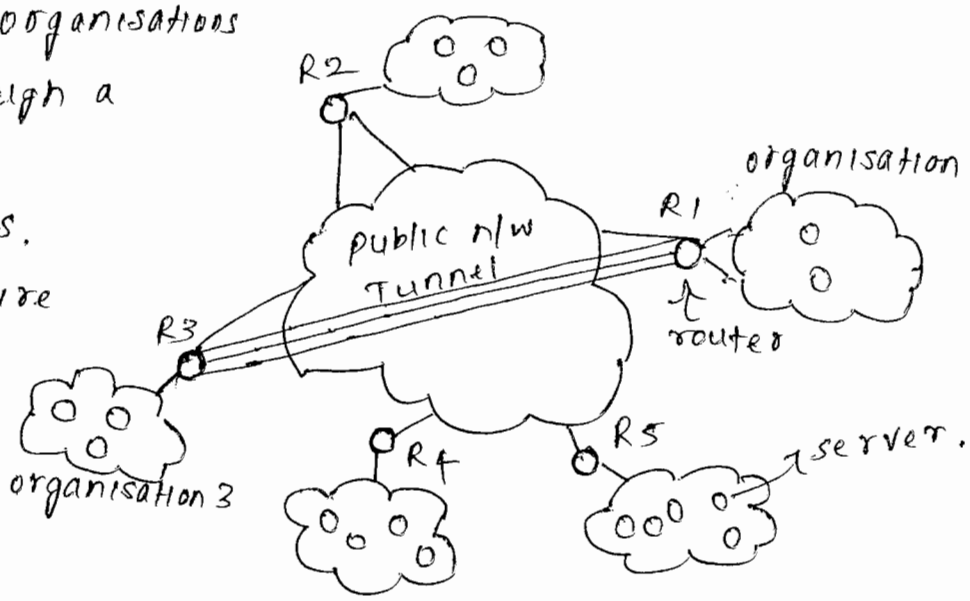
* This chapter deals with how n/ws can be overlaid or tunneled.
* In networking, tunneling is the encapsulation of a packet from one protocol to another one at the same or higher level. (VPNs & MPLS are discussed in this chapter)

VIRTUAL PRIVATE NETWORKS (VPNs)

* VPN is an networking ~~interface~~ infrastructure where by a private n/w makes use of the public n/ws.

(or) VPN is a data n/w having connections that make use of public networking facilities.

* The part of public n/w (VPN) is set up "virtually" by a private sector entity to provide public networking services to small entities.
* A VPN maintains privacy by using tunneling protocols and security procedures.
* Fig shows two organisations connected through a tunnel using public facilities. Such a structure gives both private organisations the same capabilities they have on their own n/ws but at much lower cost.



* Advantages of creating an VPN. It provides
  - Extended geographical communication
  - Reduced operational cost
  - Enhanced organisational management
  - Enhanced n/w management with simplified LANs.
  - improved productivity & globalization.

* Since user has no control over wires and routers, one of the issues with the internet is still its lack of security, especially when a tunnel is exposed to the public. (VPN security is discussed later)

* Acc to method of tunneling, there are two different types of VPNs
  - Remote access VPN
  - site-to-site VPN

Remote Access VPN.

* Remote Access VPN is a user-to-LAN connection that an organisation uses to connect its users to a private n/w from various remote locations.

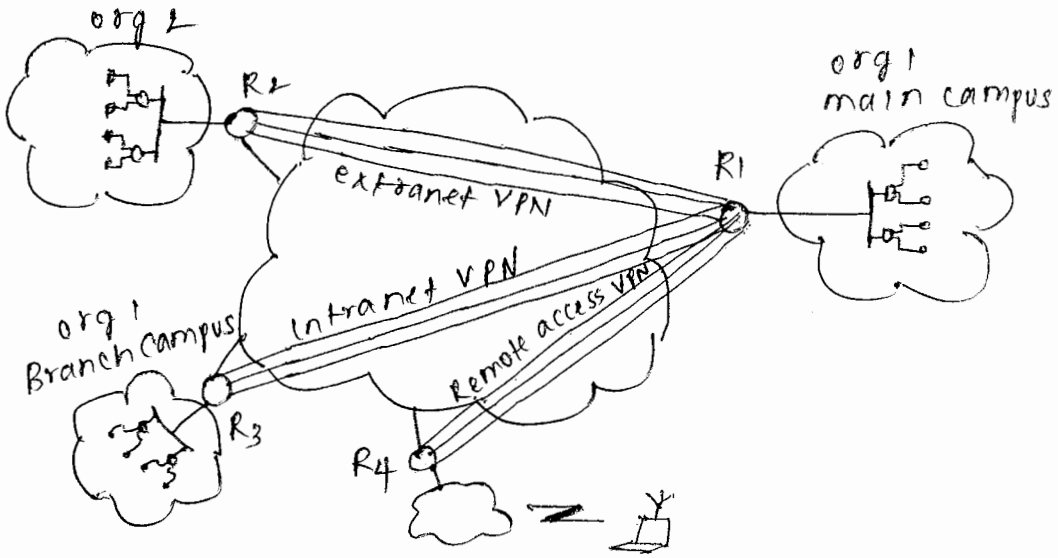
* Large Remote access VPNs are normally outsourced to an internet service provider to set up an n/w-access server. other users working off campus, can then reach the n/w access server and use the VPN software to access the corporate n/w.
  ie Remote access VPNs allows encrypted connections b/w an organisation's private n/w and remote users through a third party service provider.

* Tunneling in a remote access VPN uses mainly the PPP. other types of protocols associated with PPP are
  - L2F (layer 2 forwarding) protocol uses authentication scheme supported by PPP
  - PPTP (point to point tunneling protocol) supports 40 bit & 128 bit encryption & uses authentication scheme supported by PPP
  - L2TP (layer 2 Tunneling protocol) combines features of both L2F and PPTP.

Site-to-site VPN

* Here, an organisation can connect multiple fixed sites over a public n/w (using effective security technique)
* site-to-site VPNs can be classified as either
  - Intranet: VPNs connect an organisation's remote-site LANs into a single private n/w, or
  - Extranet: VPNs allow two organisations to work in a shared environment through a tunnel built to connect their LANs.

* Fig shows three types of VPNs to and from an head quarter organisation.



* In a site-to-site VPN, generic routing encapsulation (GRE) is normally the encapsulating protocol

Tunneling and point-to-point protocol (PPP)

* A tunnel is an connection that forms a virtual network on top of a physical network.
* Tunneling is the process of encapsulating packets and sending them over the public n/w.

(Employees who are located o/s an org's main building can use point-to-point connections to create tunnels through the internet.

Since tunneling connections normally run over the internet, they need to be secure.

Tunnel is inexpensive connection, since it uses internet

* Beside internet protocols, tunneling requires two other types of protocols.

- carrier protocols: through which information travels over the public networks.
- Encapsulating protocols: through which data is wrapped, encapsulated, & secured.

* one amazing implications of VPNs is that packets that use a protocol not supported on the internet (eg NetBeui) can be placed inside IP packet & sent safely over the internet

* Refer fig, Assume that two LANs want to use their own customised networking protocols, denoted by x, using connectionless datagram IP services.

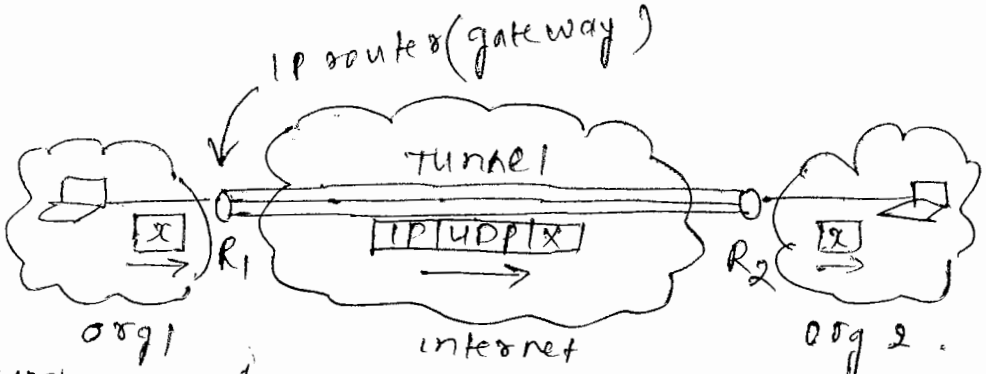


fig: customized protocol packet tunneling through the internet.

* x-type packets can not run over the internet directly. IP gateway R1 encapsulates them in transport layer UDP datagrams, & transmits over the internet to R2. when R2 receives encapsulated x packets, it decapsulates & feeds them into org 2. This connection (tunnel made thru' internet) resembles a direct physical link b/w two LANs.

Point-to-point protocol (PPP)

* Tunnel can also be defined as an encapsulating protocol for protocols at the lower layers.
* Tunneling protocols such as PPP or PPTP are encapsulating protocols that allow an org. to establish secure connections from one point to another while using public resources.
* A PPP is an serial connection b/w an user & an ISP. connection

Security in VPNs:

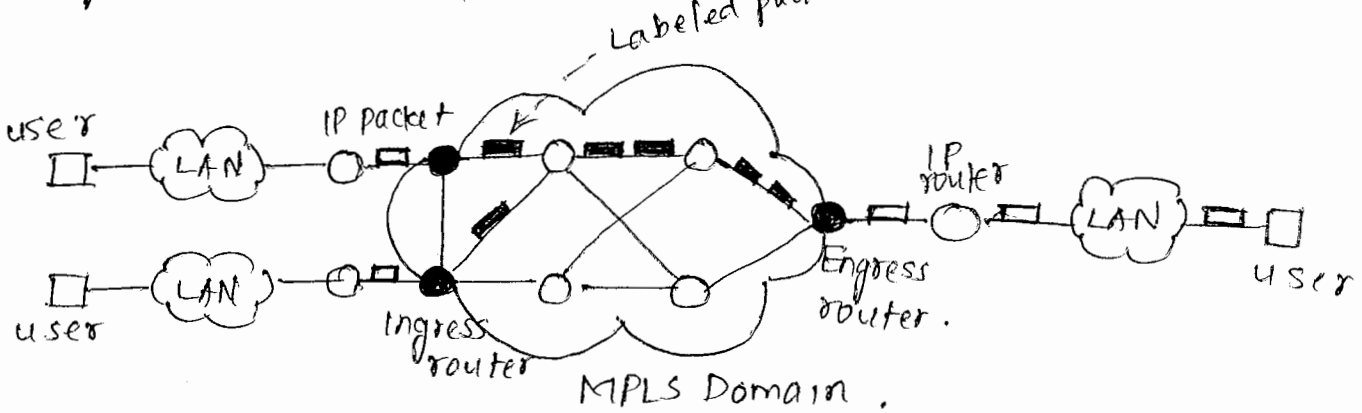
* A well protected VPN uses firewall, encryption systems, IPsec features, and an authentication server.
* Firewall provides an effective barrier b/w a private n/w & internet.
  The authentication server performs
  - authentication (who the sender is)
  - authorization (what is allowed to do)
  - ~~oth~~ accounting (what it actually does).

MULTI PROTOCOL LABEL SWITCHING (MPLS)

* MPLS are good example for VPNs.
* MPLS improves the overall performance of traditional IP routing, especially for the establishment of more effective VPNs.
  In MPLS, multiple labels can be combined in a packet to form a header used by an LSR (label switch routers) for efficient tunneling.

MPLS operation.

* Fig shows an MPLS n/w.

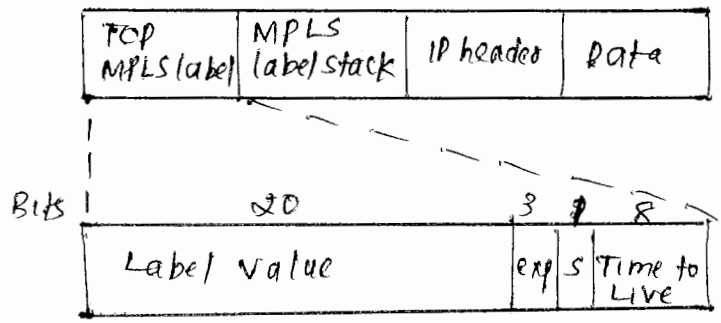


* MPLS is based on assignment of labels to packets. This make label-swapping scheme perform its routing process much more efficiently.
* An MPLS n/w consists of nodes called Label switch routers (LSR). An LSR switches labeled ~~packet~~ packets according to

* An LSR has two distinct functional components.
  - control component: uses routing protocols such as OSPF and the border gateway protocol (BGP)
  - farwording component: directs the packet from the input interface to o/p interface thru' the switching fabric.

MPSL packet format

* Fig shows MPLS header encapsulation for an IP packet.



- Label value: 20 bit field label & is significant only locally
- Exp: is a 3-bit field reserved for future experimental use.
- S: is set to 1 for the oldest entry in the stack and to 0 for all other entries.
- TTL: 8 bit field used to encope a hop-count value to prevent packets from looping forever in the n/w.

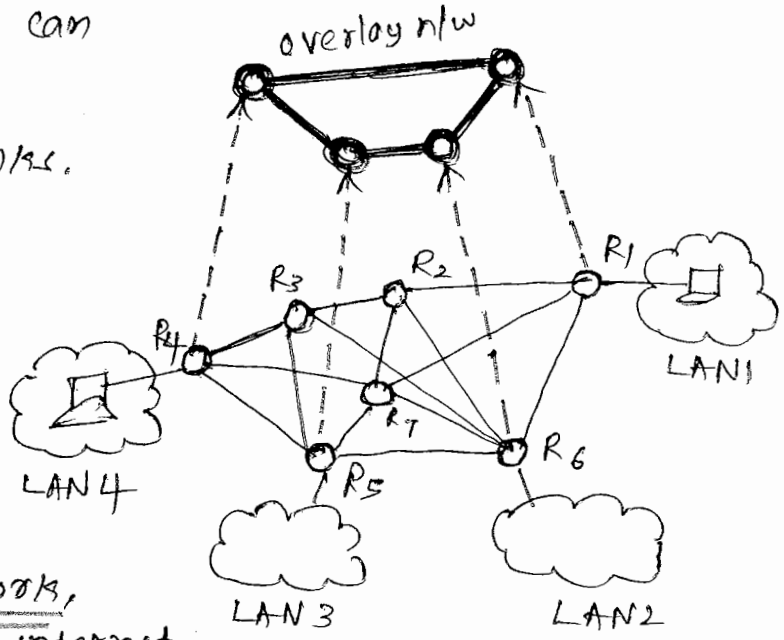
* MPLS uses label stacking to become capable of multilevel hierarchical routing.

Overlay Networks.

* An overlay network is an application specific computer network built on top of another network. In other words, Overlay networks creates a virtual topology on top of the physical topology.

* Figure shows an overlay network configured over a wide area network.

Nodes in overlay n/w can be thought of as being connected by logical links. In fig for ex, routers R_4, R_5, R_6, and R_1 are participating in creating an overlay n/w



* Ex for overlay n/w is peer to peer network, which runs on top of internet.

* overlay n/w have no control over how packets are routed in the underlying network b/w a pair of overlay source/Des. nodes. However, these n/w can control a sequence of overlay nodes through a message-passing function before reaching the destination.

why overlay n/w are needed in communication system?

- They permits routing messages to destinations when the IP address is not known in advance.
- Sometimes, they are proposed as a method to improve internet routing to achieve higher-quality streaming media.
- An overlay n/w can be deployed on end hosts ~~routing~~ running the overlay protocol software, without co-operation

* overlay n/w are self organised.

- when a node fails, the overlay n/w algo provide solutions that let the network recover and recreate an appropriate n/w structure.

Left Topics.

- Routing in MPLS Domains, Traffic engineering

UNIT 7

COMPRESSION OF DIGITAL VOICE AND VIDEO, VOIP, MULTIMEDIA NETWORKING

Syllabus

Chapter 7a: COMPRESSION OF DIGITAL VOICE AND VIDEO

* Overview of data compression
* Digital voice and compression
* Still images and JPEG compression
* Moving images and MPEG compression
* Limits of compression with loss
* Compression methods without loss
* Case study: FAX compression for transmission.

Chapter 7b: VoIP and Multimedia Networking

* Overview of IP telephony
* VoIP signalling protocols
* Real-time media Transport protocols
* Distributed Multimedia Networking
* Stream control Transmission protocol (SCTP)

- 7 Hours.

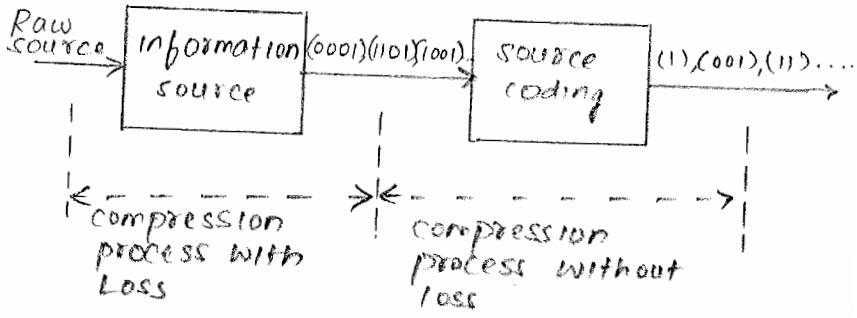
Ashok Kumar K
VIVEKANANDA INSTITUTE OF TECHNOLOGY

Chapter 7A:

COMPRESSION OF DIGITAL VOICE AND VIDEO

OVERVIEW OF DATA COMPRESSION

* Benefits of data compression.
  1.) less transmission power is required
  2.) less communication bandwidth is required
  3.) system efficiency is increased.
* Trade-offs with data compression → encoding and decoding processes of data compression increase the cost, complexity, and delay of data ~~compression~~ transmission.
* Fig shows overview of information process and compression in multimedia n/ws.
* Any type of "source" data is converted to digital form in a long information source process. outcome is the generation of digital words.

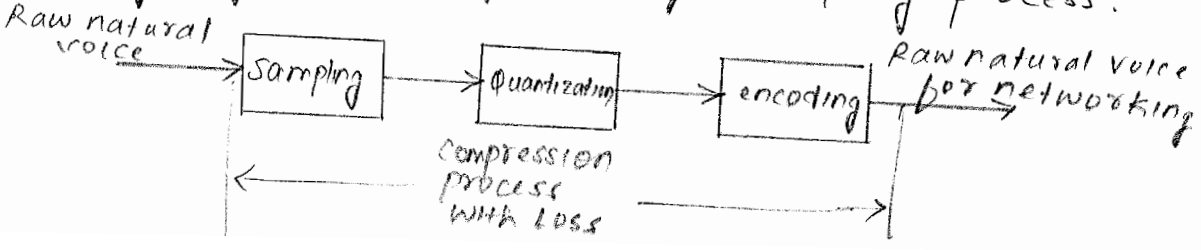


* Words are encoded in the source coding system to result in a compressed form of the data.

DIGITAL VOICE AND COMPRESSION

Signal sampling

* Fig below shows the process of digitizing the signal.
* Analog signal first go through sampling process.



* Sampling resembles an information-compprocess process with loss, because acquiring samples from an analog signal & eliminating the unsampled portions of the signal may result in some permanent loss of information.

* Sampling techniques are of several types:

1.) Pulse Amplitude Modulation (PAM) → translates sampled values to pulses with corresponding amplitudes.

2.) Pulse Width Modulation (PWM) → translates sampled values to pulses with corresponding widths.

3.) Pulse position Modulation (PPM) → translates sampled values to identical pulses but corresponding positions with to sampling points.

* Sampling rate in any of these schemes obeys the nyquist theorem, acc. to which atleast two samples on all components of the spectrum are needed in order to reconstruct a spectrum

$$f_s \geq 2 f_H$$

where

f_s → sampling rate

f_H → highest frequency component of a signal.

Quantization and Distortion.

* In practice, sampled values (real) are rounded off to available quantized ~~values~~ levels. This produces Distortion.

Distortion measure

let $x(t)$ → signal

$\hat{x}(t)$ → reproduced version (after quantization)

x_i → source sample

$\hat{x}_i$ → corresponding quantized value

Then, distortion measure) $\boxed{d(x, \hat{x}) = (x - \hat{x})^2}$ widely known as squared-error distortion

* Collection of n-samples forms a random process,

$$X_n = \{x_1, x_2, x_3 \cdots, x_n\}$$

then, reconstructed signal at the receiver can also be viewed as a random process

$$\hat{X}_n = \{\hat{x}_1, \hat{x}_2 \cdots \hat{x}_n\}$$

The distortion b/w these two sequences is the average b/w their components

$$d(X_n, \hat{X}_n) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

Note that $d(X_n, \hat{X}_n)$ itself is a random variable, since it takes on random numbers.

Thus, Total Distortion D is the expected value of $d(X_n, \hat{X}_n)$

$$D = E[d(X_n, \hat{X}_n)] = E\left[\frac{1}{n} \sum_{i=0}^{n} d(x_i, \hat{x}_i)\right]$$

$$= \frac{1}{n} E\left[d(x_1, \hat{x}_1) + d(x_2, \hat{x}_2) + \cdots + d(x_n, \hat{x}_n)\right]$$

If all samples are expected to have approx. the same distortion denoted by $d(x, \hat{x})$,

then

$$D = \frac{1}{n}\left(nE[d(x, \hat{x})]\right)$$

$D = E(x - \hat{x})^2 \rightarrow$ from squared error distortion eqn.

$$= \int_{-\infty}^{\infty} (x - \hat{x})^2 f_X(x)\, dx \quad \text{(By using def}^n \text{ of expected value)}$$

$$\boxed{D = \sum_{i=1}^{N} \int_{R_i} (x - \hat{x})^2 f_X(x)\, dx}$$

R -> set of real nos $R_1, R_2 \cdots R_k \cdots R_N$.

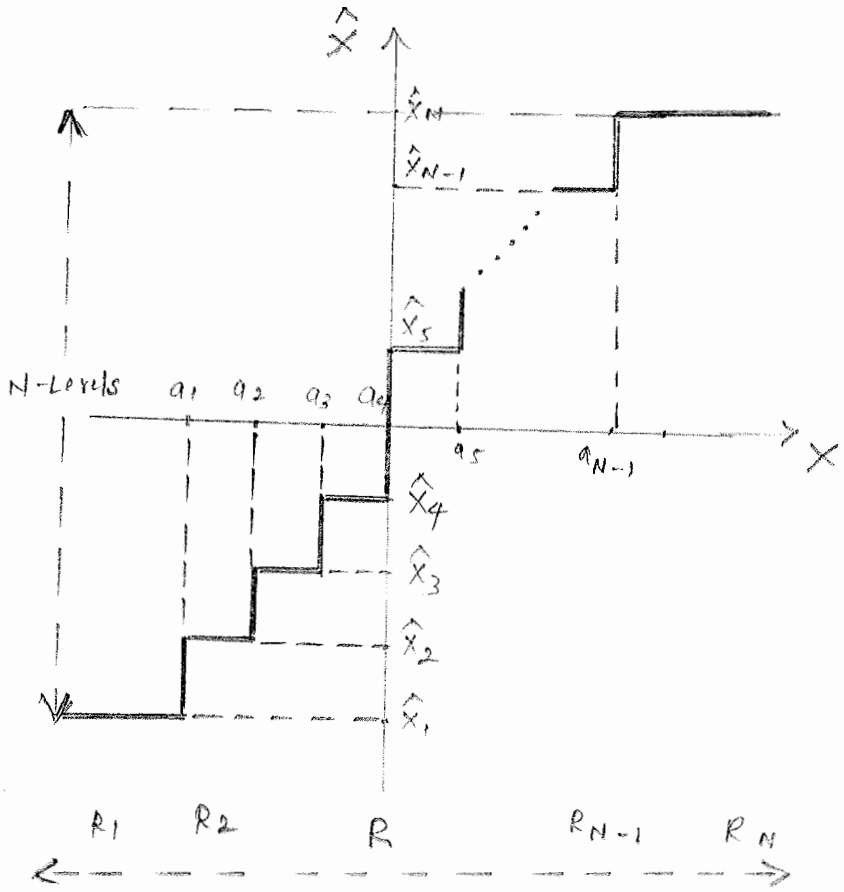
↳ min. no. of bits required to reproduce a source & guarantee that the distortion be less than a certain distortion bound D_b.

usually, $\boxed{D_b \leq D}$

* Apparently, $R = \log_2 N$ bits are required to encode N quantised levels.

* Fig. shows a model of N-level Quantization.

* For the subsets
$R_1 = \{-\infty, a_1\}$,
$R_2 = \{a_1, a_2\}$
$\vdots$
$R_N = \{a_{N-1}, \infty)$
the quantized values are
$\hat{x}_1, \hat{x}_2 \ldots \hat{x}_n$ respectively



note: optimal Quantizers.

* let $\Delta \rightarrow$ length of each region $= a_{i+1} - a_i$

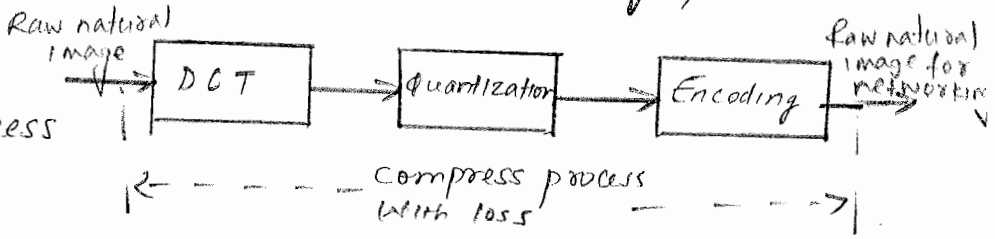
then, $$D = \int_{-\infty}^{a_1} (x - \hat{x})^2 f_x(x)\,dx + \sum_{i=1}^{N-2} \int_{a_1 + (i-1)\Delta}^{a_1 + i\Delta} (x - \hat{x}_{i+1})^2 f_x(x)\,dx + \int_{a_1 + (N-2)\Delta}^{\infty} (x - \hat{x}_N)^2 f_x(x)\,dx .$$

for D to be optimal, we must have

$$\frac{\delta D}{\delta a_1} = 0 \quad \frac{\delta D}{\delta \Delta} = 0 \quad \text{and} \quad \frac{\delta D}{\delta \hat{x}_1} = 0, \frac{\delta D}{\delta \hat{x}_2} = 0, \ldots, \frac{\delta D}{\delta \hat{x}_n} = 0 .$$

STILL IMAGES AND JPEG COMPRESSION

* The Joint photographic experts Group (JPEG) is the compression standard for still images.
* used for grayscale and quality-color images.
* Like to voice compression, JPEG is a lossy process.
* Fig shows an overview of a typical JPEG process

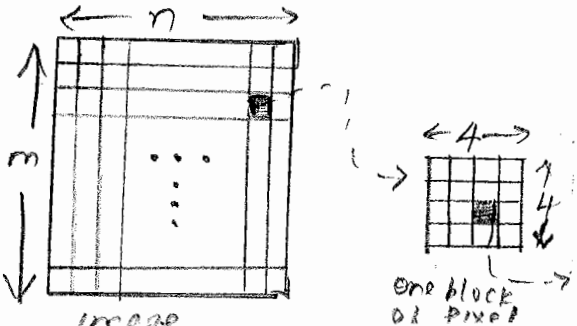


* It consists of 3 processes
  1.) Descrete cosine transform (DCT) - complex
  2.) Quantization.
  3.) Compression or encoding
  → converts a snapshot of a real image into a matrix of corresponding values.
  → converts the values generated by DCT to simple numbers in order to occupy less bandwidth.
  → makes quantized values as compact as possible. It is normally lossless & uses std compression techniques.

< Before describing these 3 blocks, nature of digital image is described here >

Raw image Sampling and DCT

* As with a voice signal, we first need samples of a raw image; a picture.
  pictures are of 2 types
  → photographs: contains no digital data.
  → images: contains digital data suitable for ~~images~~ computer n/ws.
* An image is made up of mxn blocks of picture units, or pixels, as shown.



Note: A pixel in a monochrome image typically consists of 8 bits to represent $2^8 = 256$ shades of gray, ranging from white to black, as shown below.

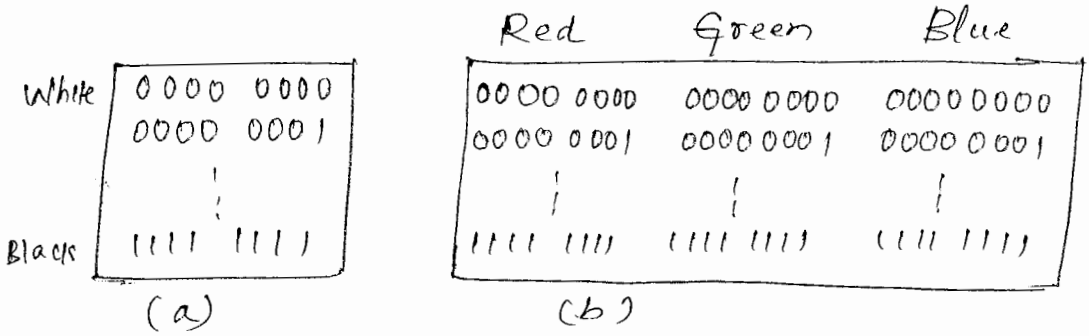


fig: still image in bits
(a) monochrome codes for still image
(b) color codes for still image.

JPEG files.

* Color images are based on the fact that any color can be formed using a combination of base colors (R G B) (ie varying the intensity of 3 primary colors)
* Each pixel will have R G B component. It will have 8 bits to represent each one.
* Thus, each pixel can be represented by using 24 bits allowing 2^{24} different colors.

Ex: A JPEG based comp screen can consists of 1024 X 1280 pixels. Consequently, this computer image requires $(1024 \times 1280) * 24 = 31457280$ bits. If a video consists of 30 images per second, a 943 Mb/s bandwidth is required.

GIF files

* Graphics interchange format (GIF) is an image file format that reduces the no. of colors to 256.
* stores upto $2^8 = 256$ colors in a table & covers range of colors
* ∴ 8 bits are used to represent a single pixel
* uses a variation of lempel-ZIV encoding

DCT process

* Discrete cosine transform (DCT) is a lossy compression process that begins by dividing a raw image into a series of standard N x N pixel blocks.
* For an N x N pixel block, the DCT process is summarized in two steps

1. Form a P[x][y] matrix to represent the collection of light intensity values taken from various points of a real raw image.
2. Convert the values of P[x][y] matrix to matrix with normalised and reduced values denoted by T[i][j] obtained as follows.

$$T[i][j] = \frac{2}{N} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \cos\left(\frac{\pi i(2x+1)}{2N}\right) \cos\left(\frac{\pi i(2y+1)}{2N}\right)$$

where,

$$C(i) = \begin{cases} 1/\sqrt{2} & \text{for } i=0 \\ 0 & \text{otherwise} \end{cases}$$

$$C(j) = \begin{cases} 1/\sqrt{2} & \text{for } j=0 \\ 0 & \text{otherwise} \end{cases}$$

note: T[i][j] reduces the BW required to transmit the image. Here, $0 \le i \le N-1$ & $0 \le j \le N-1$ T[i][j] is also N x N matrix where

Example:

* An 8 x 8 matrix P[x][y] for an image is formed as shown in fig (a)

$$\begin{bmatrix} 22 & 31 & 41 & 50 & 60 & 80 & 91 \\ 29 & 42 & 52 & 59 & 80 & 90 & 101 \\ 40 & 51 & 59 & 70 & 92 & 100 & 110 \\ 51 & 62 & 70 & 82 & 101 & 109 & 119 \\ 60 & 70 & 82 & 93 & 109 & 120 & 130 \\ 70 & 82 & 90 & 100 & 121 & 130 & 139 \\ 79 & 91 & 100 & 110 & 130 & 140 & 150 \\ 91 & 97 & 110 & 120 & 140 & 150 & 160 \end{bmatrix}$$

(a) P[x][y]

$$\begin{bmatrix} -716 & -179 & 0 & -190 & 0 & -6 & 0 & -1 \\ -179 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) T[i][j]

* This matrix is converted into matrix T[i][j] using the equation given above.
* The values of T[i][j] elements at the receiver can be converted back to matrix P[x][y] by using the following function

$$p[x][y] = \frac{2}{N} c(i)c(j) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos\left(\frac{\pi i(2i+1)}{2N}\right) \cos\left(\frac{\pi i(2j+1)}{2N}\right)$$

Quantisation

* The matrix T[i][j] is quantised to another matrix, Q[i][j] to further scaledown the values with fewer distinct numbers and more consistent patterns to get better bandwidth advantages.

procedure:

* To generate Q[i][j], the elements of matrix T[i][j] are divided by a standard number and then rounded off to their nearest integer.

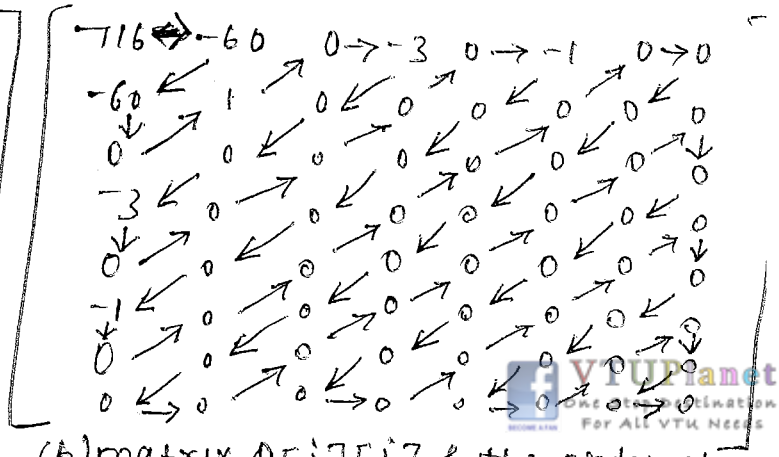
(note: elements are not divided by the same constant number as it results in too much loss.)

* To preserve as much information as possible, the elements of T[i][j] are divided by elements of an N×N matrix denoted by D[i][j], in which the values of elements decrease from upper left portion to lower right portion.

Example:

1	3	5	7	9	11	13	15
3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29

(a) Divisor matrix D[i][j]



(b) matrix Q[i][j] & the order of

* Consider an 8x8 matrix P[x,y] already corresponding to matrix T[i,j] as shown in fig(a) on page 8. Fig(a) on page 9 shows a divisor matrix D[i,j], fig(b) on page 9 shows the corresponding matrix Q[i,j] resulting from the quantization process.

note: process of quantization is not reversible. ie values of Q[i,j] cannot be exactly converted back to T[i,j]
∴ quantization phase is a lossy process.

Encoding

* last phase of JPEG process. It does the task of compression. A practical approach to compress the matrix Q (fig b on page 9) is to use run length encoding to eliminate 0s.
* logical way to scan this matrix is in the order illustrated by arrows in fig b.
* This method of scanning induces a better rule: Scanning should always start from the upper left corner element of the matrix.
* once the run length coding is processed, JPEG uses some type of huffman coding or arithmetic coding for nonzero values.

MOVING IMAGES AND MPEG COMPRESSION

* A motion image, or video is a rapid display of still images
* common standard that defines the video compression is moving pictures expert Group (MPEG), which has several branch standards.

=> MPEG-1, primary for video on CD-ROM.
- MPEG-2, for multimedia entertainmt & HDTV & satellite broadcasting industry.
- MPEG-4, for object oriented video compression & video conferencing over low BW channels
- MPEG-7, for broad range of demands requiring large BW
MPEG-21, for interaction among various MPEG groups.

process:

* A base frame is sent first, and successive frames are encoded by computing the differences.
* The receiver can reconstruct frames based on the first base frame and the submitted differences. However, frames of a completely new scene in a video may not be compressed this way, as the difference b/w two scenes is substantial.
* Depending on the relative position of a frame in a sequence, it can be compressed through one of the following types of frames:

1. Interimage (I) frames: An I frame is treated as a JPEG still image and compressed using DCT.
2. Predictive (P) frames: These frames are produced by computing the difference b/w current and a previous I or P fram
3. Bidirectional (B) frames: A B frame is like P frame. But the B frame considers the difference b/w a previous, current, and future frames.

* Fig illustrates a typical grouping of frames, ... with I, P, and B frames forming a sequence. Normally there is a P frame b/w each two groups of B frame.

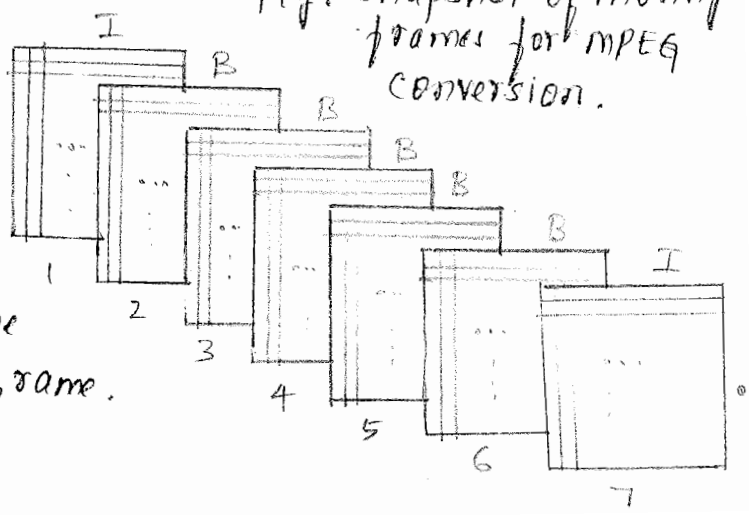


fig: snapshot of moving frames for MPEG conversion.

MP3 and Streaming Audio

* MPEG-1 layer 3 (mp3) technology compresses audio for networking and producing CD-quality sound. The sampling part of PCM is performed at a rate of 44.1 KHz to cover the maximum of 20 KHz of audible signals.
* 60 min (3600 sec) CD requires about 1.4 x 3600 = 5040 Megabits or 630MB.

LIMITS OF COMPRESSION WITH LOSS

* consider a comm system in which a source signal is processed to produce sequence of 'n' words as shown.

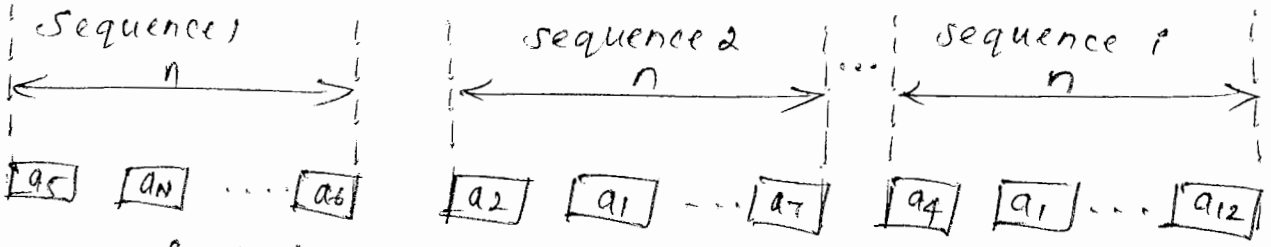


fig: A model of data sequences.

* These sequences of digital bits at the o/p of information source can be compressed in the source encoder unit to save the transmission link BW.
* An information source can be modeled by a random process $X_n = \{X_1, X_2 \ldots X_n\}$, where X_i is a random variable taking on values from a set of values as $\{a_1, a_2 \ldots a_N\}$, called alphabet.

Basics of information Theory

* let P_{K_1} and P_{K_2} be the probabilities of an information source's o/ps a_{K_1} and a_{K_2} respectively.
Let $I(P_{K_1})$ and $I(P_{K_2})$ be the information content of a_{K_1} and a_{K_2} respectively. Then, the following 5 facts apply.

1. $I(P_K)$ depends on P_K
   As probability ↑↑, information content ↓↓.
2. $I(P_K)$ = a continuous function of P_K
3. $I(P_K)$ = a decreasing function of P_K
4. $P_K = P_{K_1} \cdot P_{K_2}$ (probability of two o/ps happen in the same time)
5. $I(P_K) = I(P_{K_1}) + I(P_{K_2})$ (sum of two pieces of information)

* These facts can relate a probability of a certain data to its information content

$$I(P_K) = -\log_2 P_K = \log_2\left(\frac{1}{P_K}\right)$$

Entropy of Information

* Entropy - measure of uncertainty.
* consider an information source producing random numbers X, from a possible collection of $\{a_1, a_2 \ldots a_N\}$ with corresponding probabilities of $\{P_1, P_2 \ldots P_N\}$ and information content of $\{I(P_1), I(P_2) \ldots I(P_n)\}$ respectively. In particular, Entropy is defined as average information content of a source.

$$H_X(x) = \sum_{K=1}^{N} P_K I(P_K)$$

$$= \sum_{K=1}^{N} P_K \log_2\left(\frac{1}{P_K}\right) = -\sum_{K=1}^{N} P_K \log_2 P_K$$

Example: A source with BW 8KHz is sampled at Nyquist rate. If the result is modeled using any value from $\{-2, -1, 0, 1, 2\}$ and corresponding probabilities $\{0.05, 0.05, 0.08, 0.30, 0.52\}$. Find the entropy.

Solution: we have

entropy, $H_X(x) = -\sum_{K=1}^{N} P_K \log_2 P_K$

ie $H_X(x) = -\sum_{k=1}^{5} P_k \log_2 P_k$

$= -(P_1 \log_2 P_1 + P_2 \log_2 P_2 + P_3 \log_2 P_3 + P_4 \log_2 P_4 + P_5 \log_2 P_5)$

$= -\{0.05 \log_2 0.05 + 0.05 \log_2 0.05 + 0.08 \log_2 0.08 + 0.30 \log_2 0.30 + 0.52 \log_2 0.52\}$

$\Rightarrow$ **$H_X(x) = 0.522$ bits / sample**

The information rate in samples/sec $= 8000 \times 2$

$f_s =$ **16000 samples/sec.**

rate of information produced by the source $= f_s * H(S)$

$= 16000 * 0.522$

$=$ **8352 bits**

Joint Entropy

* Joint entropy of two discrete random variables X & Y is defined as,

$$H_{X,Y}(x,y) = -\sum_{x,y} P_{X,Y}(x,y) \log_2 P_{X,Y}(x,y)$$

where,

$P_{X,Y}(x,y) = \text{prob}[X = x \text{ and the same time } Y = y]$ & is called joint probability mass function of two random variables.

* In general, for a random process $X_n = (X_1, X_2 \ldots X_n)$ with n random variables,

$$H_{X_n}(x_n) = -\sum_{x_1 \ldots x_n} P_{x_1 \ldots x_n}(x_1 \ldots x_n) \log_2 P_{x_1, \ldots x_n}(x_1 \ldots x_n).$$

Shannon's coding Theorem

* This theorem limits the rate of data compression.
* We define a typical sequence as one in which any value a_i is repeated nP_i times. Accordingly, the probability that a_i is repeated nP_i times is obviously $P_i P_i \cdots P_i = P_i^{nP_i}$

$$\text{Prob(typical sequence)} = \prod_{i=1}^{N} P_i^{nP_i}$$

$$P_t = \prod_{i=1}^{N} P_i^{nP_i}$$

$$= \prod_{i=1}^{N} 2^{nP_i \log_2 P_i}$$

$$= 2^{(nP_1 \log_2 P_1 + \cdots + nP_N \log_2 P_N)}$$

$$\boxed{P_t = 2^{n\left(\sum_{i=1}^{N} P_i \log_2 P_i\right)}}$$

(or) $$\boxed{P_t = 2^{-n H_X(x)}}$$

Example: Assume that a sequence size of 200 of an information source chooses values from the set $\{a_1 \ldots a_5\}$ with corresponding probabilities $\{0.05, 0.05, 0.08, 0.30, 0.52\}$ Find the probability of typical sequence.

Soln: from prev ex we know that entropy, $H_X(x) = 0.522$

With $n = 200$, $N = 5$, the probability of typical sequence is the probability of a sequence in which $a_1, a_2, a_3, a_4, \& a_5$ are repeated $200 \times 0.05 = 10$ times, 10 times, 16 times, 60 times, & 104 times respectively.

Thus $$P_t = 2^{-n H_X(x)} = \boxed{2^{-200(0.522)}}$$

example: for prev ex, find the ratio of no. of typical sequences to the no. of all types of sequences.

soln: no of typical sequences $= 2^{200 \times 0.522}$

total no. of all sequences $= 5^{200}$

This ratio is almost zero, which may cause huge data loss if compressed, based on shannon's theorem.

Compression Ratio and code efficiency

* let $l_i \rightarrow$ length of code word i
  $P_i \rightarrow$ probability of code word i

then, average length of codes $\bar{R}_i = \sum_{i=0}^{N} P_i l_i$

* compression ratio $C_r = \frac{\bar{R}}{\bar{R}_x}$

where $\bar{R}_x \rightarrow$ length of source o/p before coding.

$$H_x(x) \leq \bar{R} < H_x(x) + 1$$

* code efficiency is a measure for understanding how code lengths are to the corresponding decoded data and is defined by;

$$\eta_{code} = \frac{H_x(x)}{\bar{R}}$$

COMPRESSION METHODS WITHOUT LOSS

- Arithmetic encoding ⟶ not for syllabus
- Run-length encoding
- Huffman encoding
- Lempel-ziv encoding

Run length Encoding

* Simplest, Effective for plaintext & numbers compression.
* with run length code, repeated letters are replaced by C runlength, beginning with C_c to express the compression letter count.

ie when data contain strings of repeated symbols, the strings can be replaced by following 3 characters

~~a special~~ - a special marker (here C_c)
- repeated symbol
- number of occurrences

Example:

Find the compression version of following sentance
<here b => blank.>

THISSSSSSBISBBBBANBEXAMPLEBOFBRUN-----LENGTHBCODE

solution: compression version of above sentance is

~~THIC$_c$S9~~

THIC$_c$S6B ISC$_c$B4ANBEXAMPLEBOFBRUNC$_c$-5LENGTHBCODE

note: Longer the text, smaller the compression ratio.

#Huffman Encoding

* It is an efficient frequency-dependent coding technique.
* With this algorithm, source values with smaller probabilities appear to be encoded by a longer word

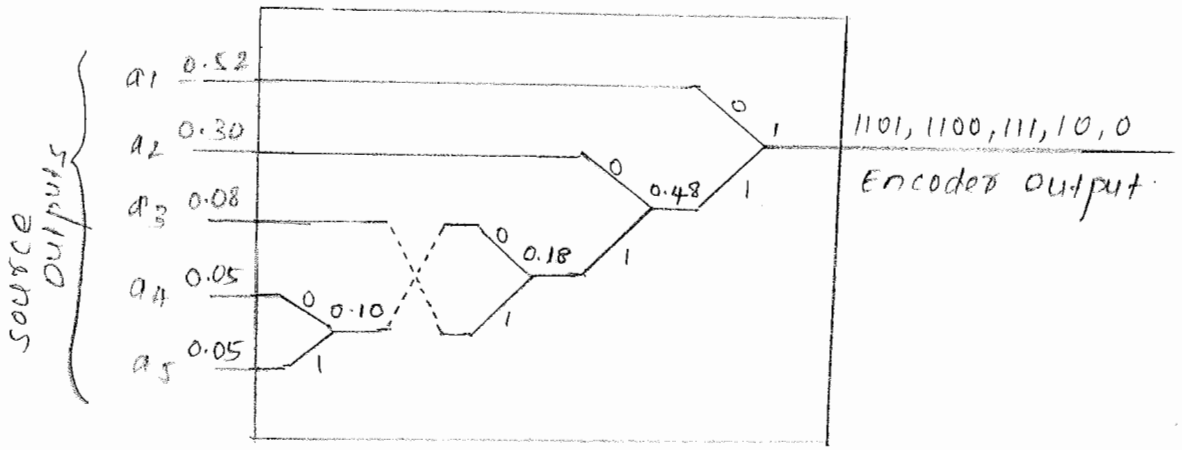
Algorithm

1. Sort output of the source in decreasing order of their probabilities, eg 0.7, 0.6, 0.6, 0.59 .... 0.02, 0.01
2. Merge the two least probabilistic outputs into a single o whose probability is sum of the corresponding probability, such as 0.02 + 0.01 = 0.03.
3. If the number of remaining outputs is 2, goto next step. otherwise goto step 1

4. Assign 0 and 1 as codes on the diagram
5. If a new o/p is the result of merging two o/ps, append the codeword with 0 and 1; otherwise stop.

Example: Design an huffman encoder for a source generating $\{a_1, a_2, a_3, a_4, a_5\}$ and with probabilities $\{0.05, 0.05, 0.08, 0.30, 0.52\}$

Solution:
Following the algorithm, the o/p of the information source is shown in the fig below.



The information related to $\{a_1, a_2, a_3, a_4, a_5\}$ is compressed to 1100, 1101, 111, 10, 0 respectively.

Lempel-Ziv Encoding

* Independent of the source statistics. Normally used for UNIX compressed files.

Algorithm

1. Any sequence of source o/p is passed in a phrase of varying length. At the first step, identify phrases of the smallest length that have not appeared so far. Note that all phrases are different, and lengths of words grow as the encoding process proceeds.
2. phrases are encoded using codewords of equal ~~words~~ length. If K_1 = number of bits are needed to describe the codeword & K_2 = number of phrases, we must have

$$K_1 = \log_2 \lceil K_2 \rceil_2$$

3. A code is the location of the prefix to the phrases.
4. A code is followed by the last bit of parser output to double check the last bit.

Example: For the following string of bits, find the encoded lempel-ziv words

1101111011000001010010001111010101100

Solution: Implementing step1 on the string, there are 14 phrases, as follows:

1-11-10-111-0-110-00-001-01-0010-0011-1101-010-1100

Thus $K_2 = 14$

& $K_1 = \log_2 \lceil 14 \rceil_2$

$= 4$.

Table shows the encoded words as steps 3 and 4 are applied on the parser o/p.

parser o/p	location	encoded o/p
1	0001	00011
11	0010	00101
10	0011	00110
111	0100	01001
0	0101	01010
110	0110	01100
00	0111	01110
001	1000	10001
01	1001	10011
0010	1010	10100
0011	1011	10111
1101	1100	11001
010	1101	11010
1100	1110	11100

CHAPTER 7B:

VOIP AND MULTIMEDIA NETWORKING

OVERVIEW OF IP TELEPHONY

+ Basic component of an IP telephone system include IP telephones, internet backbone, and signaling servers. (refer fig)

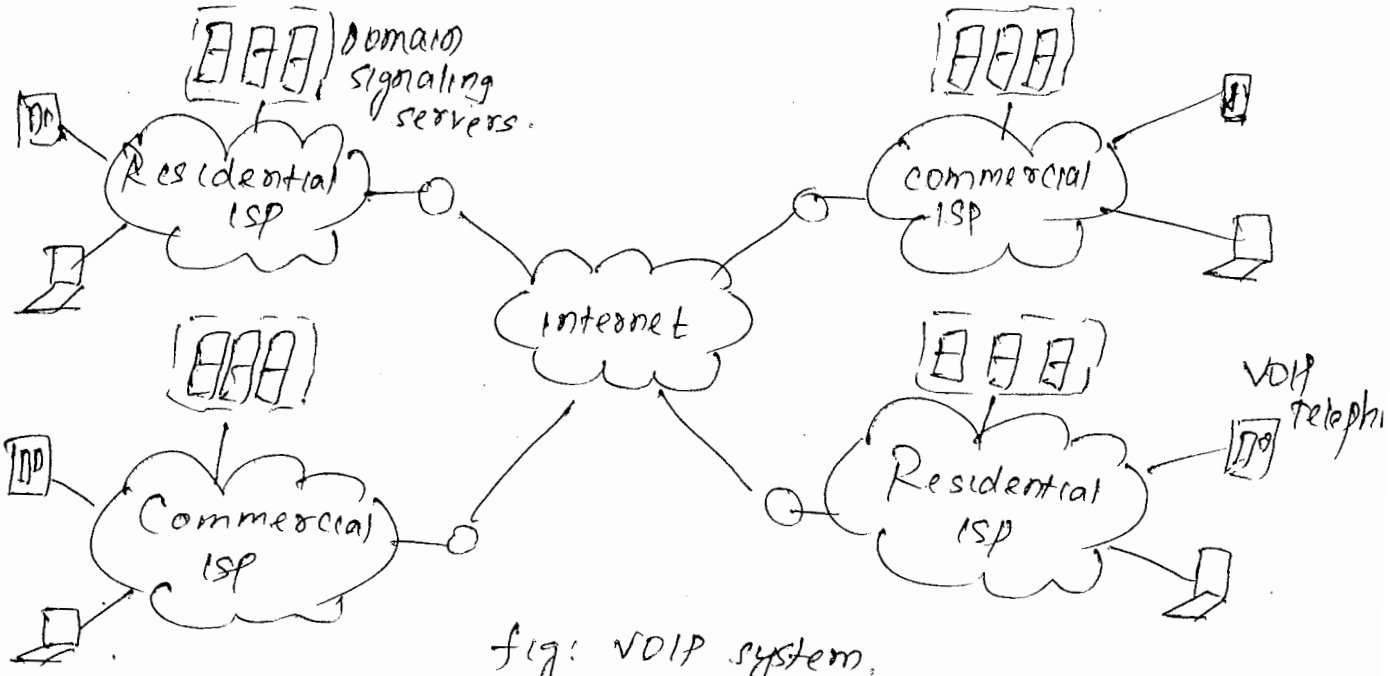


fig: VOIP system.

IP telephone can be used to make telephone calls over IP networks.
Voice over IP (VoIP) or IP telephony uses packet switched n/ws to carry voice traffic in addition to data traffic.

A VOIP network is operated thru' two set of protocols

- signalling protocols
- Real time packet-transport protocols.
  - Handle call setup & are controlled by signalling servers.

VOIP Quality of service.

* A VOIP connection has several QoS factors
  - packet loss is accepted to certain extent
  - packet delay is normally unacceptable.
  - jitter, as the variation in packet arrival time, is not acceptable after a certain limit.

VOIP SIGNALLING PROTOCOLS.

* signaling is required for call set up, call management, and call termination.
* IP telephone systems can use either a distributed or centralized signaling scheme.
  - distributed: enables two IP telephones to communicated using client/server model.
  - centralized: Uses conventional model & provide some level of guarantee.

* Three well known signalling protocols are -
  1. Session Initiation protocol (SIP)
  2. H.323 protocols.
  3. Media gateway control protocol (MGCP).

Session Initiation Protocol (SIP).

* operates in Application layer in the five-layer TCP/IP model.
* performs both unicast and multicast sessions and supports user mobility.

SIP components:

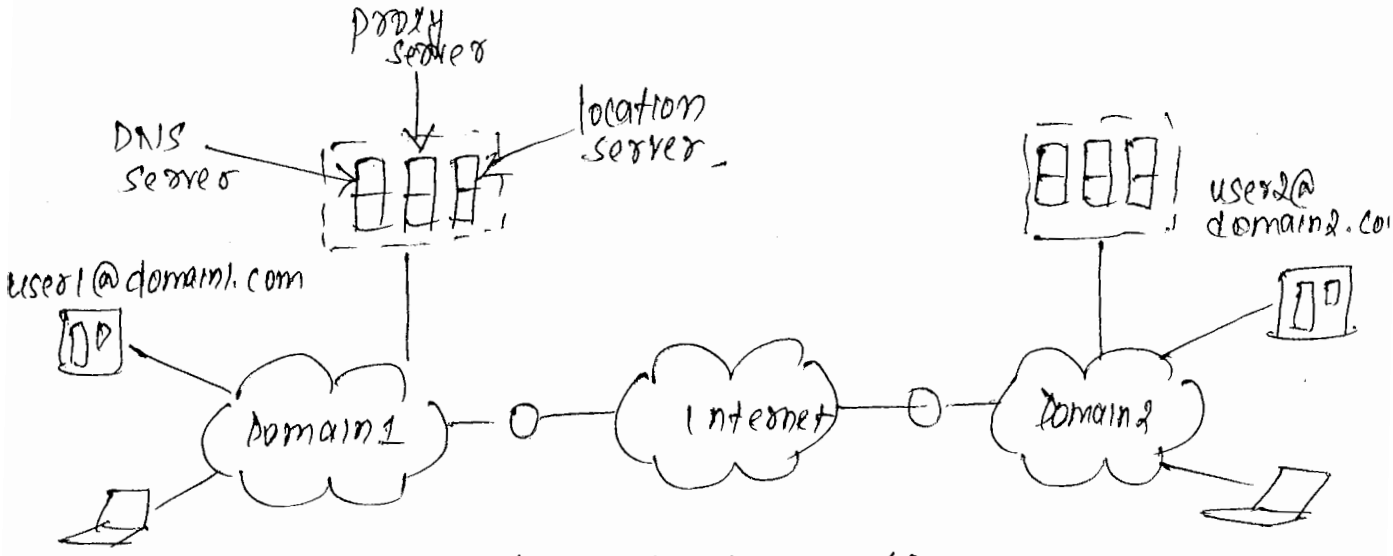


fig: overview of ISP.

* A call is initiated from a user agent (user's IP telephone system) user agent is identified using its associated domain.

SIP consists of following 5 servers:

1. **DNS server:**
   - maps the domain name to an IP address in the user information Database (UID)
   - Each user is normally configured with more than one DNS server.

2. **Proxy server:**
   - forwards requests from a user agent to a different location and handles authorizations by checking whether the caller is authorized to make a particular call.

3. Location server:

- Responsible for UID management
- Interacts with database during call setup.
- Each proxy server is normally configured with more than one location server.

4. Redirect Server:

- Performs call forwarding and provides Alternative paths for user agent.

5. Registrar Server:

- Responsible for registering users in the system & updating the UID that the location server consults.

Session signalling and Numbering.

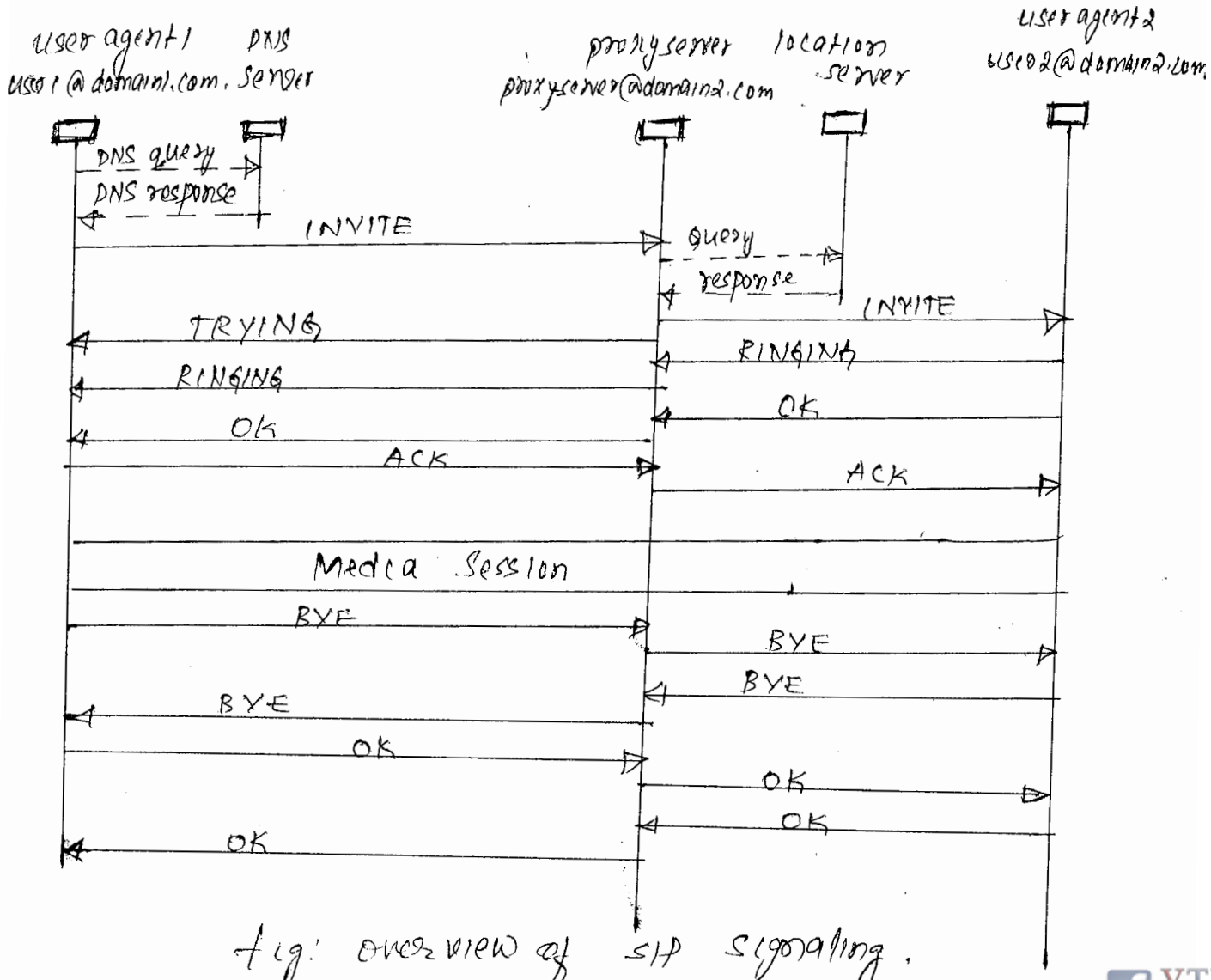


fig: Overview of SIP signaling.

* Fig shows SIP session b/w agents 1 & 2 ie user1@domain1.com & user2@domain2.com.

- Here user1 places a call to contact user2.
- user1 first communicates with its DNS server to map the domain name to an IP address in the SIP UID (thru' DNS query and response)
- INVITE message: used for session creation. It contains informations such as from, to, via, and callid, in addition to routing inform
- proxy server inturn communicates with location server of called part
- once user2 receives a connection query (INVITE), TRYING signal is propagated to user1 from proxy server indicating the call is being routed.
- RINGING signal is transmitted from user2 back to user1
- when user2 accepts the call, OK signal is issued back to user1 to indicate the called party has accepted the call
- This ~~sig~~ last signal is acknowledged by an ACK msg without any response.
- TWO IP phones communicates directly thru' media session using real time protocol.
- At the end of conversation, Bye message is used for a session termination, ending the call.

H.323 protocols.

* implemented in layer 5 of TCP/IP model, and run over either TCP or UDP.
* these protocols ~~provide~~ interact to provide ideal telephone comm^n, providing phone nos to IP addr mapping, handling digitized audio streaming in IP telephony, & providing signaling functions for call setup and call management.

H.323 components.

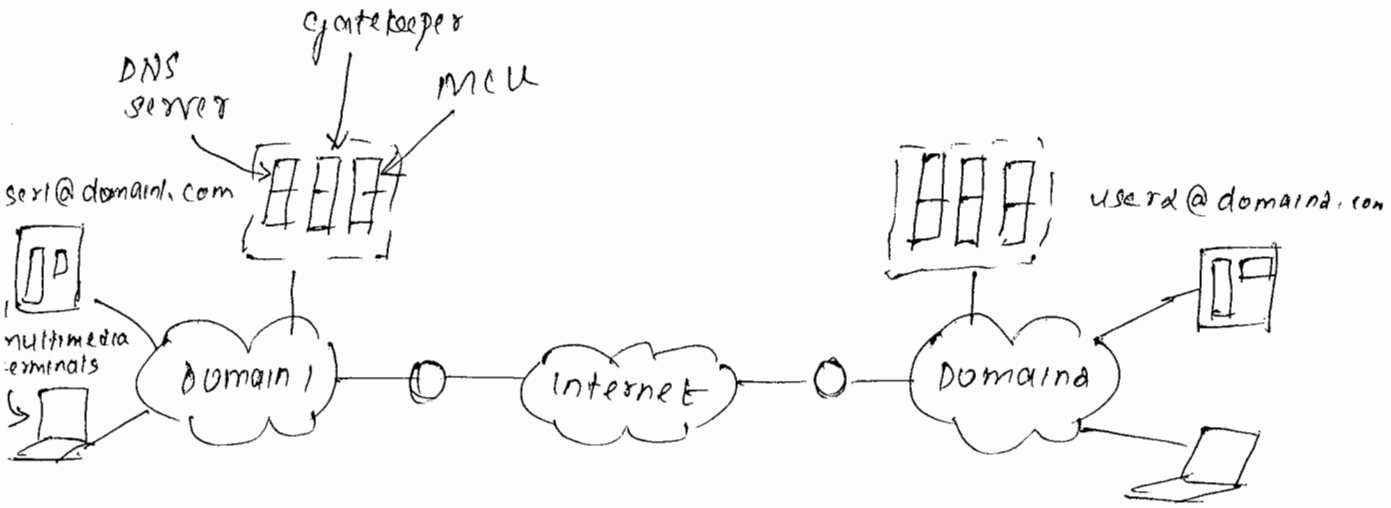


fig: Overview of H.323 protocol connection.

1. Multimedia Terminals: Designed to support video and data traffic & to provide support for IP telephony
2. DNS Server: As in SIP, DNS maps a domain name to an IP addr.
3. Gateway: is a router that serves as an interface b/w IP telephone system & traditional telephone system
4. Gatekeeper: Central center that performs all the locn and signalling funcns It monitors & coordinates the activities of the gateway. gateway also performs some signaling functions.
5. Multicast or Multipoint control unit (MCU): provides some multipoint services such as conference calls.

REAL TIME MEDIA TRANSPORT PROTOCOLS.

includes
- Real time Transport protocol (RTP)
- Real time control protocol (RTCP).

RTP

* provides some basic ~~functa~~ functionalities to real time applications and includes some specific functions to each application.
* It runs on top of the transport protocol as UDP.

Real-time session and Data Transfer.

* A session is a logical connection b/w an active client and an active server and is defined by following entities -
  1. RTP port no: represents the destn port addr. of RTP session.
  2. IP address of the RTP entity which involves an RTP session
     ↳ either unicast or multicast.

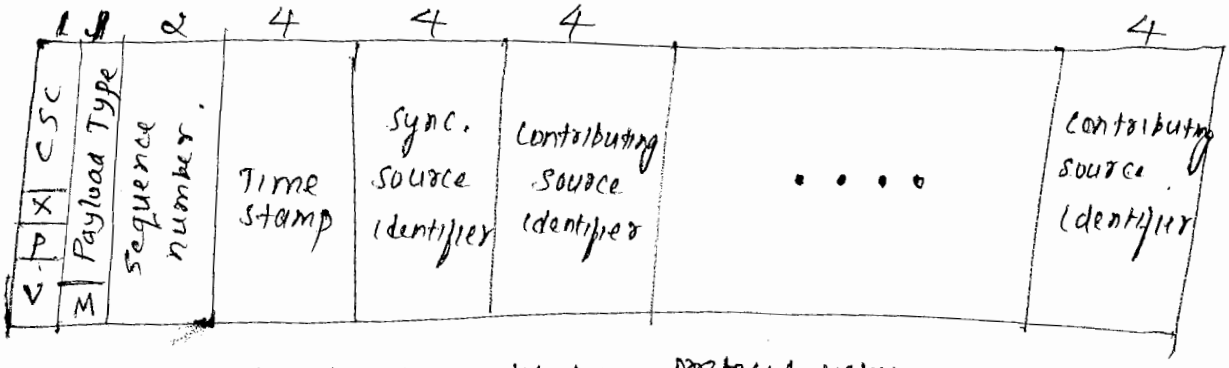
* RTP uses two relays for data transmission -
  ↳ An intermediate system that acts as both sender and ~~s~~ receiver.
  It convert data format from a system into a form that other system can process easily.

Relays are of two types

1. Mixer Relay: is an RTP relay that combines the data from two or more RTP entities into a single stream of data. It can either retain or change the data format
2. Translator Relay: is a device that generates one or more RTP packets for each incoming RTP packet. Format of outgoing packet may be different from that of the incoming packet.

RTP packet header

Byte:



1. Version (V): 2 bit field, indicating protocol version
2. Padding (P): 1 bit, indicating existing of a padding field at the end of payload.
3. Extension (X): 1 bit field indicating the use of an extension header for RTP.
4. Contributing source count (CSC): 4 bit, indicates no. of contributing source identifiers.
5. Marker (M): 1 bit, indicating boundaries in a stream of data to appln. for video Applcns, it is used to indicate end of frame
6. Payload Type: 7 bit, specifies Type of RTP payload. Also contains inform on the use of compression or encryption.
7. Sequence no: 16 bit, Sender uses this field to identify a particular packet within a sequence of packets. This field is used to detect packet loss & for packet ~~reordin~~ re ordering.
8. Timestamp: 32 bit, enables the receiver to recover timing inform. It indicates the timestamp when the first byte of data in the payload was generated.
9. Synchronization Source identifier: randomly generated field used to identify the RTP source in an RTP session.
10. Contributing source Identifier: Optional field in the header to indicate the contributing sources for the data.

Estimation of Jitter in Real time ~~packet~~ Traffic.

↳ measure of delay experienced by RTP packets in a given session.

let,

t_i = time stamp of RTP data packet i indicated by source.

a_i = Arrival time of RTP data packet i at the receiver.

d_i = Measure of ~~the~~ difference b/w interarrival time of RTP packets at receiver & the one for packet departure from the source.
This value represents the difference in packet spacing at source & receiver.

$$d_i = (a_i - a_{i-1}) - (t_i - t_{i-1})$$

let

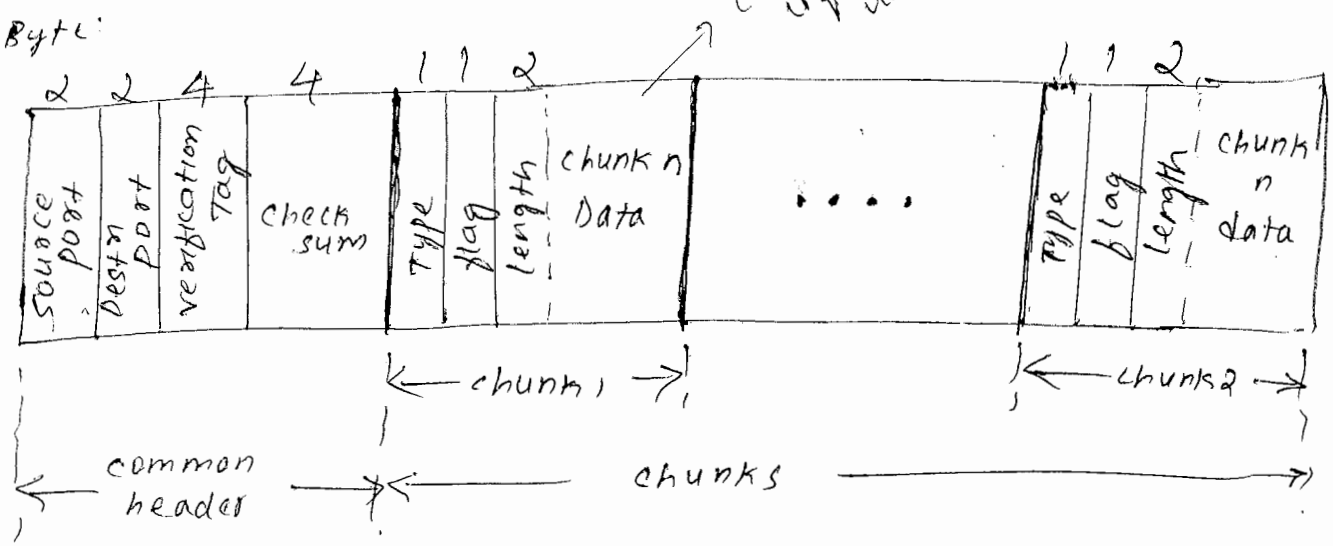
$E[i]$ = Estimate of average jitter until the time of packet i arrival

$$E[i] = K(E[i-1] + |d_i|)$$

STREAM CONTROL TRANSMISSION PROTOCOL (SCTP)

* It provides a general purpose transport protocol for message-oriented applications.
* features of SCTP are —
1. protocol is error free.
2. Has ordered & unordered delivery modes
3. SCTP has effective methods to avoid flooding, congestion, and masquerade attacks.
4. This protocol is multipoint and allows several streams within a connection.

SCTP packet structure



* SCTP packet is also called PDU. Each packet consists of common header & chunks.
* Chunk header starts with a chunk type field used to distinguish data chunks & any other types of control chunks.
* flag & length field are used to indicate chunk size.
* verification tag is exchanged b/w end point servers at startup to verify two servers involved.
* SCTP packets are protected by a 32 bit checksum.

Each packet has n chunks, & each chunk is of two types
- payload data chunks
  ↳ for transmitting actual ~~data~~ streaming data
- control chunks
  ↳ for signaling & control.
    ↳ these are of several types as follows.

1. Initiation - to initiate a session (SCTP) blw two end points
2. Initiation Ack
3. selective Ack
4. Heartbeat request
5. Heart beat Ack.
6. Abort - to close a session
7. Shutdown - to initiate a graceful close of session.
8. Shutdown ack.
9. Operation error - to notify other party of a certain error
10. State cookie - sent by source to its peer to complete initialization process
11. Cookie Ack.
12. Shutdown complete.