Traffic Management

- The main objectives of traffic management are efficient use of network resources & deliver QoS.
- Traffic Management is classified into three levels that are Packet level, Flow level and Flow aggregated level.

Traffic Management at Packet Level

o Queueing & scheduling at switches, routers and multiplexers.

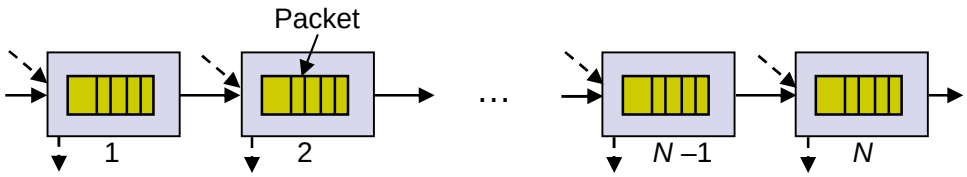


Figure: - End-to-End QoS of a packet along a path traversing N Queueing System

- The path traversed by packet through a network can be modeled as sequence of Queueing systems as shown in above figure.
- A packet traversing network encounters delay and possible loss at various multiplexing points.
- End-to-end performance is sum of the individual delays experienced at each system.
- Average end-to-end delay is the sum of the individual average delay.
- To meet the QoS requirements of multiple services, a queueing system must implement strategies for controlling the transmission bit rates.

The different strategies for Queue scheduling are:-

1. FIFO QUEUEING
2. PRIORITY QUEUEING
3. FAIR QUEUEING
4. WEIGHTED FAIR QUEUEING

1) FIFO QUEUEING

- Transmission Discipline: First-In, First-Out
- All packets are transmitted in order of their arrival.
- Buffering Discipline:- Discard arriving packets if buffer is full
- Cannot provide differential QoS to different packet flows
- Difficult to determine performance delivered
- Finite buffer determines a maximum possible delay
- Buffer size determines loss probability, but depends on arrival & packet length statistics.

FIFO Queueing with Discard Priority

FIFO queue management can be modified to provide different characteristics of packet-loss performance to different classes of traffic.

- The above Figure 7.42 (b) shows an example with two classes of traffic.
- When number 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 the buffer is not full.

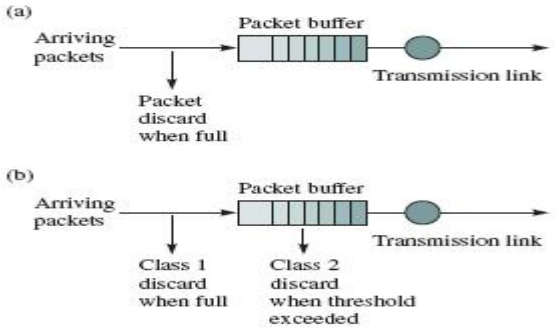


FIGURE 7.42 (a) FIFO queueing; (b) FIFO queueing with discard priority

2) Head of Line (HOL) Priority Queueing

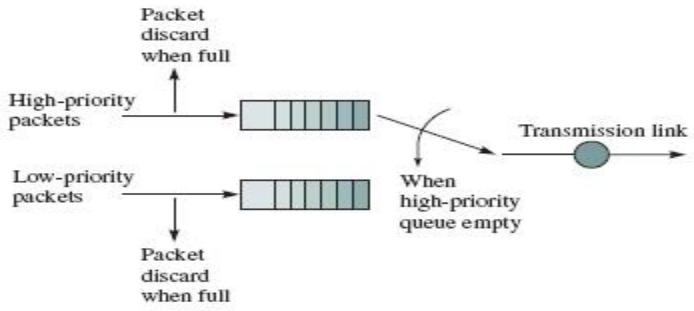


FIGURE 7.43 HOL priority queueing

- Second queue scheduling approach which defines number of priority classes.
- A separate buffer is maintained for each priority class.
- High priority queue serviced until empty and high priority queue has lower waiting time
- Buffers can be dimensioned for different loss probabilities
- Surge in high priority queue can cause low priority queue to starve for resources.
- It provides differential QoS.
- High-priority classes can hog all of the bandwidth & starve lower priority classes
- Need to provide some isolation between classes

Sorting packets according to priority tags/Earliest due Date Scheduling

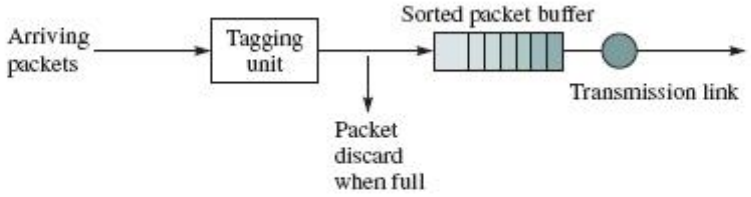


FIGURE 7.44 Sorting packets according to priority tag

- Third approach to queue scheduling
- Sorting packets according to priority tags which reflect the urgency of packet needs to be transmitted.
- Add Priority tag to packet, which consists of priority class followed by the arrival time of a packet.

- Sort the packet in queue according to tag and serve according to HOL priority system
- Queue in order of “due date”.
- The packets which requires low delay get earlier due date and packets without delay get indefinite or very long due dates

3) Fair Queueing / Generalized Processor Sharing

- Fair queueing provides equal access to transmission bandwidth.
- Each user flow has its own logical queue which prevents hogging and allows differential loss probabilities
- C bits/sec is allocated equally among non-empty queues.
- The transmission rate = C / n bits/second, where n is the total number of flows in the system and C is the transmission bandwidth.
- Fairness: It protects behaving sources from misbehaving sources.
- Aggregation:
  - Per-flow buffers protect flows from misbehaving flows
  - Full aggregation provides no protection
  - Aggregation into classes provided intermediate protection
- Drop priorities:
  - Drop packets from buffer according to priorities
  - Maximizes network utilization & application QoS
  - Examples: layered video, policing at network edge.

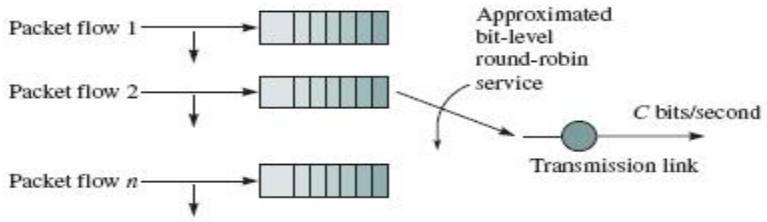


FIGURE 7.45 Fair queueing

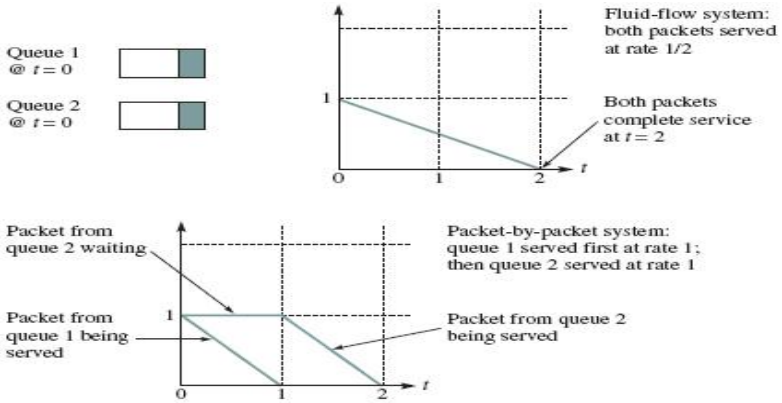


FIGURE 7.46 Fluid-flow and packet-by-packet fair queueing (two packets of equal length)

The above figure 7.46 illustrates the differences between ideal or fluid flow and packet-by-packet fair queueing for packets of equal length.

- Idealized system assumes fluid flow from queues, where the transmission bandwidth is divided equally among all non-empty buffers.
- The figure assumes buffer1 and buffer 2 has single L-bit packet to transmit at t=0 and no subsequent packet arrive.
- Assuming capacity of C=L bits/second=1 packet/second.
- Fluid-flow system transmits each packet at a rate of ½ and completes the transmission of both packets exactly at time=2 seconds.
- Packet-by-packet fair queueing system transmits the packet from buffer 1 first and then transmits from buffer 2, so the packet completion times are 1 and 2 seconds.

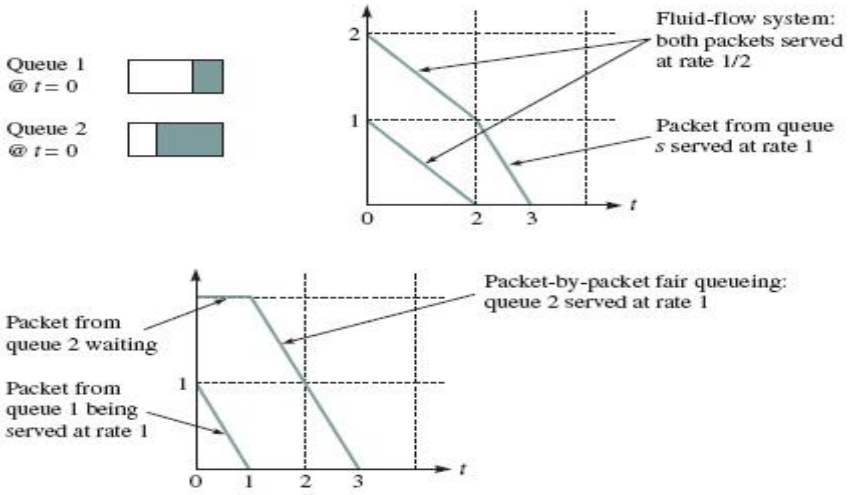


FIGURE 7.48 Fluid flow and packet-by-packet fair queueing (two packets of different lengths)

The above figure 7.48 illustrates the differences between ideal or fluid flow and packet-by-packet fair queueing for packets of variable length.

- The fluid flow fair queueing is not suitable, when packets have variable lengths.
- If the different user buffers are serviced one packet at a time in round-robin fashion, then we do not obtain fair allocation of transmission bandwidth.
- Finish tag is number used for the packet and the packet with smallest finish tag will be served first, and finish tag is computed as follows.
- Finish tag is used as priorities in packet-by-packet system.

Consider Bit-by-Bit Fair Queueing

- Assume n flows, n queues
- 1 round = 1 cycle serving all n queues
- If each queue gets 1 bit per cycle, then 1 round is the number of opportunities that each buffer has had to transmit a bit.
- Round number = number of cycles of service that have been completed

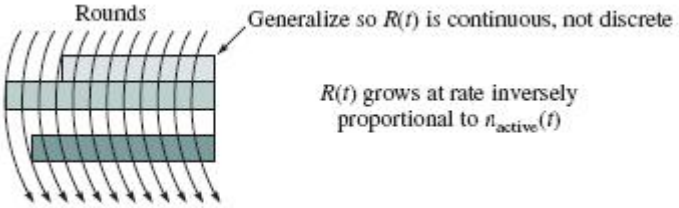


FIGURE 7.47 Computing the finishing time in packet-by-packet fair queueing and weighted fair queueing

- If packet arrives to idle queue:
  Finishing time = round number + packet size in bits
- If packet arrives to active queue:

Finishing time = finishing time of last packet in queue + packet size

Computing the Finishing Time

- F(i,k,t) = finish time of kth packet that arrives at time t to flow i
- P(i,k,t) = size of kth packet that arrives at time t to flow i
- R(t) = round number at time t

- Fair Queueing:

$$F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)$$

4) Weighted Fair Queueing (WFQ)

- WFQ addresses the situation in which different users have different requirements.
- Each user flow has its own buffer and each user flow also has weight.
- Here weight determines its relative bandwidth share.
- If buffer 1 has weight 1 and buffer 2 has weight 3, then when both buffers are nonempty, buffer 1 will receive 1/(1+3)=1/4 of the bandwidth and buffer 2 will receive ¾ of the bandwidth.

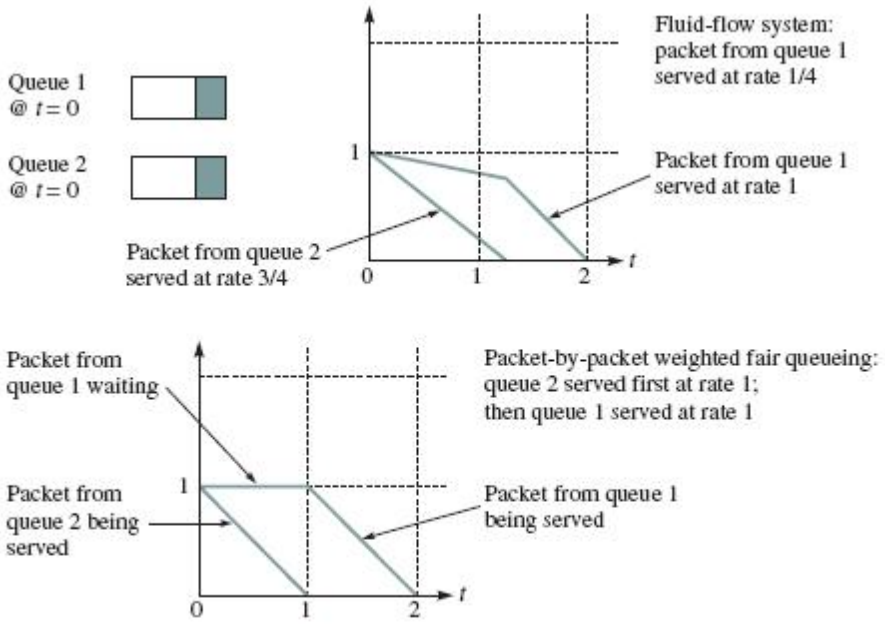


FIGURE 7.49 Fluid flow and packetized, weighted fair queueing

In the above figure,

- In Fluid-flow system, the transmission of each packet from buffer 2 is completed at time t=4/3, and the packet from buffer 1 is completed at t=2 seconds.
- In the above figure buffer1 would receive 1 bit/round and buffer 2 would receive 3 bits/second.
- Packet-by-packet weighted fair queueing calculates its finishing tag as follows

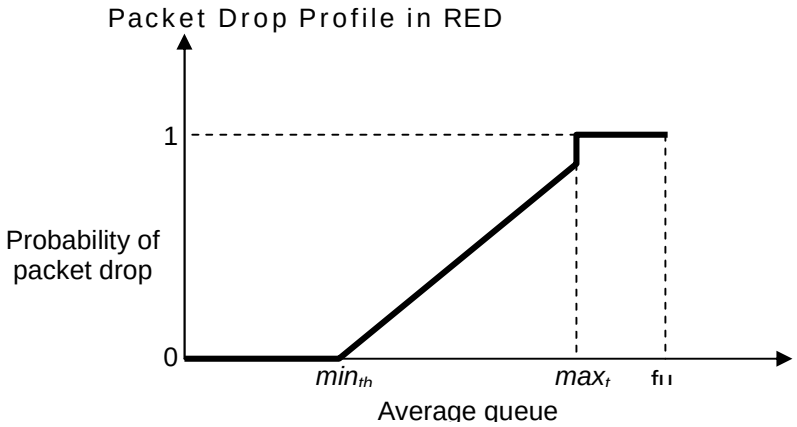
  $F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)/wi$
- The above figure also shows the completion times for Packet-by-packet weighted fair queueing.
- The finish tag for buffer1 is F(1,1)=R(0)+1/1 =1 and finish tag for buffer 2 is F(2,1) =R(0) + 1/3 =1/3.
- Therefore the packet from buffer 2 is served first and followed by packet from buffer 1.

Buffer Management: - Random Early Detection (RED)

- An approach to preventing unfair buffer hogging by detecting congestion when a buffer begins to reach certain level and it notifies the source to reduce the rate at which they send packets.
- Packets produced by TCP will reduce input rate in response to network congestion
- RED is a buffer management technique that attempts to provide equal access to FIFO system by randomly dropping arriving packets before the buffer overflows.
- A dropped packet provides feedback information to the source and informs the source to reduce its transmission rate.
- Early drop: discard packets before buffers are full
- Random drop causes some sources to reduce rate before others, causing gradual reduction in aggregate input rate.
- Min_{th} and Max_{th} are the two thresholds defined
- RED algorithm uses average queue length, when average queue length is below Min_{th} , RED does not drop any arriving packets.
- When average queue length is between Min_{th} and Max_{th}, RED drops an arriving packet with an increasing probability as the average queue length increases.
- Packet drop probability increases linearly with queue length
- RED improves performance of cooperating TCP sources.
- RED increases loss probability of misbehaving sources

Algorithm:

- Maintain running average of queue length
- If Qavg < minthreshold, do nothing
- If Qavg > maxthreshold, drop packet
- If in between, drop packet according to probability
- Flows that send more packets are more likely to have packets dropped



Traffic Management at the Flow Level

- Management of individual traffic flows & resource allocation to ensure delivery of QoS(e.g. Delay, jitter, loss)
- Traffic management at flow level operates on the order of milliseconds to seconds.
- It is concerned with managing the individual traffic flow to ensure the QoS (e.g. delay, jitter, loss) requested by user is satisfied.
- The purpose of Traffic Management at the Flow Level is to control the flows of traffic and maintain performance even in presence of traffic overload.
- The process of managing the traffic flow in order to control congestion is called congestion control.
- Congestion occurs when a surge of traffic overloads network resources

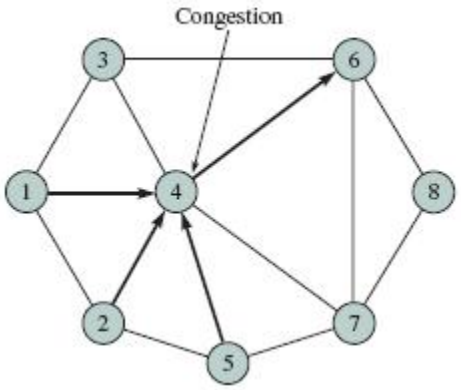


FIGURE 7.50 A congested switch

Approaches to Congestion Control:

- Preventive Approaches: Scheduling & Reservations
- Reactive Approaches: Detect & Throttle/Discard

Ideal effect of congestion control:
Resources used efficiently up to capacity available

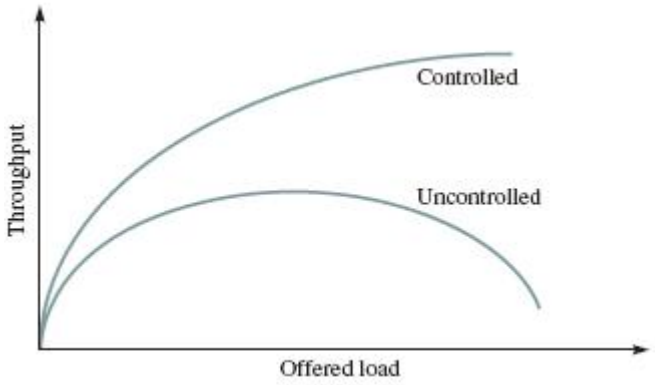


FIGURE 7.51 Throughput drops when congestion occurs

Open-loop control and closed-loop control are the two logical approaches of congestion control.

Open-Loop Control

- It prevents congestion from occurring.
- It does not depend on feedback information to react to congestion.
- Network performance is guaranteed to all traffic flows that have been admitted into the network
- It depends on three Key Mechanisms and they are:-
  - Admission Control
  - Policing
  - Traffic Shaping

Admission Control

- It is a network function that computes the resource (bandwidth and buffers) requirements of new flow and determines whether the resources along the path to be followed are available or not available.
- Before sending packet the source must obtain permission from admission control.
- Admission control decides whether to accept the flow or not.
- Flow is accepted, if the QoS of new flow does not violate QoS of existing flows
- QoS can be expressed in terms of maximum delay, loss probability, delay variance, or other performance measures.
- QoS requirements:
  - Peak, Avg., Min Bit rate
  - Maximum burst size
  - Delay, Loss requirement
- Network computes resources needed
  - "Effective" bandwidth
- If flow accepted, network allocates resources to ensure QoS delivered as long as source conforms to contract

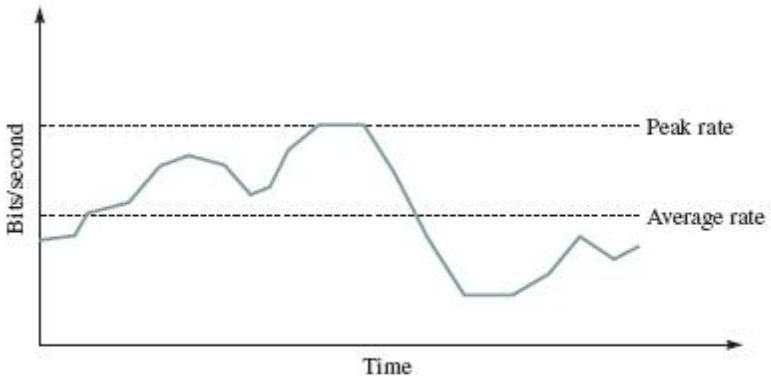


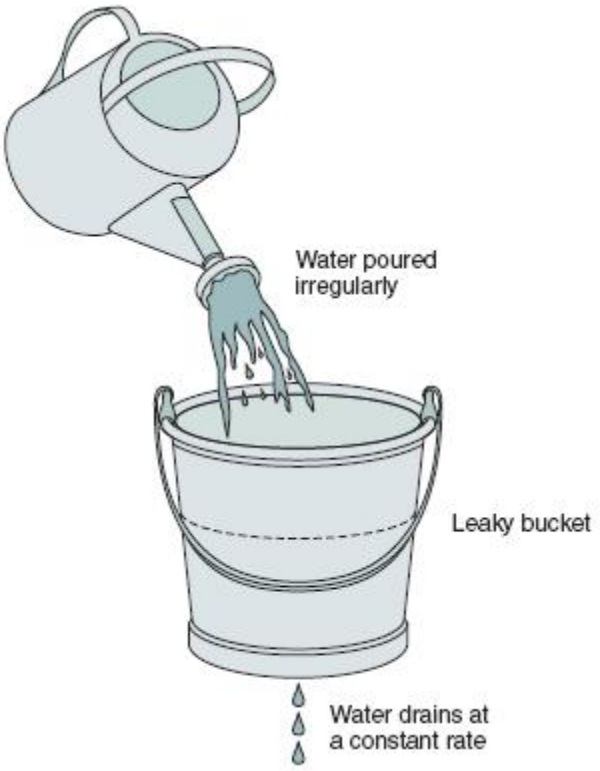
FIGURE 7.52 Example of a traffic flow

Policing

- Network monitors traffic flows continuously to ensure they meet their traffic contract.
- The process of monitoring and enforcing the traffic flow is called policing.
- When a packet violates the contract, network can discard or tag the packet giving it lower priority
- If congestion occurs, tagged packets are discarded first
- Leaky Bucket Algorithm is the most commonly used policing mechanism
    - Bucket has specified leak rate for average contracted rate
    - Bucket has specified depth to accommodate variations in arrival rate
    - Arriving packet is conforming if it does not result in overflow

Leaky Bucket algorithm can be used to police arrival rate of a packet stream

FIGURE 7.53 A leaky bucket



Let X = bucket content at last conforming packet arrival
Let ta be last conforming packet arrival time = depletion in bucket

Leaky Bucket Algorithm

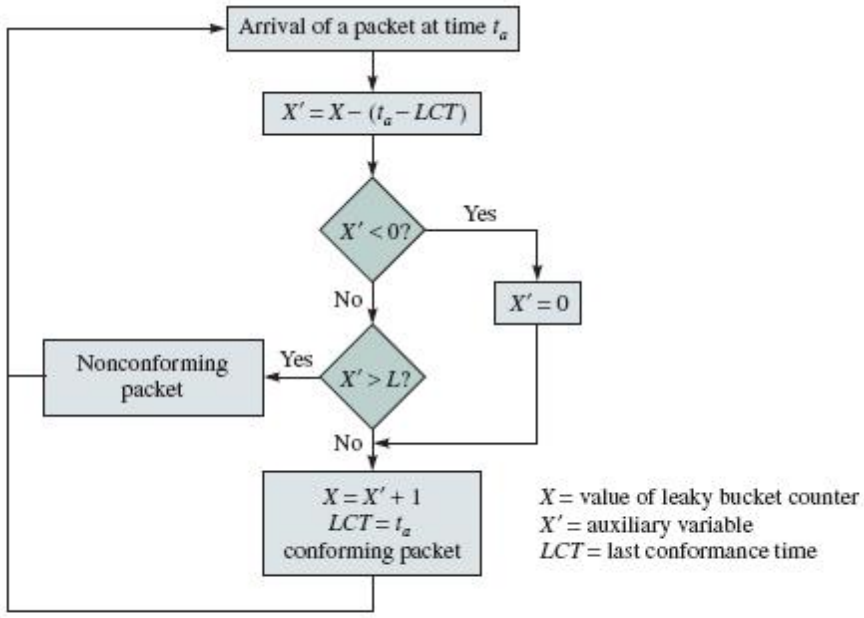


FIGURE 7.54 Leaky bucket algorithm used for policing

- The above figure shows the leaky bucket algorithm that can be used to police the traffic flow.
- At the arrival of the first packet, the content of the bucket is set to zero and the last conforming time (LCT) is set to the arrival time of the first packet.
- The depth of the bucket is L+I, where I depends on the traffic burstiness.
- At the arrival of the kth packet, the auxiliary variable X' records the difference between the bucket content at the arrival of the last conforming packet and the interarrival time between the last conforming packet and the kth packet.
- If the auxiliary variable is greater than L, the packet is considered as nonconforming, otherwise the packet is conforming. The bucket content and the arrival time of the packet are then updated.

Leaky Bucket Example: - The operation of the leaky bucket algorithm is illustrated in the below figure.

- Here the value I is four packet times, and the value of L is 6 packet times.
- The arrival of the first packet increases the bucket content by four (packet times).
- At the second arrival the content has decreased to three, but four more are added to the bucket resulting in total of seven.
- The fifth packet is declared as nonconforming since it would increase the content to 11, which would exceed L+I (10).
- Packets 7, 8, 9 and 10 arrive back to back after the bucket becomes empty. Packets 7, 8 and 9 are conforming, and the last one is nonconforming.
- Non-conforming packets not allowed into bucket & hence not included in calculations.

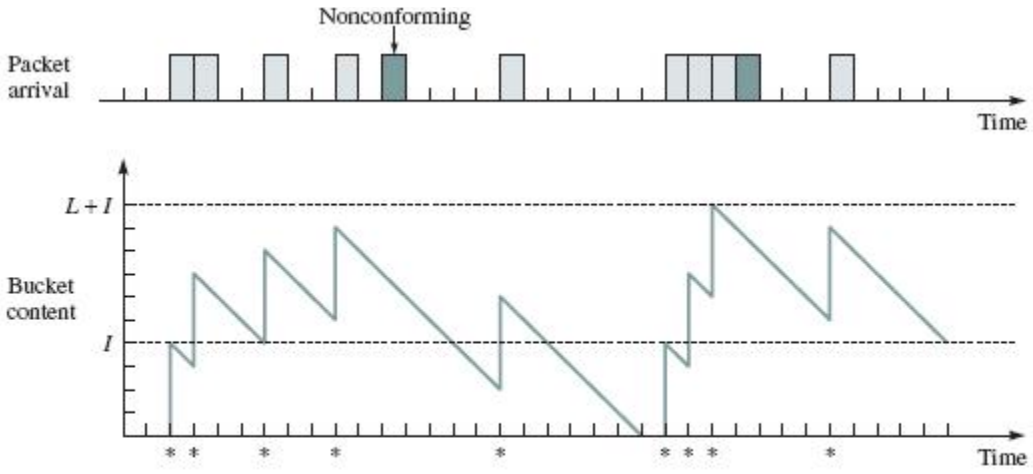


FIGURE 7.55 Behavior of leaky bucket

Dual Leaky Bucket

- Dual leaky bucket is use to police multiple traffic parameters like PCR, SCR, and MBS:
- Traffic is first checked for SCR at first leaky bucket.
- Nonconforming packets at first bucket are dropped or tagged.
- Conforming (untagged) packets from first bucket are then checked for PCR at second bucket.
- Nonconforming packets at second bucket are dropped or tagged.

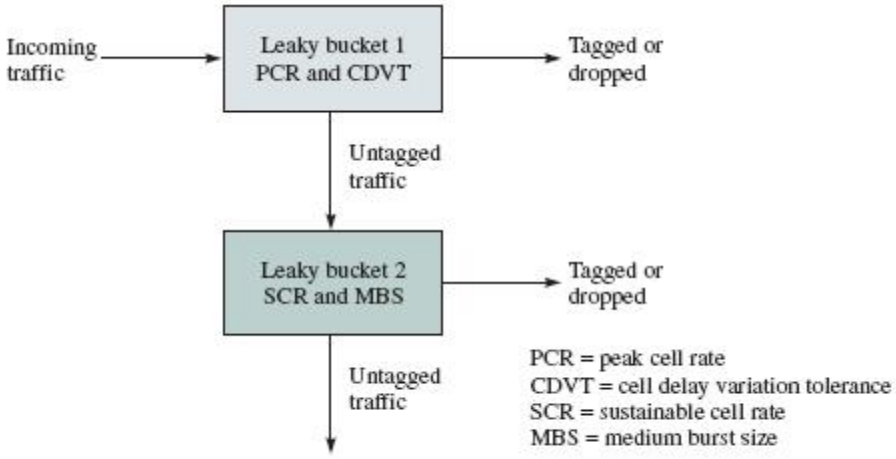
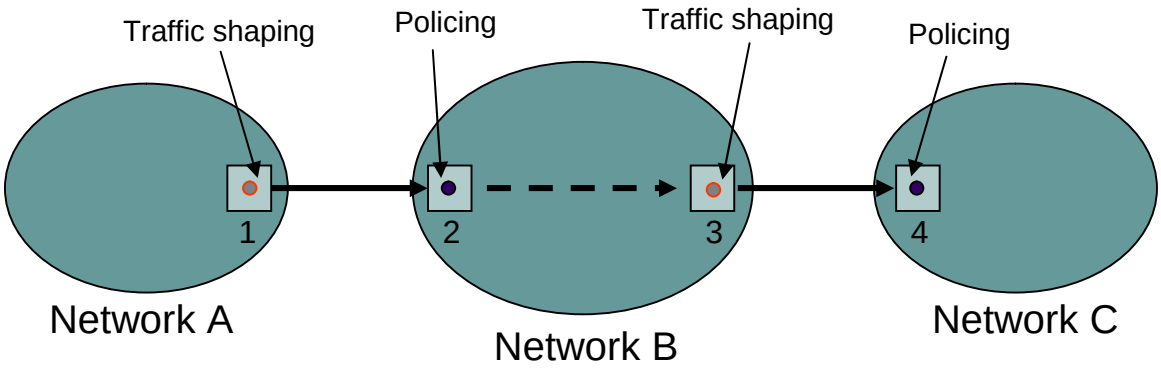


FIGURE 7.57 A dual leaky bucket configuration

Traffic Shaping



- Networks police the incoming traffic flow
- Traffic shaping is used to ensure that a packet stream conforms to specific parameters
- Networks can shape their traffic prior to passing it to another network
- In the above figure, the traffic shaping device is located at the node just before the traffic flow leaves a network, while the policing device is located at the node that receives the traffic flow from another network.

Leaky Bucket Traffic Shaper

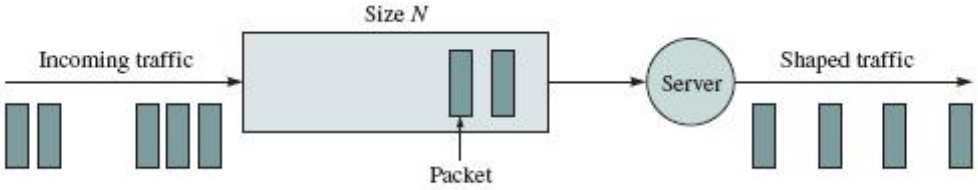


FIGURE 7.59 A leaky bucket traffic shaper

- Incoming packets are first stored in a buffer.
- Packets are served periodically so that the stream of packets at the output is smooth.
- Incoming packets are first stored in a buffer.
- Packets are served periodically so that the stream of packets at the output is smooth.
- A traffic shaping device needs to introduce certain delays for packets that arrive earlier than their scheduled departures and require a buffer to store these packets.
- Leaky bucket traffic shaper is too restrictive, since the output rate is constant when the buffer is not empty.

Token Bucket Traffic Shaper

- Token bucket is a simple extension of leaky bucket traffics shaper
- Tokens are generated periodically at constant rate and are stored in token bucket.
- Token rate regulates transfer of packets.
- If the 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 sufficient tokens available, packets enter network without delay
- If the token bucket is empty, arriving packets have to wait in the packet buffer.
- The size K determines how much burstiness allowed into the network

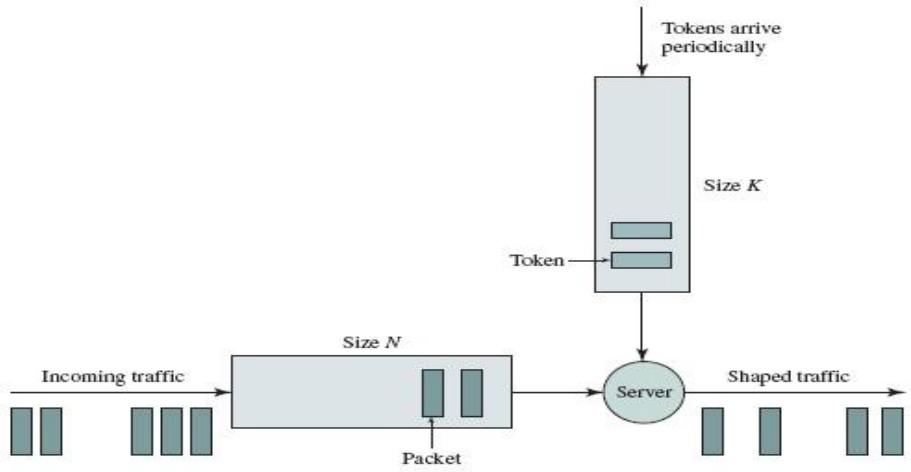


FIGURE 7.60 Token bucket traffic shaper

Closed-Loop Flow Control

- Congestion control
  - Feedback information is used to regulate the flow from sources into network based on buffer content, link utilization, etc.
  - Examples: TCP at transport layer; congestion control at ATM level
- Feedback information may be sent by End-to-end or Hop-by-hop.

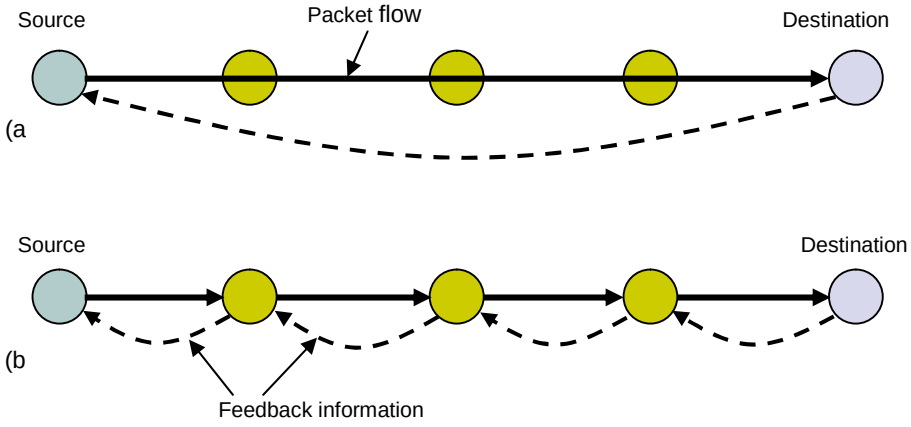
End-to-end closed loop control

- Feedback information about state of network is propagated back to source which regulate packet flow rate.
- Feedback information may be forwarded directly by a node that detects congestion, or it may be forwarded to destination first which then it relays information to source.
- The transmission of feedback information introduces propagation delay, so the information may not be accurate when the source receives the information.

Hop-by-hop control

- It reacts faster than end-to-end counterpart due to shorter propagation delay.
- State of the network is propagated to the upstream node as shown in below figure.
- When a node detects congestion it tells to its upstream neighbor to slow down its transmission rate.
- The Back Pressure created from one down stream node to another upstream node may continue all the way to the source.

End-to-End vs. Hop-by-Hop Congestion Control



Implicit vs. Explicit Feedback: - The information can be implicit or explicit.

Explicit Feedback

- The node detecting congestion initiates an explicit message to notify the source about the congestion in the network.
- The explicit message can be sent as separate packet often called as choke packets or piggybacked on a data packet.
- The explicit message may be bit information or it may contain rich amount of information.

Implicit Feedback

- In implicit Feedback, no such explicit messages are sent between the nodes.
- Here congestion is controlled by using time out based on missing acknowledgements from destination to decide whether congestion has been encountered in the network.
- TCP congestion control is one example that regulates the transmission rate by using the implicit feedback information derived from missing acknowledgement.

Traffic Management at the flow aggregated level / Traffic Engineering

- Routing of aggregate traffic flows across the network for efficient utilization of resources and meeting of service levels
- Traffic Management at the Flow-Aggregate Level is called “Traffic Engineering”.
- Management exerted at flow aggregate level
- Distribution of flows in network to achieve efficient utilization of resources (bandwidth)
- Shortest path algorithm to route a given flow not enough
    - Does not take into account requirements of a flow, e.g. bandwidth requirement
    - Does not take account interplay between different flows
- Must take into account aggregate demand from all flows.
- Refer figure 7.63 and page number 560-561 for more information.

Why Internetworking?

- To build a "network of networks" or internet
  - operating over multiple, coexisting, different network technologies
  - providing ubiquitous(universal) connectivity through IP packet transfer
  - achieving huge economies of scale
- To provide universal communication services
  - independent of underlying network technologies
  - providing common interface to user applications
- To provide distributed applications
  - Rapid deployment of new applications
    - Email, WWW, Peer-to-peer
  - Application independent of network technologies
    - New networks can be introduced

TCP / IP Architecture

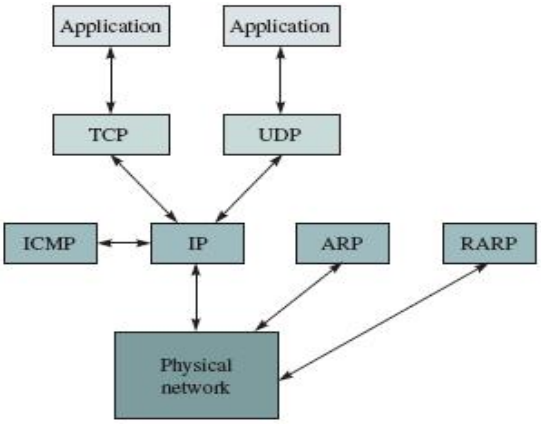


FIGURE 8.1 TCP/IP protocol suite

- The TCP/IP protocol suite usually refers not only to the two most well-known protocols called TCP and IP but also to other related protocols such as UDP, ICMP, HTTP, TELNET and FTP.
- Basic structure of TCP/IP protocol suite is shown in above figure.
- Protocol data unit (PDU) exchanged between peer TCP protocols is called segments.
- Protocol data unit (PDU) exchanged between peer UDP protocols is called datagrams.
- Protocol data unit (PDU) exchanged between peer IP protocols is called packets.

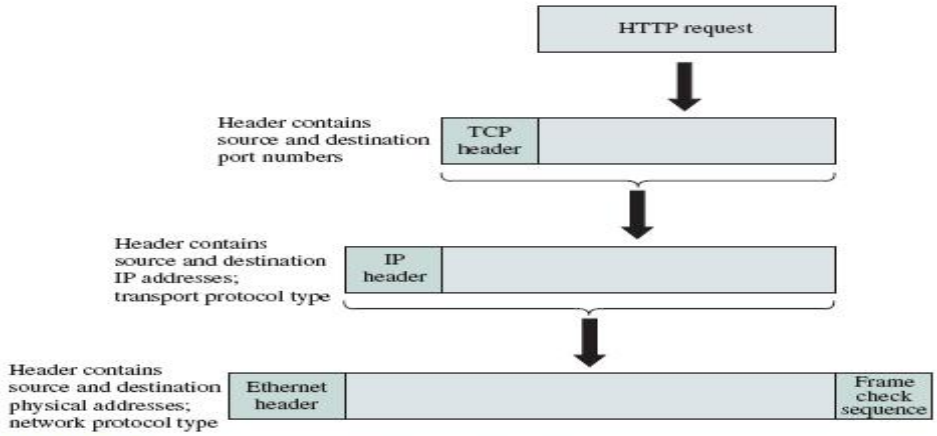


FIGURE 8.2 Encapsulation of PDUs in TCP/IP and addressing information in the headers

- In the above figure an HTTP GET command is passed to the TCP layer, which encapsulates the message into a TCP segment.
- The segment header contains an ephemeral port number for the client process and well known port 80 for HTTP server process.
- The TCP segment is passed to IP layer where it is encapsulated in an IP packet.
- The IP packet contains source and destination network address.
- IP packet is then passed through network interface and encapsulated into PDU of underlying network.
- In the network interface, the IP packet is encapsulated into an Ethernet frame, which contains physical addresses that identify the physical endpoints for the Ethernet sender and receiver.

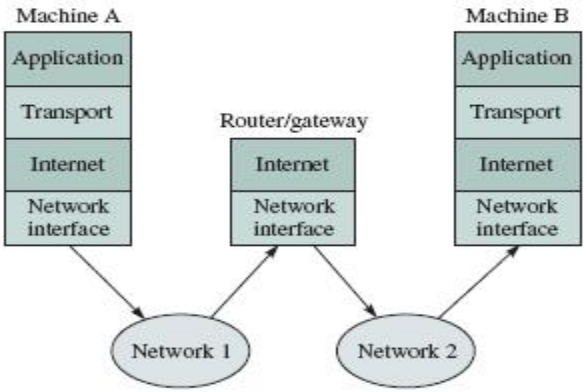


FIGURE 8.3 The Internet and network interface layers

- IP packets transfer information across Internet
- Host A IP → router→ router…→ router→ Host B IP
- IP layer in each router determines next hop (router)
- Network interfaces transfer IP packets across networks

- Internet Names
  - Each host has a unique name
    - Independent of physical location
    - Domain Name will facilitates memorization by humans
  - Host Name
    - Name given to host computer
  - User Name
    - Name assigned to user

Internet Addresses

- Each host has globally unique logical 32 bit IP address
- Separate address for each physical connection to a network
- Routing decision is done based on destination IP address
- IP address has two parts:
  - netid and hostid
  - netid unique
  - netid facilitates routing
- Dotted Decimal Notation is used for representation:
  Ex: - int1.int2.int3.int4
  128.100.10.13
  DNS(Domain Name Service) resolves IP name to IP address

Physical Addresses

- LANs (and other networks) assign physical addresses to the physical attachment to the network
- The network uses its own address to transfer packets or frames to the appropriate destination
- IP address needs to be resolved to physical address at each IP network interface
- Example: Ethernet uses 48-bit addresses
    - Each Ethernet network interface card (NIC) has globally unique Medium Access Control (MAC) or physical address
    - First 24 bits identify NIC manufacturer; second 24 bits are serial number
    - 00:90:27:96:68:07 12 hex numbers

Internet Protocol

- It provides best effort, connectionless packet delivery, packets may be lost, out of order, or even duplicated, so it is the responsibility of higher layer protocols to deal with these, if necessary.
- The header is of fixed-length component of 20 bytes plus variable-length consisting of options that can be up to 40 bytes.

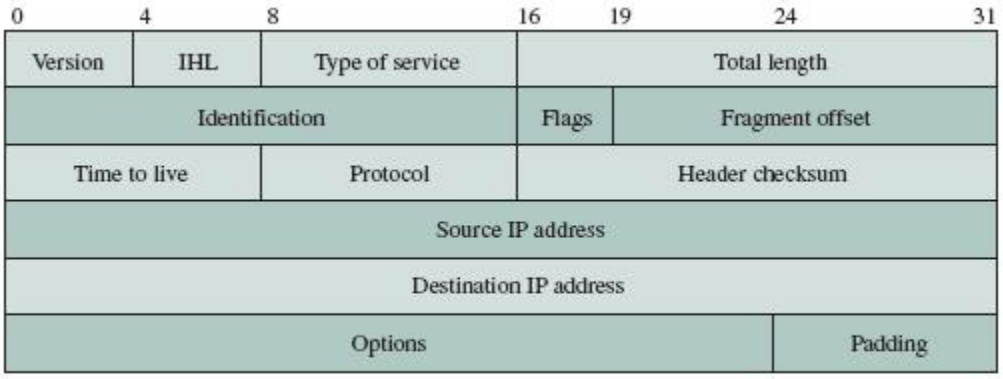


FIGURE 8.4 IP version 4 header

Version: This field identifies the current IP version and it is 4.
Internet header length (IHL): It specifies the length of the header in 32-bit words. If no options are used, IHL will have value of 5.
Type of service (TOS): This field specifies the priority of packet based on delay, throughput, reliability and cost. Three bits are used to assign priority levels and four bits are used for specific requirement (i.e. delay, throughput, reliability and cost).
Total length: The total length specifies the number of bytes of the IP packet including header and data, maximum length is 65535 bytes.
Identification, Flags, and Fragment Offset: These fields are used for fragmentation and reassembly.
Time to live (TTL): It specifies the number of hops; the packet is allowed to traverse in the network. Each router along the path to the destination decrements this value by one. If the value reaches zero before the packet reaches the destination, the router discards the packet and sends an error message back to the source.
Protocol: specifies upper-layer protocol that is to receive IP data at the destination. Examples include TCP (protocol = 6), UDP (protocol = 17), and ICMP (protocol = 1).

Header checksum: verifies the integrity of the IP header of the IP packet.

- IP header uses check bits to detect errors in the header
- A checksum is calculated for header contents
- Checksum recalculated at every router, so algorithm selected for ease of implementation in software

Source IP address and destination IP address: contain the addresses of the source and destination hosts.
Options: Variable length field allows packet to request special features such as security level, route to be taken by the packet, and timestamp at each router. Detailed descriptions of these options can be found in [RFC 791].
Padding: This field is used to make the header a multiple of 32-bit words.

IP Header Processing

1. Compute header checksum for correctness and check that fields in header (e.g. version and total length) contain valid values
2. Consult routing table to determine next hop
3. Change fields that require updating (TTL, header checksum)

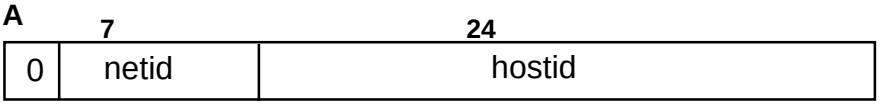
IP Addressing

- RFC 1166
- Each host on Internet has unique 32 bit IP address
- Each address has two parts: Netid and Hostid
- Netid is unique & administered by
    - American Registry for Internet Numbers (ARIN)
    - Reseaux IP Europeens (RIPE)
    - Asia Pacific Network Information Centre (APNIC)
- The Net ID identifies the network the host is connected to.
- The host ID identifies each individual system connected to network.
- Dotted Decimal Notation is used for representation:
- The IP address of 10000000 10000111 01000100 00000101 is 128.135.68.5 in dotted-decimal notation

Classful IP Addresses

- The IP address structure is divided into five address classes: Class A, Class B, Class C, Class D and Class E
- The class is identified by the Most Significant Bit (MSB) of the address as shown below.
- Class A has 7 bits for network IDs and 24 bits for host IDs, allowing up to 126 networks and about 16 million hosts per network.
- Class B has 14 bits for network IDs and 16 bits for host IDs, allowing about 16,000 networks and about 64,000 hosts per network.
- Class C has 21 bits for network IDs and 8 bits for host IDs, allowing about 2 million networks and 254 hosts per network.
- Class D addresses is used for multicast services that allow host to send information to a group of hosts simultaneously.
- Class E addresses are reserved for experiments.

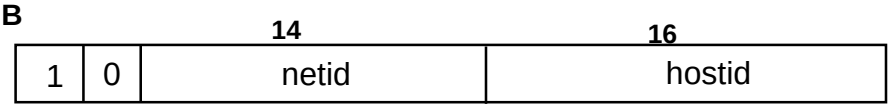
Class A



- 126 networks with up to 16 million hosts

1.0.0.0 to 127.255.255.255

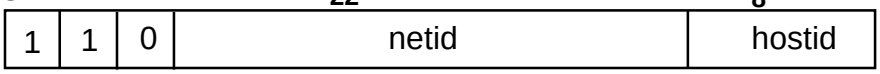
Class B



- 16,382 networks with up to 64,000 hosts

128.0.0.0 to 191.255.255.255

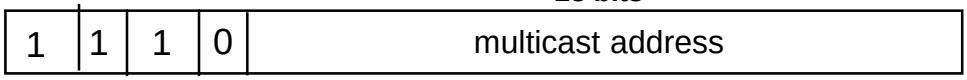
Class C



- 2 million networks with up to 254 hosts

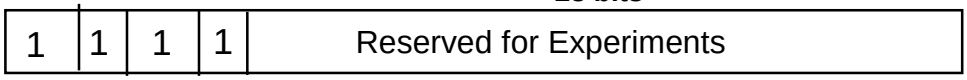
192.0.0.0 to 223.255.255.255

Class D



224.0.0.0 to 239.255.255.255

Class E



240.0.0.0 to 254.255.255.255

Subnet Addressing

- Subnet addressing was introduced in the mid 1980s when most large organizations are moving their computing platforms from mainframes to networks of workstations.
- Subnetting adds another level of hierarchical level called "Subnet".
- Inside the organization the network administrator can choose any combination of lengths for subnet and host ID fields.
- Example: - consider an organization that has been assigned a class B IP address with a network ID of 150.100. Suppose the organization has many LANS, each consisting of not more than 100 hosts. Then seven bits are sufficient to uniquely identify each host in a subnetwork. The other nine bits can be used to identify the subnetworks within organization
- To find the subnet number, the router needs to store an additional quantity called subnet mask, which consists of binary 1s for every bit position of the address except the host ID field where binary 0s are used.
- For the IP address 150.100.12.176, the subnet mask is 11111111 11111111 11111111 10000000, which corresponds to 255.255.255.128.

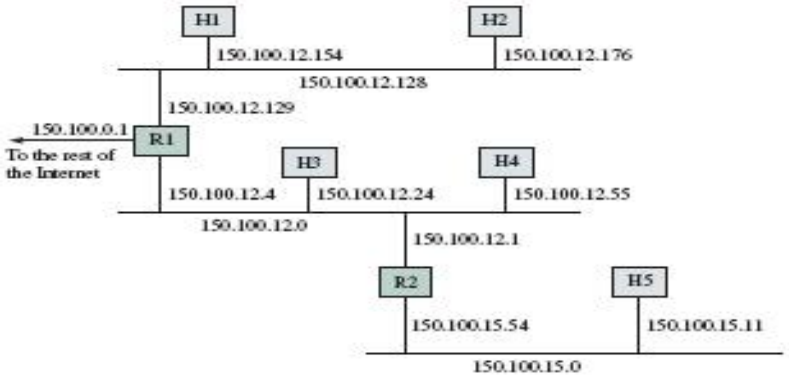
- The router can determine the subnet number by performing a binary AND between subnet mask and the IP address.

The IP address is 10010110 01100100 00001100 10110000 i.e. 150.100.12.176 AND with subnet mask 11111111 11111111 11111111 10000000 i.e. 255.255.255.128 to get subnet number 10010110 01100100 00001100 10000000 i.e.150.100.12.128 and which is also called as First Address and is used to identify the subnetwork inside the organization.

- The IP address 150.100.12.255 is used to broadcast packets inside the subnetwork. Thus the host connected to subnetwork must have IP address in the range 150.100.12.129 to 150.100.12.254.

IP Routing

- IP layer in end-system hosts and in the router work together to route packets from source to destination.
- IP layer in each host and router maintains a routing table, which is used to route the packets based on IP address.
- If a destination host is directly connected to the originating host by a link or by a LAN, then the packet is sent directly to destination host using appropriate network interface, otherwise, the routing table specifies that the packet is to send to default gateway.
- When a router receives an IP packet from one of the network interfaces, then router examines its routing table to see whether the packet is destined to itself or not, if so, delivers to router's own address, then the router determines the next–hop router and associated network interface, and then forwards the packet.
- Each row in routing table must provide information like: destination IP address, IP address of next-hop router, several flag fields, outgoing network interface, and other information such as subnet mask, physical address.
- H flag indicates whether the route in the given row is to a host (H=1) or to a network.
- G flag indicates whether the route in the given row is to a router (gateway, G=1) or to a directly connected destination (G=0).
- Each time a packet is to be routed, the routing table is searched in the following order.
- First, the destination column is searched to see whether table contains an entry for complete destination IP address.
- If so, then IP packet is forwarded according to next-hop entry and G flag.
- Second, if the table does not contain complete destination IP address, then routing table is searched for the destination network ID.
- If an entry found, the IP packet is forwarded according to next-hop entry and G flag.
- Third, if table does not contain destination network ID, the table is searched for default router entry, and if one is available, the packet is forwarded there.
- Finally if none of the above searches are successful, the packet is declared undeliverable and an ICMP "host unreachable error" packet is sent back to originating host.



Example—Routing with Subnetworks

Suppose that host H5 wishes to send an IP packet to host H2 in Figure 8.7. H2 has IP address 150.100.12.176 (1001 0110. 0110 0100. 0000 1100. 1011 0110). Let us trace the operations in carrying out this task.

The routing table in H5 may look something like this:

Destination	Next-Hop	Flags	Network Interface
127.0.0.1	127.0.0.1	H	lo0
default	150.100.15.54	G	emd0
150.100.15.0	150.100.15.11		emd0

The first entry is the loopback interface, the H indicates a host address, and `lo0` by convention is always the loopback interface. The second entry is the default entry, with next-hop router R2 (150.100.15.54), which is a router, so G = 1, and with Ethernet interface `emd0`. The third entry does not have H set, so it is a network address; G is also not set, so a direct route is indicated and the next-hop entry is the IP address of the outgoing network interface.

CIDR

- CIDR stands for Classless Inter-Domain Routing.
- CIDR was developed in the 1990s as a standard scheme for routing network traffic across the Internet.
- Before CIDR technology was developed, Internet routers managed network traffic based on the class of IP addresses. In this system, the value of an IP address determines its subnetwork for the purposes of routing.
- CIDR is an alternative to traditional IP subnetting that organizes IP addresses into subnetworks independent of the value of the addresses themselves. CIDR is also known as supernetting as it effectively allows multiple subnets to be grouped together for network routing.

CIDR Notation: - CIDR specifies an IP address range using a combination of an IP address and its associated network mask. CIDR notation uses the following format -

xxx.xxx.xxx.xxx/n

where n is the number of (leftmost) '1' bits in the mask. For example,

192.168.12.0/23 applies the network mask 255.255.254.0 to the 192.168 network, starting at 192.168.12.0. This notation represents the address range 192.168.12.0 - 192.168.13.255. Compared to traditional class-based networking, 192.168.12.0/23 represents an aggregation of the two Class C subnets 192.168.12.0 and 192.168.13.0 each having a subnet mask of 255.255.255.0. In other words,

192.168.12.0/23 = 192.168.12.0/24 + 192.168.13.0/24

Additionally, CIDR supports Internet address allocation and message routing independent of the traditional class of a given IP address range. For example,

10.4.12.0/22 represents the address range 10.4.12.0 - 10.4.15.255 (network mask 255.255.252.0). This allocates the equivalent of four Class C networks within the much larger Class A space.

You will sometimes see CIDR notation used even for non-CIDR networks. In non-CIDR IP subnetting, however, the value of n is restricted to either 8 (Class A), 16 (Class B) or 24 (Class C). Examples:

- 10.0.0.0/8
- 172.16.0.0/16
- 192.168.3.0/24

CIDR aggregation requires the network segments involved to be contiguous (numerically adjacent) in the address space. CIDR cannot, for example, aggregate 192.168.12.0 and 192.168.15.0 into a single route unless the intermediate .13 and .14 address ranges are included (i.e., the 192.168.12/22 network).

ARP (Address Resolution Protocol)

- The address resolution protocol (ARP) is a protocol used by the Internet Protocol (IP) specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol.
- The protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer. It is used when IPv4 is used over Ethernet.
- It is also used for IP over other LAN technologies, such as Token Ring, FDDI, or IEEE 802.11, and for IP over ATM.
- ARP is a Link Layer protocol because it only operates on the local area network or point-to-point link that a host is connected to.
- The hardware address is also known as the Medium Access Control (MAC) address, in reference to the standards which define Ethernet.
- The Ethernet address is a link layer address and is dependent on the interface card which is used.
- IP operates at the network layer and is not concerned with the link addresses of individual nodes which are to be used. The ARP is therefore used to translate IP addresses into MAC address.

- In the below figure suppose host H1 wants to send an IP packet to H3, but does not know the MAC address of H3. H1 first broadcast an ARP request packet asking the destination host, which is identified by H3's IP address, to reply. All hosts in the network receive the packet, but only the intended host, which is H3, responds to H1.
- The ARP response packet contains H3's MAC address and IP addresses.
- H1 caches H3's MAC address in its ARP table so that H1 can simply look up H3's MAC address in the table for future use.

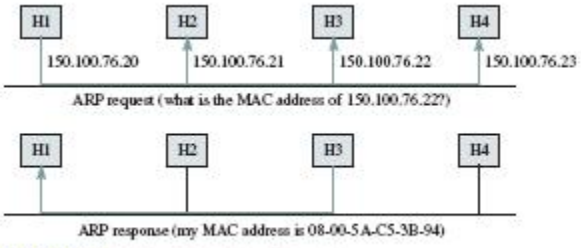


FIGURE 8.8 Address Resolution Protocol

- The ARP client and server processes operate on all computers using IP over Ethernet. The processes are normally implemented as part of the software driver that drives the network interface card.

RARP (Reverse Address Resolution Protocol)

- RARP is a link layer networking protocol, used to resolve an IP address from a given hardware address (such as an Ethernet address).
- RARP requires one or more server hosts to maintain a database of mappings from Link Layer address to protocol address.
- To obtain its IP address, the host broadcasts an RARP request packet containing its MAC address on the network.
- All hosts in the network receive the packet, but only the server replies to the host by sending an RARP response containing the host's MAC and IP address.

IP fragmentation and Reassembly

- The Internet Protocol allows IP fragmentation so that datagrams can be fragmented into pieces small enough to pass over a link with a smaller MTU than the original datagram size.
- The Identification field, and Fragment offset field along with Don't Fragment and More Fragment Flags are used for Fragmentation and Reassembly of IP datagrams.
- In a case where a router in the network receives a PDU larger than the next hop's MTU, it has two options. Drop the PDU and send an ICMP message which says "Packet too Big", or to Fragment the IP packet and send over the link with a smaller MTU.
- If a receiving host receives an IP packet which is fragmented, it has to reassemble the IP packet and hand it over to the higher layer.
- Reassembly is intended to happen in the receiving host but in practice it may be done by an intermediate router, for example network address translation requires re-calculating checksums across entire packets, and so routers supporting this will often recombine packets as part of the process.

- The details of the fragmentation mechanism, as well as the overall architectural approach to fragmentation, are different in IPv4 and IPv6.
- In IPv4, routers do the fragmentation, whereas in IPv6, routers do not fragment, but drop the packets that are larger than the MTU size. Though the header formats are different for IPv4 and IPv6, similar fields are used for fragmentation, so the algorithm can be reused for fragmentation and reassembly.
- IP fragmentation can cause excessive retransmissions when fragments encounter packet loss and reliable protocols such as TCP must retransmit all of the fragments in order to recover from the loss of a single fragment.
- Thus senders typically use two approaches to decide the size of IP datagrams to send over the network.
- The first is for the sending host to send an IP datagram of size equal to the MTU of the first hop of the source destination pair.
- The second is to run the "Path MTU discovery" algorithm, to determine the path MTU between two IP hosts, so that IP fragmentation can be avoided.
- The flag field has three bits, one unused bit, one “don’t fragment”(DF) bit, and one “more fragment”(MF) bit.
- If DF bit is set to 1, it forces the router not to fragment the packet. If the packet length is greater than MTU, the router will discard the packet and send an error message to the source host.
- The MF bit tells the destination host whether or not more fragments follow. If there are more, the MF bit is set to 1; otherwise, it is set to 0.
- Fragment offset field identifies the location of a fragment in a packet.

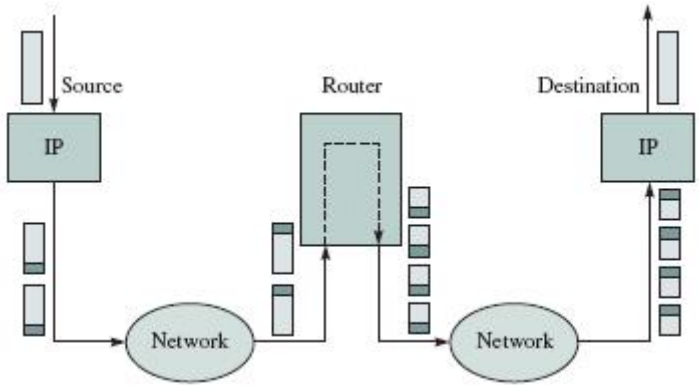


Figure: Packet fragmentation

Example—Fragmenting a Packet

Suppose a packet arrives at a router and is to be forwarded to an X.25 network having an MTU of 576 bytes. The packet has an IP header of 20 bytes and a data part of 1484 bytes. Perform fragmentation and include the pertinent values of the IP header of the original packet and of each fragment.

The maximum possible data length per fragment $= 576 - 20 = 556$ bytes. However, 556 is not a multiple of 8. Thus we need to set the maximum data length to 552 bytes. We can break 1484 into $552 + 552 + 380$ (other combinations are also possible).

Table 8.1 shows the pertinent values for the IP header where x denotes a unique identification value. Other values, except the header checksum, are the same as in the original packet.

	Total length	ID	MF	Fragment offset
Original packet	1504	x	0	0
Fragment 1	572	x	1	0
Fragment 2	572	x	1	69
Fragment 3	400	x	0	138

TABLE 8.1 Values of the IP header in a fragmented packet

Deficiencies of IP

- Lack of error control, flow control and congestion control
- Lack of assistance mechanisms

What happens if something goes wrong?

- If a router must discard a datagram because it can not find a router to the final destination
- The time-to-leave field has a zero value
- If the final destination host must discard all fragments of a datagram because it has not received all fragments within a pre-determined time limit

IP has no built in mechanisms to notify the original hosts, in erroneous situations
IP also lacks a mechanism for host and management queries
• A host wants to know whether a router or another host is active
• Sometimes network manager needs information from another host or router

Internet Control Message Protocol [ICMP] is companion to IP, designed to compensate these deficiencies

- ICMP is a network layer protocol
- Its messages are encapsulated inside IP datagrams before going to lower layer
- Ping and Traceroute uses ICMP messages,

ICMP Messages

1) Error Reporting Messages
2) Query Messages

1) Error Reporting
    - Destination unreachable
    - Source quench

- Time exceeded
- Parameters problems
- Redirection

ICMP messages [Error reporting]

1. Destination unreachable

When the subnet or a router can not locate the destination

Or

When a packet with DF bit, can not be delivered because a 'small-packet' network stands in the way

2. Time exceeded

When a packet is dropped because its counter has reached zero. This event is a symptom that packets are looping enormous congestion or the time values are being set too low.

3. Parameter problem

Indicates that an illegal value has been detected in the header field

Indicates a bug in the sending host's IP software Or Possibly in the software of a router transited.

4. Source quench

To throttle hosts that send too many packets, When a host receives this message, it slows down sending packets

5. Redirect

Is used when a router notices that a packet seems to be routed wrong

It is used by the router to tell the sending host about the probable error.

2) Query Messages
   - Echo request and reply
   - Time-stamp request and reply
   - Address mask request and reply

1. ECHO & ECHO Reply

To see if a given destination is reachable and alive, upon receipt of ECHO message, the destination is expected to send an ECHO REPLY message back.

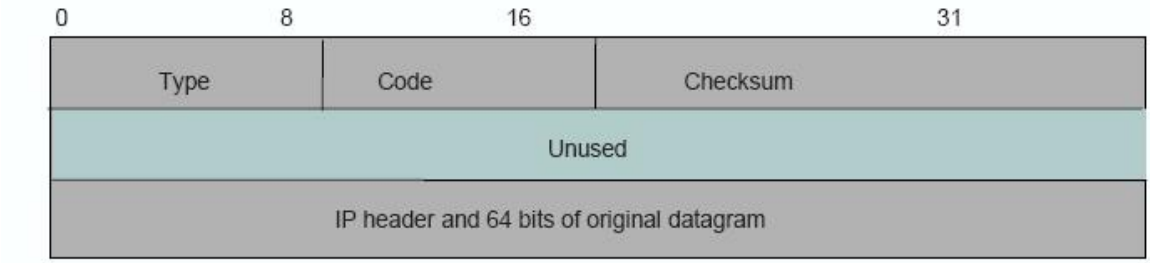
2. Time stamp & Time stamp reply

Similar to ECHO queries, except that the arrival time of the message and departure time of the reply are recorded in the reply.

This facility is used to measure network performance.

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

ICMP Basic Error Message Format



Type of message: some examples
0 Network Unreachable; 3 Port Unreachable
1 Host Unreachable 4 Fragmentation needed
2 Protocol Unreachable 5 Source route failed
11 Time-exceeded,
code=0 if TTL exceeded
• Code: purpose of message
• IP header & 64 bits of original datagram
– To match ICMP message with original data in IP packet

Echo Request & Echo Reply Message Format

Echo request: type=8; Echo reply: type=0
– Destination replies with echo reply by copying data in request onto reply message
• Sequence number to match reply to request
• ID to distinguish between different sessions using echo services
• Used in PING

ICMP functions

1) Announce network errors: Such as host or Entire portion of the network being unreachable, due to some type of failure. A TCP or UDP packet directed at a port number with no receiver attached is also reported via ICMP.
2) Announce network congestion: When a router begins buffering too many packets, due to an inability to transmit them as fast as they are being received, it will generate ICMP Source Quench messages. Directed at the sender, these messages should cause the rate of packet transmission to be slowed.
3) Assist Troubleshooting: ICMP supports an Echo function, which just sends a packet on a round--trip between two hosts. Ping, a common network management tool, is based on this feature. Ping will transmit a series of packets, measuring average round—trip times and computing loss percentages.
4) Announce Timeouts: If an IP packet's TTL field drops to zero, the router discarding the packet will often generate an ICMP packet announcing this fact.

UNIT 2 Question Bank

1. Consider a packet-by-packet fair queuing system with three logical buffers and with a service rate of one unit / second. Show the sequence of transmissions for this system for the following packet arrival pattern:
   (i) Buffer 1: arrival at time t=0, length =2 ;arrival at t=4, length= 1
   (ii) Buffer 2: arrival at time t=1, length =3 ;arrival at t=2, length= 1
   (iii) Buffer 3: arrival at time t=3, length =5 ;
   (Jan 10, 10M)
2. With a neat diagram explain the internal network operation of the network.
3. What is congestion? Discuss the general principles of congestion control? (Aug 06, 10M)
4. Explain the random early detection. (Feb 06, 6M)
5. Explain leaky bucket algorithm (Feb 05 , 8M) (July 09, 8M)
6. Explain the Token bucket policy for the traffic shaping. (July 07, 5M)
7. A computer on a 6Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 1Mbps. It is initially filled to capacity with 8 megabits. How long the computer transmit at the full 6Mbps (July 07, 5M)
8. Explain FIFO and Priority Queues for the traffic management at the packet level
9. Explain Fair Queuing for the traffic management at the packet level
10. Explain Weighted-Fair Queuing for the traffic management at the packet level
11. Write short notes on admission control and policing.
12. Write short notes on traffic shaping
13. Distinguish between end-to-end and hop-by-hop closed loop control
14. Distinguish between implicit and explicit feedback
15. Give the differences between leaky bucket and token bucket algorithm
16. Write a note on Traffic management at the flow-aggregate level.
17. Explain with diagram the TCP/IP architecture
18. Explain IPV4 header. (Feb 06, 6M) (July 09, 6M) (Jan 10, 6M)
19. Explain the IP addressing scheme. (Feb 05, 6M)
20. Distinguish between address resolution protocol and reverse address resolution protocol. (Feb 05, Aug 05, 6M) (July 07, 5M)

21. Illustrate with a diagram the five address formats used in internet (AUG 05, 6M)
22. Briefly explain Address Resolution Protocol. (July05 5M)
23. What is ICMP? Explain the functions of ICMP. (Jan 08 5M)
24. A university has 150 LANs with 100 hosts in each LAN.
    a. Suppose the university has one Class B address. Design an appropriate subnet addressing scheme.
    b. Design an appropriate CIDR addressing scheme.

    (Aug 06, 6M) (Jan 10, 6M)
25. A network on the internet has a subnet mask 255.255.240.0. What is the maximum no. of hosts it can handle? (AUG 05, 6M), (Feb 06, 6M)
26. A large number of consecutive IP address are available at 198.16.0.0. Suppose that four organizations A, B, C and D request for 4000, 2000, 4000 and 8000 addresses respectively. For each of these, give the first IP address assigned, the last IP address assigned, and the mask in dotted decimal notation.

    (Aug 06, 4M)
27. A large number of consecutive IP address are available starting at 200.40.160.0. Suppose that 3 organizations A, B, and C request for 4000, 2000 and 1000 addresses respectively. For each of these, give the first IP address assigned, the last IP address assigned, and the mask in dotted decimal notation.(Aug 09, 6M)
28. Write a note on CIDR
29. What is fragmentation? How packets are fragmented and reassembled by the IP?
30. Identify the address class of the following IP addresses: 200.58.20.165; 128.167.23.20; 16.196.128.50; 50.156.10.10; 250.10.24.96.
31. Convert the IP addresses in Problem above to their binary representation.
32. Identify the range of IPv4 addresses spanned by Class A, Class B, and Class C.
33. What are all the possible subnet masks for the Class C address space? List all the subnet masks in dotted-decimal notation, and determine the number of hosts per subnet supported for each subnet mask.
34. A host in an organization has an IP address 150.32.64.34 and a 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?
35. A small organization has a Class C address for seven networks each with 24 hosts. What is an appropriate subnet mask?
36. A packet with IP address 150.100.12.55 arrives at router R1 in Figure 8.8. Explain how the packet is delivered to the appropriate host.
37. Perform CIDR aggregation on the following /24 IP addresses: 128.56.24.0/24; 128.56.25.0./24; 128.56.26.0/24; 128.56.27.0/24.
38. Perform CIDR aggregation on the following /24 IP addresses: 200.96.86.0/24; 200.96.87.0/24; 200.96.88.0/24; 200.96.89.0/24.
39. Suppose a router receives an IP packet containing 600 data bytes and has to forward the packet to a network with maximum transmission unit of 200 bytes. Assume that the IP header is 20 bytes long. Show the fragments that the router creates and specify the relevant values in each fragment header (i.e., total length, fragment offset, and more bit).