Unit – IV

COMPUTER NETWORKS: Network Layer

The Network Layer: Network Layer Design Issues - Routing Algorithms - Congestion Control Algorithms- Quality Of Service – Internetworking

Routing

Routing packets from source to destination is the main function of the network layer. The routing algorithm is responsible for deciding which output line an incoming Packet should be transmitted on.

In Datagram subnet Routing decision must be made anew for every arriving data packet since the best route may have changed since last time.

In Virtual Circuits Subnet Routing decisions are made only when a new virtual circuit is being set up. Data packets just follow the previously established route. This is sometimes called session routing because a route remains in force for an entire user session.

Types of routing algorithms

- No adaptive (static)
    - Do not use measurements of current conditions
    - Static routes are downloaded at boot time
- Adaptive Algorithms(dynamic)
    - Change routes dynamically
        - Gather information at runtime
            - locally
            - from adjacent routers
            - from all other routers
        - Change routes
            - Every delta T seconds
            - When load changes
            - When topology changes

Optimality Principle

- If router *j* is on the optimal path from *i* to *k*, then the optimal path from *j* to *k* also falls along the same route.

Sink tree

- The set of optimal routes to a particular node forms a sink tree.
- Sink trees are not necessarily unique
- Goal of all routing algorithms
    - Discover sink trees for all destinations

Shortest Path routing algorithm

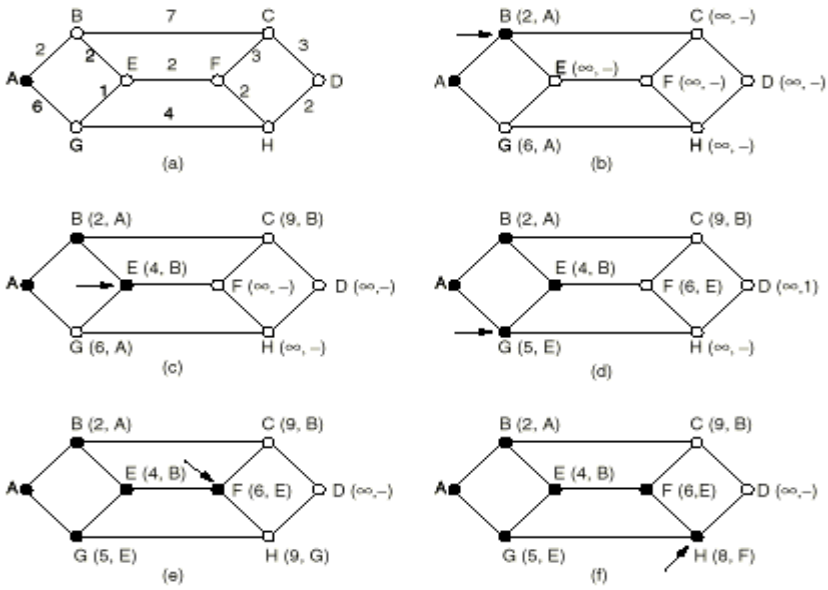
A graph of the subnet is built, with each node of the graph representing a router and each arc of the graph representing a communication line.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

Shortest path algorithm first developed by E. W. Dijkstra

Steps:

1. Mark the source node as permanent and designate it as the working node.
2. Compute the tentative distance from the source to all nodes adjacent to the working node.
3. Select the node having low value and compute tentative distance for that node. If previous is shorter than the current tentative distance replace the tentative distance of the destination and record the label of the working node there.



Flooding

Static algorithm

Every incoming packet is sent out on every outgoing line except the one it arrived on.

Generates vast numbers of duplicate packets, unless some measures are taken to damp the process.

1.Hop counter

contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Hop counter should be initialized to the length of the path from source to destination.

If the sender does not know how long the path is, the full diameter of the subnet is assigned and it is worst case.

2. Keep track of which packets have been flooded, to avoid sending them out a second time.

To achieve this the source router put a sequence number in each packet it receives from its hosts.

3. A variation of flooding **selective flooding** is used.

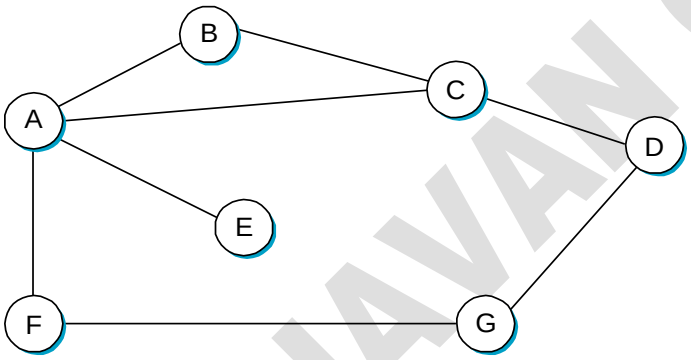
The routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.

Distance Vector Routing (an adaptive routing algorithm)

- Bellman-Ford Routing
- Ford Fulkerson Algorithm
- Original ARPANET routing algorithm
- Previously used on Internet (Routing Information Protocol)
- AppleTalk and Cisco routers use improved versions of this algorithm

No node has complete information about the costs of all network links. It is possible by gradual calculation of path by exchanging information with neighbors. Key thing is each node knows the cost of links to its neighbors

Each node constructs a one-dimensional array containing the "distances" or "costs" to all other nodes, distributes it to its immediate neighbours. If no link exists between two nodes, the cost of a direct link between the nodes is "infinity".



	A	**B**	**C**	**D**	**E**	**F**	**G**
A	**0**	**1**	**1**	∞	**1**	**1**	∞
B	**1**	**0**	**1**	∞	∞	∞	∞
C	**1**	**1**	**0**	**1**	∞	∞	∞
D	∞	∞	**1**	**0**	∞	∞	**1**
E	**1**	∞	∞	∞	**0**	∞	∞
F	**1**	∞	∞	∞	∞	**0**	**1**
G	∞	∞	∞	**1**	∞	**1**	**0**

- With this information, routing table at A is

	Cost	Next Hop
B	**1**	**B**
C	**1**	**C**
D	∞	**-**
E	**1**	**E**
F	**1**	**F**
G	∞	**-**

- Each node sends a message to neighbors with a list of distances.
- F --> A with G is at a distance 1
- C --> A with D at distance 1.

	Cost	Next Hop
B	**1**	**B**
C	**1**	**C**
D	**2**	**C**
E	**1**	**E**
F	**1**	**F**
G	**2**	**F**

Final Distance Matrix is

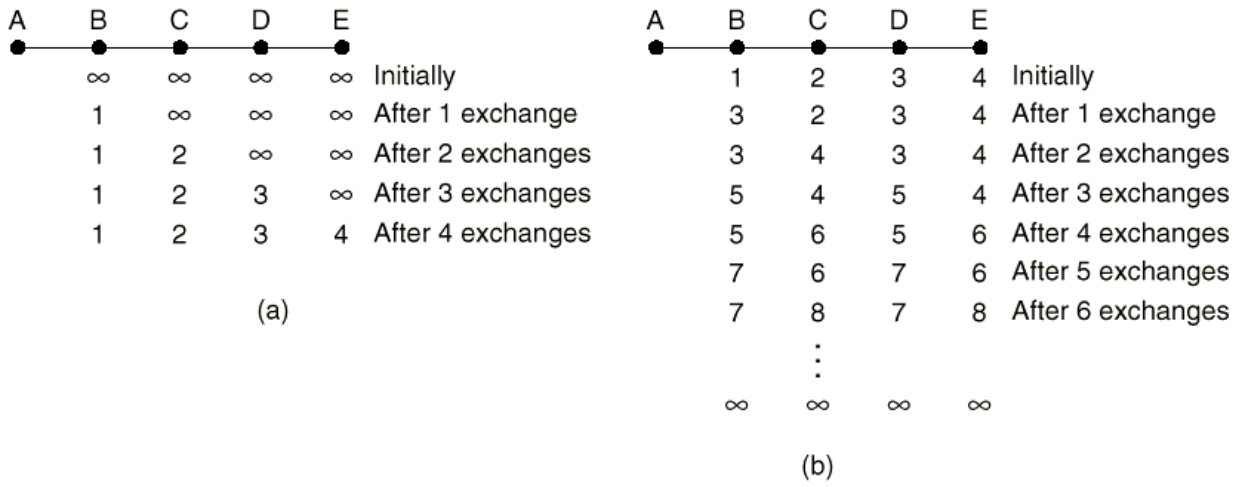
	A	B	C	D	E	F	G
A	**0**	**1**	**1**	**2**	**1**	**1**	**2**
B	**1**	**0**	**1**	**2**	**2**	**2**	**3**
C	**1**	**1**	**0**	**1**	**2**	**2**	**2**
D	**2**	**2**	**1**	**0**	**3**	**2**	**1**
E	**1**	**2**	**2**	**3**	**0**	**2**	**3**
F	**1**	**2**	**2**	**2**	**2**	**0**	**1**
G	**2**	**3**	**2**	**1**	**3**	**1**	**0**

So,

- Neighboring routers periodically exchange information from their routing tables.

- Routers replace routes in their own routing tables anytime that neighbors have found better routes.
- Information provided from neighbors
    - Outgoing line used for destination
    - Estimate of time or distance
        - can be number of hops, time delay, packet queue length, etc.

The count-to-infinity problem



Link state routing - an adaptive routing algorithm

Two primary problems caused DVR to failure.

1. It did not take line bandwidth into account when choosing routes.
2. Took too long to converge (the count-to-infinity problem).

For these reasons, it was replaced by an entirely new algorithm, now called link state routing.

Five Steps

1.) Discover your neighbors and learn their addresses.

2.) Measure the cost (delay) to each neighbor.

3.) Construct a packet containing all this information

4.) Send this packet to all other routers.

5.) Compute the shortest path to every other router.

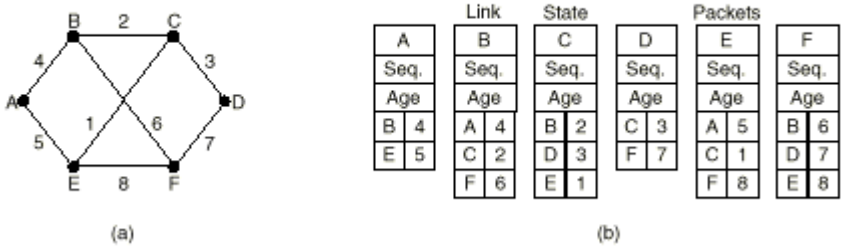
Discovering Your Neighbors

- Send "Hello" packet on each point-to-point line. Destination node replies with its address.

Measuring Line Cost

- Send an "ECHO" packet over the line.
- Destination is required to respond to "ECHO" packet immediately.
- By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay.
- An interesting issue is whether to take the load into account when measuring the delay.
- To factor the load in, the round-trip timer must be started when the ECHO packet is queued.
- To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.
- If load is ignored and only bandwidth is considered, this problem does not occur.
- Alternatively, the load can be spread over both lines, but this solution does not fully utilize the best path.

Build Link-State Packets



Building the link state packets is easy. The hard part is determining when to build them.

- build them periodically, at regular intervals.
- build them when some significant event occurs,

line or neighbor going down or coming back up again or changing its properties appreciably.

Distributing the Link State Packets

- Use selective flooding

In flooding,

- Sequence numbers prevent duplicate packets from being propagated
- Lower sequence numbers are rejected as obsolete(outdated).

Problems

1. If the sequence numbers wrap around, confusion will reign.
2. If a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet will be rejected as a duplicate.

Computing new routes

- Dijkstra's Shortest Path algorithm is used to determine the shortest path to each destination.

Hierarchical routing

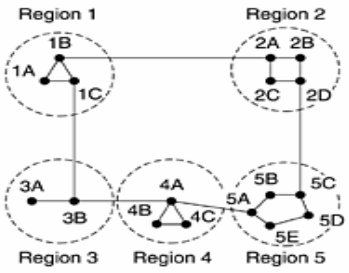
As networks grow in size, the router routing tables grow proportionally. This leads to many p**roblems as,**

- More memory is consumed by ever-increasing tables
- More CPU time is needed to scan them
- More bandwidth is needed to send status reports about them.

In hierarchical routing

- the routers are divided into regions,
- With each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.
- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on
- When routing is done hierarchically, there are entries for all the local routers and all other regions have been condensed into a single router.
- Though it save table space sometimes increase the path length.

Figure 5-15. Hierarchical routing.



(a)

Full table for 1A

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

(b)

Hierarchical table for 1A

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

(c)

Broadcast and Multicast routing

The process of sending a packet to all destinations simultaneously is called broadcasting. Various methods have been proposed for doing it.

[1] Simply send a distinct packet to each destination

[2] flooding

[3] multi-destination routing

- Each packet contains a list of destinations
- Routers duplicate packet for all matching outgoing lines

[4] spanning tree routing

- a subset of the subnet that includes all routers but contains no loops.

[5] Reverse Path Forwarding

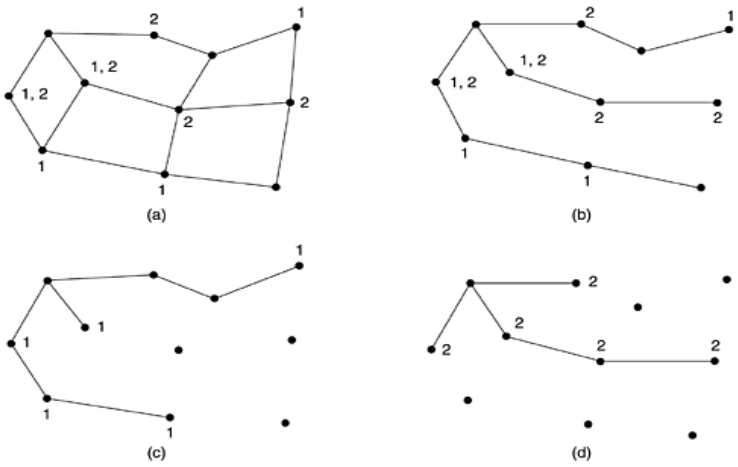
- Used When knowledge of a spanning tree is not available
- Provides an approximation of spanning tree routing
- Routers check to see if incoming packet arrives from the same line that the router uses to route outgoing packets to the broadcast source
  - If so, the router duplicates the packet on all other outgoing lines
  - Otherwise, the router discards the packet

Multicast routing

A method to broadcast packets to well-defined groups

- Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups.
- It is important that routers know which of their hosts belong to which groups.
- Either hosts must inform their routers about changes in group membership, or routers must query their hosts periodically.
- Routers tell their neighbors, so the information propagates through the subnet.

Figure 5-17. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.



Congestion control

- When too many packets are present in (a part of) the subnet, performance degrades. This situation is called congestion.
- When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered.

Factors that cause congestion

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets
- Slow processor

If the routers' CPUs are slow at performing (queuing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity.

- low-bandwidth lines

Congestion: principles

- Congestion control solutions are divided into two groups:

1. Open loop

2. Closed loop

Open loop

Open loop solutions attempt to solve the problem by good design. Make sure the problem does not occur.

- Decide when to accept traffic and discard packets and which ones
- Make scheduling decisions at various points in the network.

All of these decisions are without regard to the current state of the network.

The open loop algorithms divided into ones that

1. Act at the source

2. Act at the destination.

The closed loop solutions are based on the concept of a **feedback loop**. They are also divided into two subcategories:

1. Explicit feedback

2. Implicit feedback.

Explicit feedback algorithms

Packets are sent back from the point of congestion to warn the source.

Implicit algorithms

The source deduces(realize) the existence of congestion by making local observations, such as the time taken for acknowledgements to come back.

- Monitor the system : where and when congestion occurs? variety of metrics used are
  - % packets discarded

  - average queue length
  - number of packets that time out
  - average packet delay

- Pass collected info to places where actions can be taken = source of traffic
  - (extra) packet: The router detecting the congestion has to send a packet to the traffic source or sources, announcing the problem
  - flags (in other packets): a bit or field can be reserved in every packet for routers to fill in whenever congestion gets above some threshold level.

  When a router detects this congested state, it fills in the field in all outgoing packets, to warn the neighbors.

  - probe (query) packets: hosts or routers periodically send probe packets out to explicitly ask about congestion

- Adjust system operation
  - Increase resources: bandwidth
  - Decrease load: deny, degrade service

Congestion control in datagram subnet

- Congestion Control is concerned with efficiently using a network at high load.
- Several techniques can be employed. These include:
  - Warning bit
  - Choke packets congestion detection and recovery.
  - Load shedding
  - Random early discard congestion avoidance.

- Traffic shaping

Warning Bit

- A special bit (warning bit) in the packet header is set by the router to warn the source when congestion is detected.
- The bit is copied and piggy-backed on the ACK and sent to the sender.
- The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.
    - Algorithm at source
        - As long as warning bits arrive: reduce traffic
        - Less warning bits: increase traffic
        - Used in DecNet and Frame relay

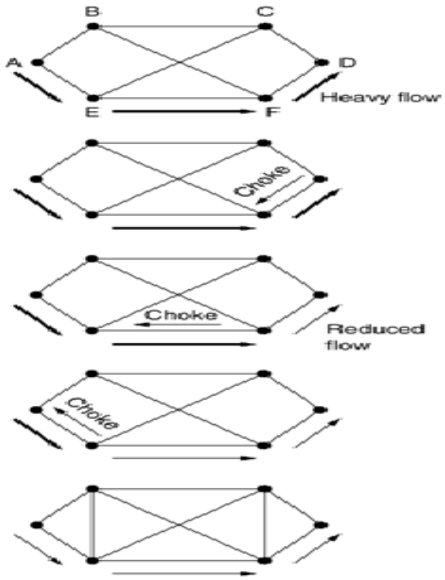
Choke Packet

- A more direct way of telling the source to slow down.
- A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
- The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
    - Host receiving choke packet
        - reduces traffic to the specified destination
        - ignores choke packets for a fixed interval
        - new choke packets during next listening interval?
            - Yes: reduce traffic

- No: increase traffic

Hop-by-hop choke packets

- Over long distances or at high speeds choke packets are not very effective.
- A more efficient method is to send to choke packets hop-by-hop.
- This requires each hop to reduce its transmission even before the choke packet arrive at the source.



Load shedding

- When routers are being inundated (swamped) by packets that they cannot handle, they just throw them away.
- When buffers become full, routers simply discard packets.
- Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
- File transfer - cannot discard older packets

It will cause a gap in the received data.

- Real-time voice or video - throw away old data and keep new packets.
- Intelligent discard policy
- Applications must mark their packets in priority classes to indicate how important they are.

Random Early Discard (RED)

- This is an approach in which the router discards one or more packets *before* the buffer becomes completely full.
- Each time a packet arrives, the RED algorithm computes the average queue length, ***avg***.
- If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.
- If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
- If *avg* is between the two thresholds, this might indicate the onset (beginning) of congestion. The probability of congestion is then calculated.

Jitter control

- The variation in the packet arrival times is called jitter.
- Important for audio and video applications.

Example

Having some packets taking 20 msec and others taking 30 msec to arrive will give an uneven quality to the sound or movie.

- The jitter can be bounded by computing the expected transit time for each hop along the path.

- When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule.
- This information is stored in the packet and updated at each hop.
    - If the packet is ahead of schedule, it is held just long enough to get it back on schedule.
    - If it is behind schedule, the router tries to get it out the door quickly.
- In video on demand, jitter is eliminated, by buffering at the receiver and then fetching data for display from the buffer and not from the network in real time.
- In applications require real-time interaction (Internet telephony and videoconferencing),the delay inherent in buffering is not acceptable.

Quality of Service (QoS)

Providing some kind of guarantees for reliability, Delay, jitter and bandwidth is called as Quality of Service. Network is engineered to provide some Quality beyond "Not to exceed B bits/s"

QoS requirements

A stream of packets from a source to a destination is called a **flow**. In connection-oriented network, all the packets belonging to a flow follow the same route and connectionless network, follow different routes.

No single technique provides efficient QoS. Practical solutions often combine multiple techniques.

1. OverProvisioning
2. Buffering
3. Traffic Shaping

    The Leaky Bucket Algorithm

    The Token Bucket Algorithm

4. Resource Reservation

5. Admission Control

6. Proportional Routing- splitting among different lines

7. Packet Scheduling

1. **Overprovisioning**

- Providing so much router capacity,

  buffer space and bandwidth.

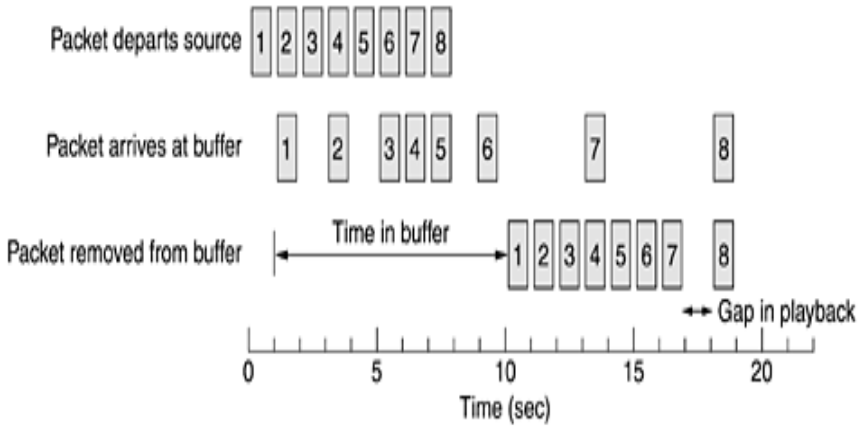
- Packets fly easily

Problem: Expensive

Buffering

Flows buffered on receiving side before being delivered. Problem: increase delay adv: smooth jitter

Figure 5-31. Smoothing the output stream by buffering packets.



Traffic Shaping

- Traffic shaping is regulating the average rate of data transmission.
- It is used when packet is emitted irregularly or not uniform.

Reason:

- Non uniform output is common if the server is handling many streams at once.
- Buffering is not always possible

  Ex: video conferencing

- Traffic shaping controls the *rate* at which packets are sent.
- Used in ATM and Integrated Services networks.
- When a connection is setup, the user and the subnet agree on a certain traffic pattern (i.e. shape) for that circuit.

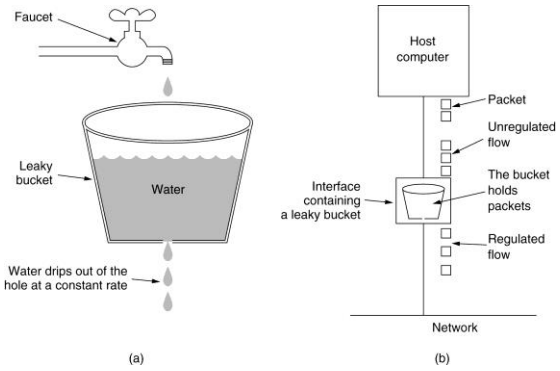
  --- service level agreement.

- It reduces congestion.
- Not important for file transfer but for real time data.
- Monitoring a traffic flow is **traffic policy**.
- Two traffic shaping algorithms are:
  - Leaky Bucket
  - Token Bucket

Leaky and token bucket algorithms

It was first proposed by Turner (1986) and is called the leaky bucket algorithm.

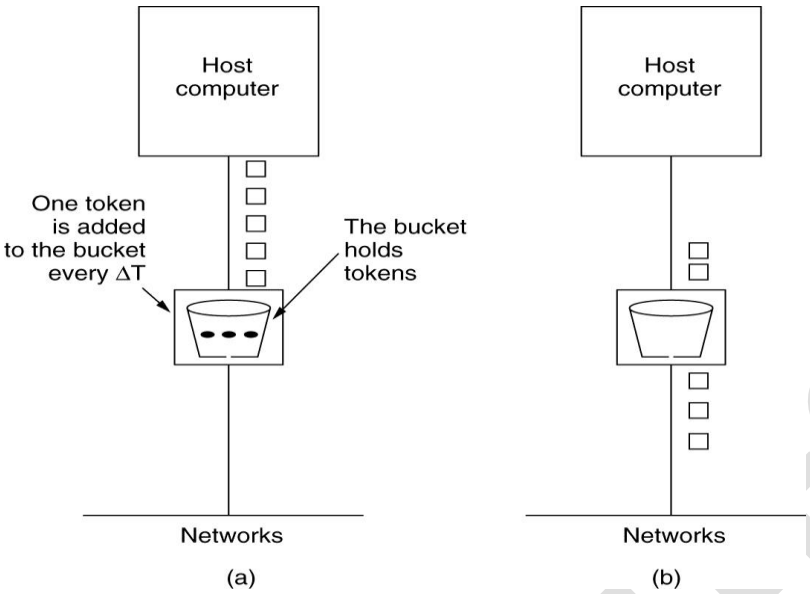
- Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue.
- If a packet arrives at the queue when it is full, the packet is discarded.



- The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input.
- The host injects one packet per clock tick onto the network.
- Thus, uniform flow of packets, smoothing out bursts and reducing congestion.

Token Bucket

- The Token Bucket Algorithm allows the output rate to vary, depending on the size of the burst.
- In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



Leaky bucket	Token bucket
Discards packets	Does not discard packets
Sends packets at an average rate.	A packet can only be transmitted if there are enough tokens to cover its length in bytes
Constant rate	Allows for large bursts to be sent faster by speeding up the output.
Does not allow saving	Allows saving up tokens (permissions) to send large bursts.

Resource reservation and packet scheduling

Resource reservation

Once a specific route for a flow is established it is possible to reserve resources along that route to make sure the needed capacity is available.

Three different kinds of resources can potentially be reserved: Bandwidth, Buffer space and CPU cycles.

Bandwidth

If a flow requires 1 Mbps and the outgoing line has a capacity of 2 Mbps flow can be routed through this line.

Buffer space

Arriving packet is deposited on the network interface card by the hardware itself. The router software copies it to a buffer in RAM and queue that buffer for transmission on the chosen outgoing line. If buffer is not available, the packet has to be discarded.

For a good quality of service, some buffers can be reserved for a specific flow so that flow does not have to compete for buffers with other flows.

CPU Cycle:

Router takes CPU time to process a packet.

Making sure that the CPU is not overloaded is needed to ensure timely processing of each packet.

Packet scheduling

- Implemented in hosts/routers to control link allocation
- Queuing algorithms
  - Fair queuing
  - Weighted Fair Queuing (WFQ)

Fair queuing

The essence of the algorithm is that routers have separate queues for each output line, one for each flow. When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue.

In this way, with n hosts competing for a given output line, each host gets to send one out of every n packets.

Weighted fair queuing:

One problem with this algorithm is that it gives all hosts the same priority.

In many situations, it is desirable to give video servers more bandwidth than regular file servers so that they can be given two or more bytes per tick.