20MCA13-Computer Networks

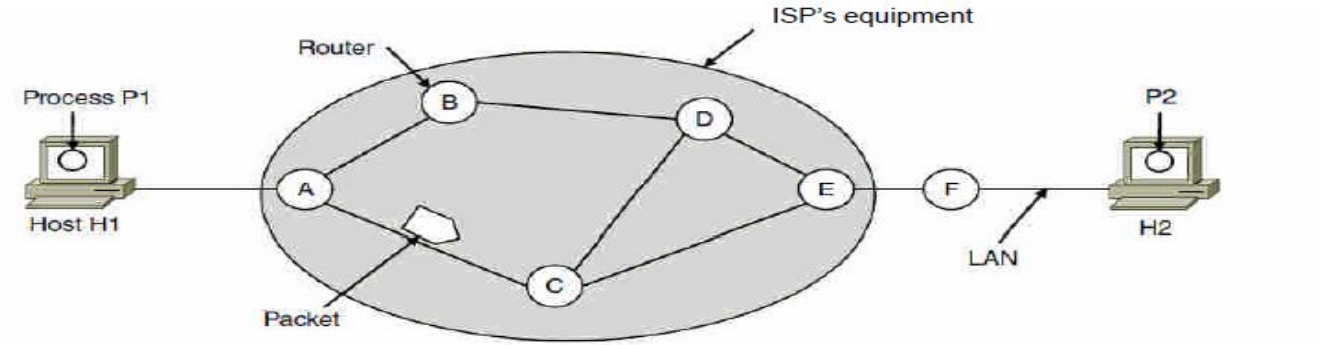
Unit-3
The Network Layer

- **Main service: Provide facilities for getting data from** a source to a destination - **routing.**
- Again, distinguish between connectionless and connection oriented services. There are two *implementation techniques:*
  - Virtual circuits are complete routes that are set up in advance.
  - Datagrams comprise individual packets of which the route is determined on the fly: they hop from router to router.
- Note: The services are independent of their implementation:

Network Layer Design Issues

- Store-and-forward packet switching
- Services provided to transport layer
- Implementation of connectionless service
- Implementation of connection-oriented service
- Comparison of virtual-circuit and datagram networks

Store-and-Forward Packet Switching



The environment of the network layer protocols.

Services Provided to the Transport Layer

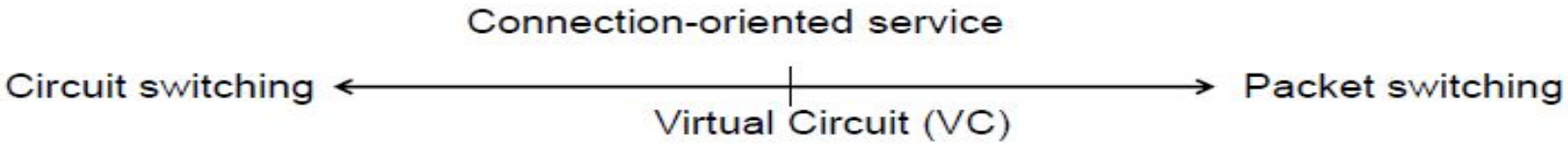
1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
   - even across LANs and WANs

Service Provided to the Transport Layer

- Two camps on whether the network should provide connection-oriented service or connectionless service:
  - **Connectionless service** (such as Internet):
    - Let the host do all processing of packet reordering, error control, flow control, etc.
    - Each packet carries the full destination address.
    - Network services simply with primitive SEND PACKET and RECEIVE PACKET and little else.
  - **Connection oriented service** (such as ATM, MPLS, VLAN)
    - Let the subnet provide a reliable, connection oriented service.
    - Quality of service (QoS) can be guaranteed, for real time traffic such as voice and video

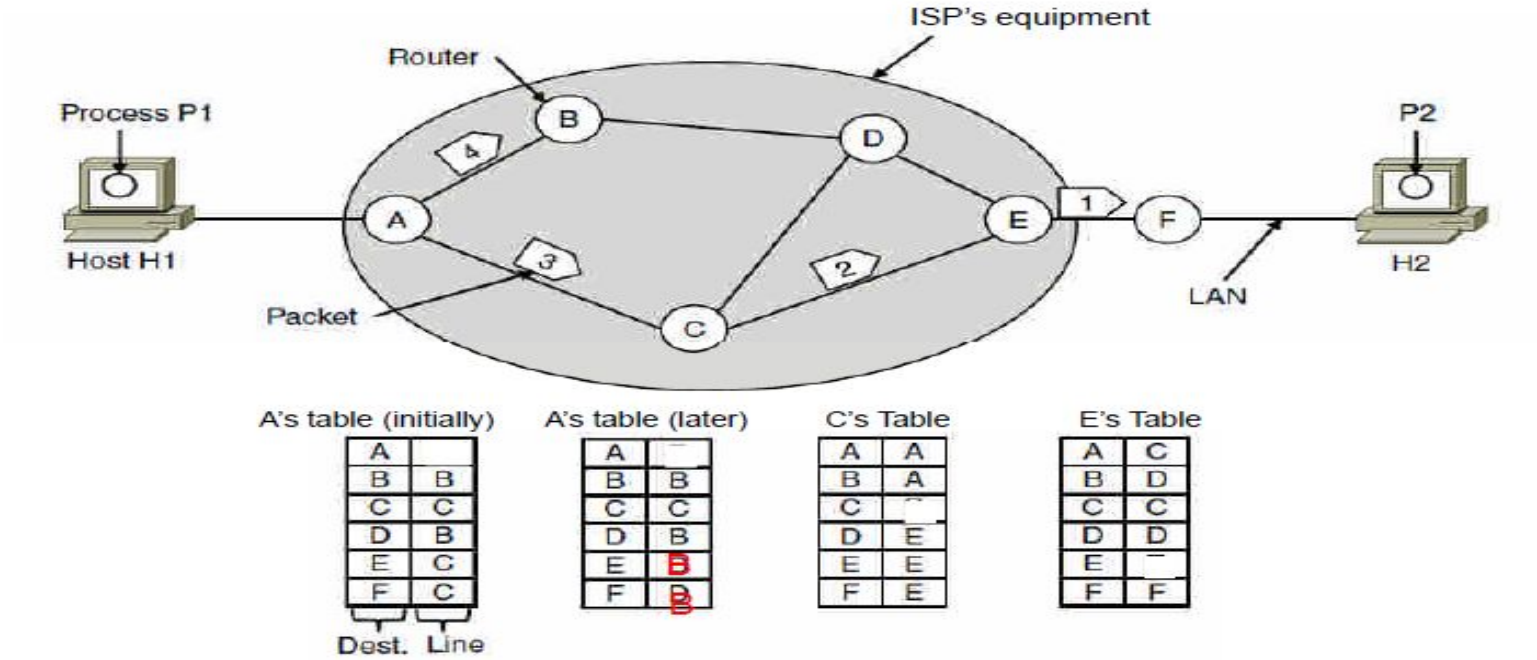
NETWORK LAYER DESIGN ISSUES - Internal Organization

1. **Virtual circuits (VC)** - connection setup (choosing of the route to the destination), forwarding packets over the exactly that route (the router maintains a table with the unique virtual circuit #), and connection release (termination of the VC).
2. **Datagrams** - no routers are working in advance, even if the service is connection oriented. Each packet is sent independently from the previous ones and routed over a different route. The routers do not maintain virtual circuit #s. Each packet is carrying the full destination address. Establishment of connections is done by the end stations, and does not require any special work from the routers.



- **Datagram** subnet
  - If connection service is offered, packets are injected into the subnet individually and routed independently.
- **Virtual circuit (VC)**subnet
  - If connection oriented service is used, a path from the source router to the destination router must be established before any packet can be sent.

Implementation of Connectionless Service



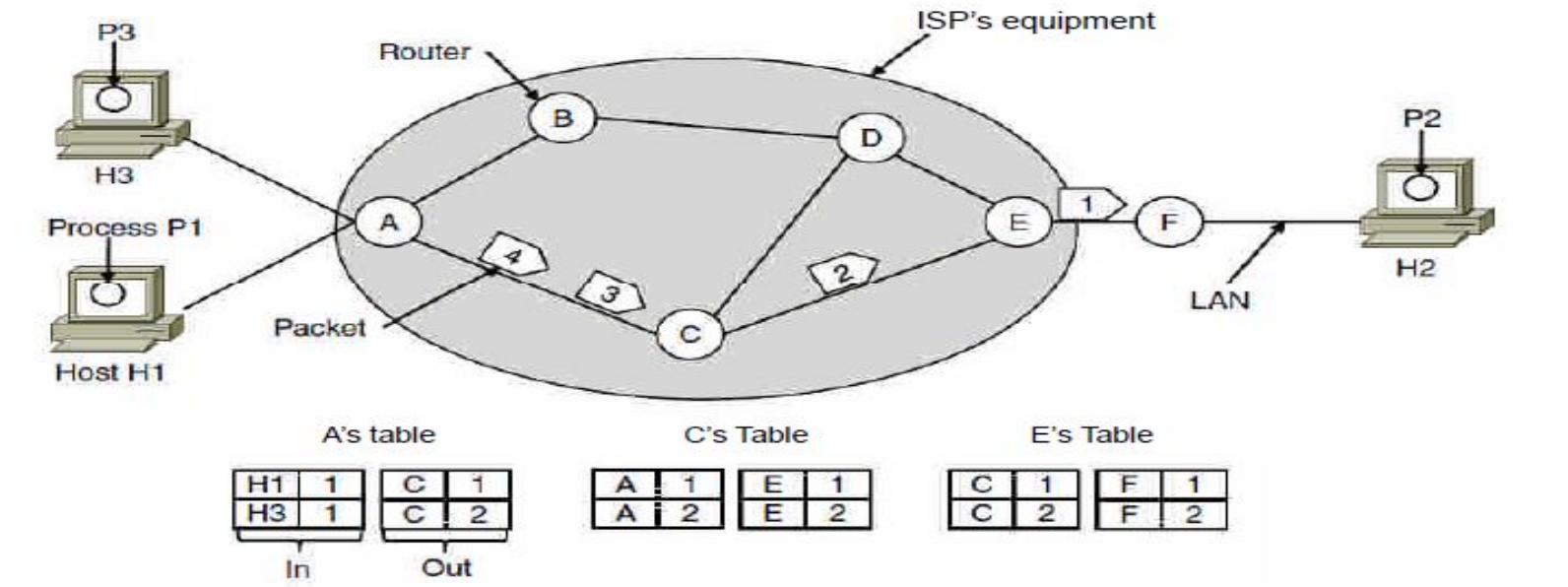
Routing within a datagram network

Routing algorithms

a) Every router has an internal table telling where to send packets for each possible destination.
   - Each table entry is a pair consisting of a destination and the outgoing line to use for that destination.

b) The algorithm that manages the tables and makes the routing decision is called the **routing algorithm**.

Implementation of Connection-Oriented Service



Routing within a virtual-circuit network

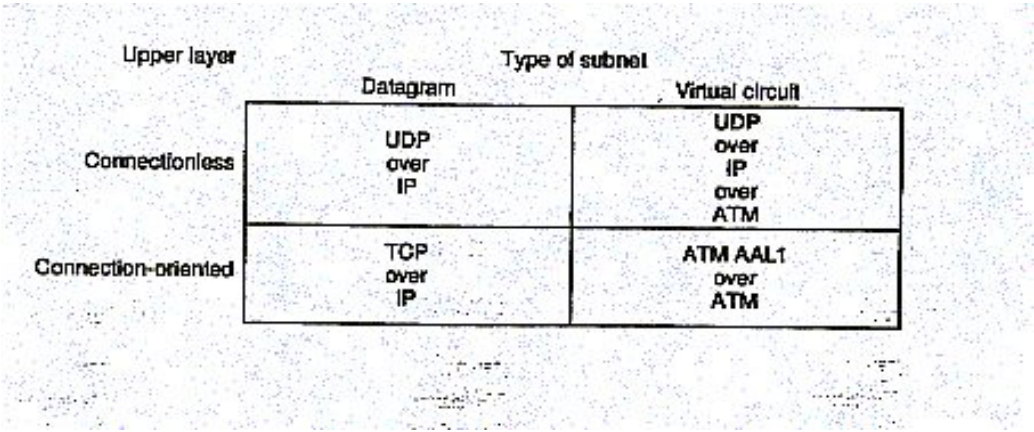
- A virtual-circuit subnet is needed for connection-oriented service.
- When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.
- …..Label Switching

Comparison of VC and Datagram Subnets

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

Comparison of VC and Datagram Subnets

- All variations of connectionless, or connection-oriented, over VC, or over datagrams, are possible - for example IP (connectionless) over ATM (VC) subnet.



Upper layer	Type of subnet: Datagram	Type of subnet: Virtual circuit
Connectionless	UDP over IP	UDP over IP over ATM
Connection-oriented	TCP over IP	ATM AAL1 over ATM

Routing

- Main issue: Routers that constitute the network layer of a network, should cooperate to find the best routes between all pairs of stations.
- Observation: All optimal routes from station A to other stations in the network, jointly constitute a sink tree:
- This means: Routers have to collaborate to build the sink tree (or something that comes near to that) for each source station.

Routing Algorithms

- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Routing in ad hoc networks

- The routing algorithm is part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
  - If the subnet uses datagram internally, this decision must be made anew for every arriving data packet.
  - If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up (called **session routing**).
- One router can be viewed as having two processes inside it.
  - One handles each packet as it arrives, looking up the outgoing line to use for it in the routing table.---**Forwarding**
  - The other is responsible for filling and updating the routing tables.---The **routing** algorithm.

ROUTING ALGORITHMS

Common Requirements:

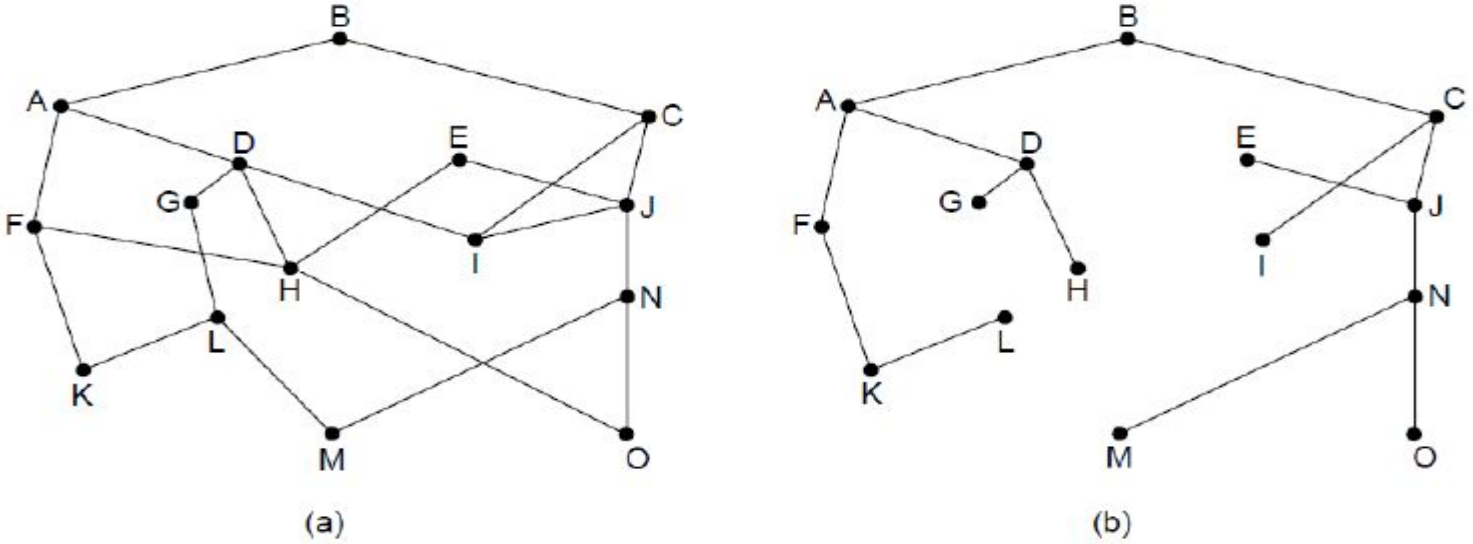
- Correctness - no deadlocks, livelocks, unreachable states
- Simplicity - fast handling of packets, less failures
- Robustness - dealing with failures, changes of the topology and of the traffic.
- Stability - the algorithm should congregate to equilibrium
- Fairness - no starvation, load balancing
- Optimality - short packet delay, max. throughput, number of hops

1.

1. **Non-adaptive algorithms (Static Routing) -** do not base the routing decisions on measurements. The routing is in most cases static (computed in advance), off-line and downloaded to the routers.
2. **Adaptive algorithms (Dynamic Routing) -** change their routing algorithms (continuous, periodic, occasionally) to reflect changes in the topology, and sometimes in the traffic. The information source can be local, nodes along the route, or all nodes.

ROUTING ALGORITHMS

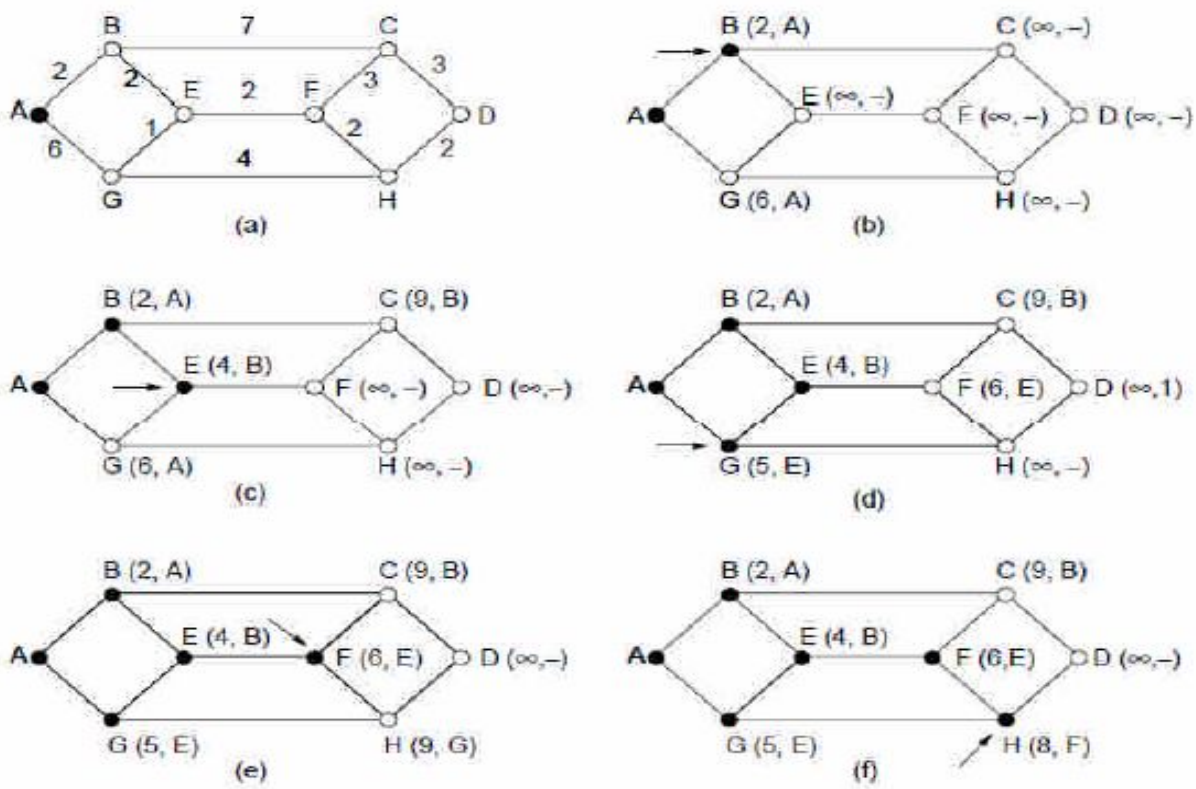
- The **optimality principle**:
  - If router *J* is on the optimal path from router *I* to router *K*, then the optimal path from *J* to *K* also falls along the same route. (Proof by contradiction).
- The optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**.
  - A sink tree is not necessarily unique.
  - The goal of all routing algorithms is to discover and use the sink trees for all routers.
  - Since a sink tree is a tree, it does not contain any loops, so each packet will be delivered within a finite and number of hops.



(a) A network. (b) A sink tree for router *B*.

ROUTING ALGORITHMS - Static

Shortest Path Routing - A frequently used metric is the number of hops. Each router examines each of the nodes adjacent to it, calculates the new distance sum, and if it is less than the label on that node, we have the shortest path, so the node is relabeled.



The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node

- A subnet is represented as a *graph* (*node*, *edge*), i.e. *G*(*N*,*E*).
  - Each edge is associated with a cost, which is a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay and other factors.
  - Several algorithms can be used for computing the shortest path between two nodes of a known graph.
- The Dijkstra algorithm:
  - Each node is labeled with its distance from the source node along the best known path.
    - Initially, no paths are known, so all nodes are labeled with infinity.
    - As the algorithm proceeds and paths are found, the labels may changes
      - A label may be either tentative or permanent.
      - Initially, all labels are tentative.
      - When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

Dijkstra's algorithm to compute the shortest path through a graph.

```
#define MAX_NODES 1024                          /* maximum number of nodes */
#define INFINITY 1000000000                     /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];              /* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                                /* the path being worked on */
      int predecessor;                          /* previous node */
      int length;                               /* length from source to this node */
      enum {permanent, tentative} label;        /* label state */
  } state[MAX_NODES];

  int i, k, min;
  struct state *p;

      . . .
```

. . .

```
for (p = &state[0]; p < &state[n]; p++) {        /* initialize state */
    p->predecessor = –1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0;  state[t].label = permanent;
k = t;                                           /* k is the initial working node */
do {                                             /* Is there a better path from k? */
    for (i = 0; i < n; i++)                      /* this graph has n nodes */
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
```

. . .

. . .

```
        /* Find the tentatively labeled node with the smallest label. */
        k = 0; min = INFINITY;
        for (i = 0; i < n; i++)
                if (state[i].label == tentative && state[i].length < min) {
                        min = state[i].length;
                        k = i;
                }
        state[k].label = permanent;
    } while (k != s);

    /* Copy the path into the output array. */
    i = 0;  k = s;
    do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

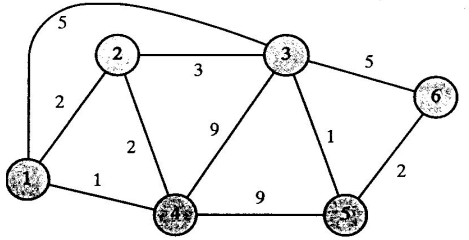
- Flooding
  - By flooding, every incoming packet is sent out on every outgoing line except the one it arrived on. To avoid generating vast number of duplicate packets, two approaches can be exploited:
  - Using a hop counter contained in the header of each packet.
    - counter is decremented at each hop, with the packet being discarded when the counter reaches zero.
  - How long is the path ?
  - Keeping track of which packets have been flooded, to avoid sending them out a second time
    - It necessitates the source router putting a sequence number in each packet it receives from its hosts.
  - Each router needs a list per source router telling which sequence numbers originating at that source have already been seen.
  - Using a counter for each list can prevent the list from growing without bound.

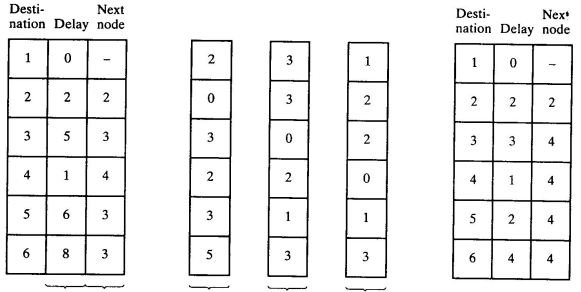
- Flooding is not practical, but has some uses:
  - In military applications, for its robustness.
  - In distributed database applications, for database update concurrently.
  - In wireless networks
  - A metric, against which other routing algorithms can be compared.
  - In the setup of flooding, the routers only need to know their neighbors.

ROUTING ALGORITHMS - adaptive

Distance Vector Routing (Bellman-Ford, Ford-Fulkenson). It was used in early versions of ARPANET and in Internet (RIP), DECnet,, AppleTalk and Cisco.

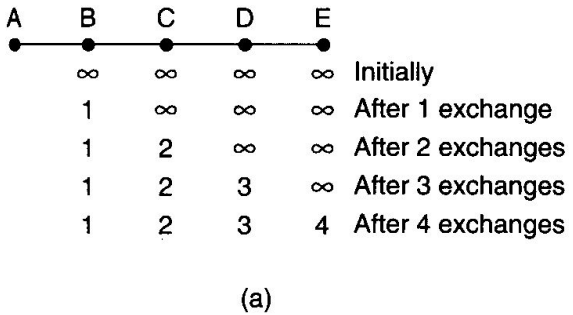
- Each router maintains a table (i.e. a vector) indexed by, and containing one entry for each router in the subnet. The entry contains the preferred outgoing line for this destination and an estimate giving the best known distance to that destination (# of hops, time delay, etc.).
- Once every T msec each router sends to (and receives from) each neighbor a list of estimated distance to each destination. The router recalculates the distances.



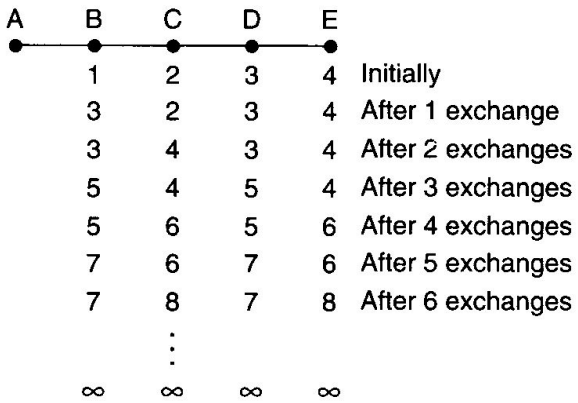


ROUTING ALGORITHMS - adaptive

Count-to-Infinity Problem - the distance vector routing propagates the good news, but leisurely to the bad news.



(a)



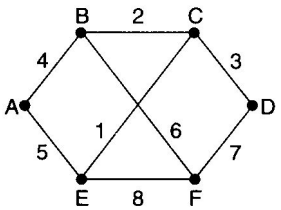
ROUTING ALGORITHMS - adaptive

Link State Routing (Second Generation in ARPANET) - the first generation did not consider the speed, but only the queue length, and took too long to converge.

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors (e.g. by ECHO packets)

 measure RTT/2 and calculate only the queue delay (to avoid load oscillation) or both, the queue delay and the communication load transformed to "link utilization".

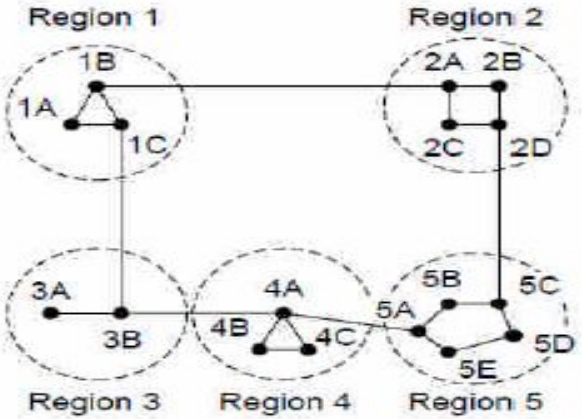
3. Construct a packet telling all it has just learned, and send the packet to all routers.
4. Compute the shortest path to every other router. The complete topology and all delays are experimentally measured and distributed to every router.



Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

- Hierarchical Routing
  - As networks grow in size, the router routing tables grow proportionally.
  - Issues then arise, on router memory, CPU time to scan them, and bandwidth to send status reports
  - 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.
  - The penalty to be paid is in the form of increased path length.
  - Kamoun&Kleinrock's result: The optimal number of levels for an N router network is *ln N, requiring a total of e ln N entries per router.*



(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)

Hierarchical routing.

Congestion Control Algorithms

- Congestion:
  - When too many packets are present in (a part of) the subnet, performance degrades.
  - When a queue builds up at some node for an output line, adding more memory may reduce packet loss probability.
  - If routers have an infinite amount of memory, congestion gets worse.
  - By the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent.
  - Slow processors and/or low-bandwidth lines may cause congestion.

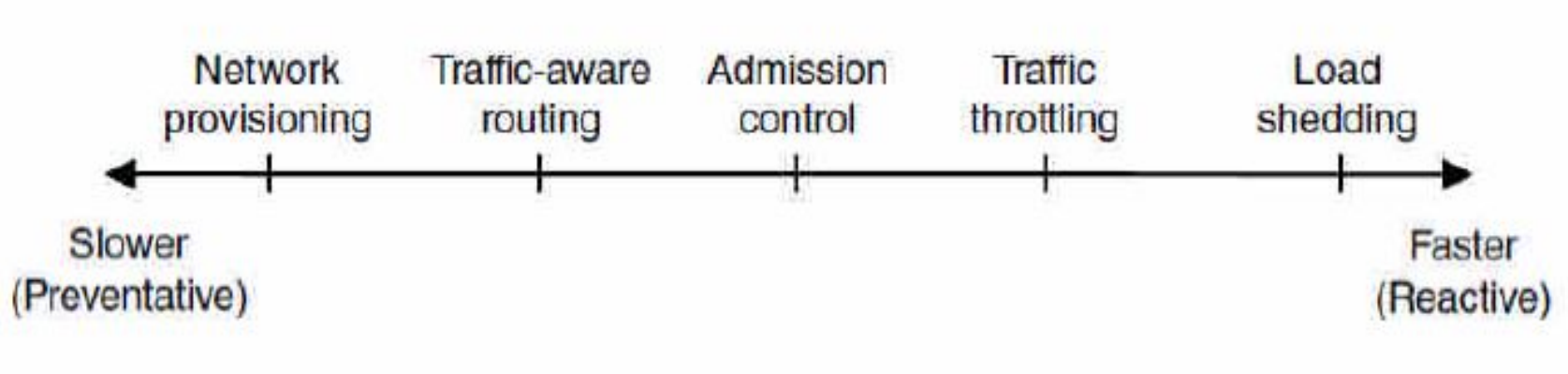
Congestion Control Algorithms

- Difference between congestion control and flow control:
  – Congestion control
    • Make sure the subnet is able to carry the offered traffic.
    • Global issue, involving the behavior of all the hosts, all the routers, the store-and-forwarding hosts, processing within the routers, and all the other factors that tend to diminish the carrying capacity of the subnet.
  – Flow control
    • Relate to the point-to-point traffic between a given sender and a given receiver.
    • Make sure that a faster sender can not continually transmit data faster than the receiver can absorb it.

Congestion Control Algorithms

- Congestion Control Algorithms
  - Approaches to congestion control
  - Traffic-aware routing
  - Admission control
  - Admission control
  - Traffic throttling
  - Load shedding

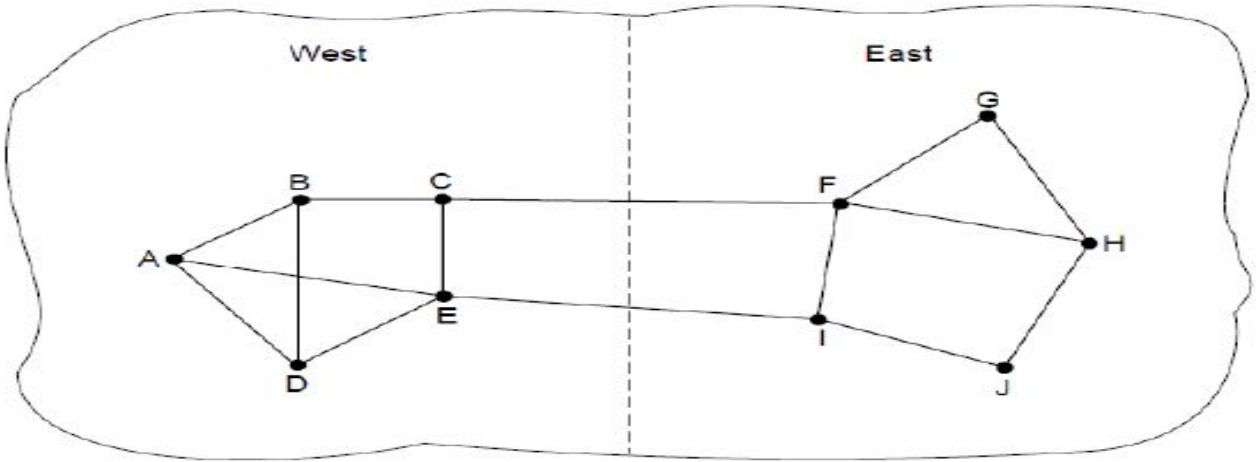
Approaches to Congestion control



- Two solutions to congestion:
  - Increase the resource
    - Preventive approaches in a longer timescale.
  - Decrease the load
    - React to it once it has occurred, taking effects in a shorter time scale.

  -Provisioning
    - Upgrade links and routers regularly according to utilization at the earliest months.
  - Traffic-aware routing
    - Dynamic/adaptive routing
  - Admission control
  - Traffic throttling
    - The network delivers feedback to the sources whose traffic flows are responsible for the problem.
  - Two issues:
    - » How to identify the onset of congestion and
    - » How to inform the source that needs to slow down.
  - Load shedding
    - The network discards packets that it cannot deliver

Traffic aware routing

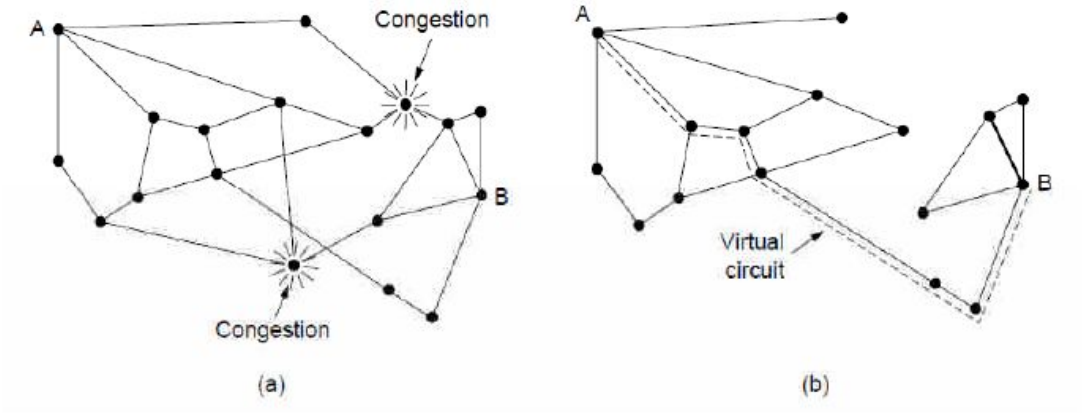


A network in which the East and West parts are connected by two links. (One path vs multiple path routing)

Traffic aware routing

- A technique widely used in virtual-circuit networks
  - Do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
- Traffic descriptor
  - A commonly used one is the **leaky bucket or token bucket.**
- Admission control can also be combined with traffic aware routing, by considering routes around traffic hotspots as part of the setup procedure.
  - E.g. Load-sensitive routing

Admission control



(a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from A to B is also shown.

Traffic Throttling

Congestion avoidance:

– Determine when congestion is approaching.

• By monitoring the utilization of the output line, the buffering of queued packets insides the routers, and/or the number of packets that are lost due to insufficient buffering.

• E.g, ExponentiallyWeighted Moving Average:

$$d_{new} = \alpha\, d_{old} + (1 - \alpha)\, s$$

where **d** is the estimate of the **queuing delay**, **s** is a **sample of the instantaneous queue length**/time, and the constant $a \in [0,1]$ determines how fast the router forgets recent history.

– Routers deliver timely feedback to the senders that are causing the congestion:

• Possible methods: Choke packets, explicit congestion notification, hop-by-hop backpressure.

Choke Packets

The router sends a **choke packet back to the source host directly.**

• The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets further along the path and is then forwarded in the usually way.

– When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by *X percent.*

• The host should ignore choke packets referring to that destination for a fixed interval.

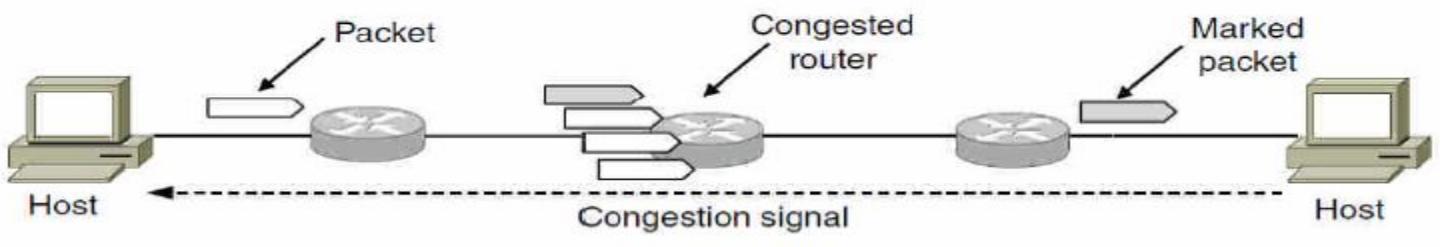
• After that period has expired, the host listens for more choke packets for another interval.

– If one arrives, the line is still congested. The host reduces the flow still more and begins ignoring choke packets again.

– If no choke packets arrive during the listening period, the host may increase the flow again.

– E.g. the SOURCE QUENCH message in the early Internet.

Traffic Throttling (2)

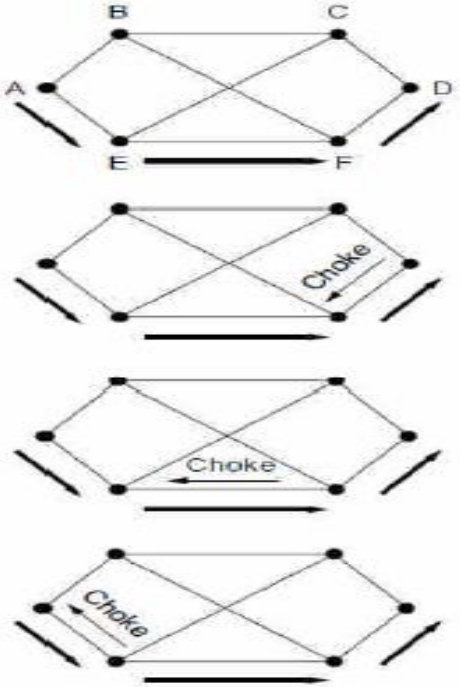


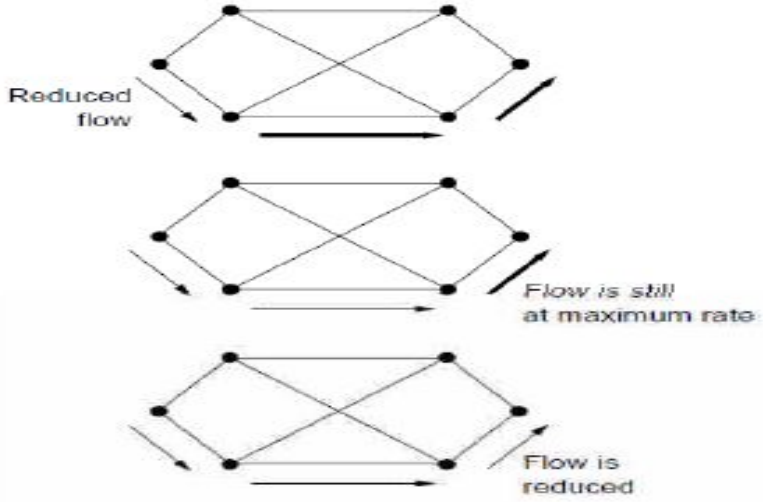
Explicit congestion notification

Explicit Congestion Notification

- The router signals the warning state of congestion by setting a special bit in the packet header.
  – When the packet arrived at the destination, the transport entity copies the bit into the next acknowledgement/reply sent back to the source.
  – The source then cut back on traffic.
  – Advantage: No additional packets are injected into the network.

Hop-by-Hop Backpressure (1)



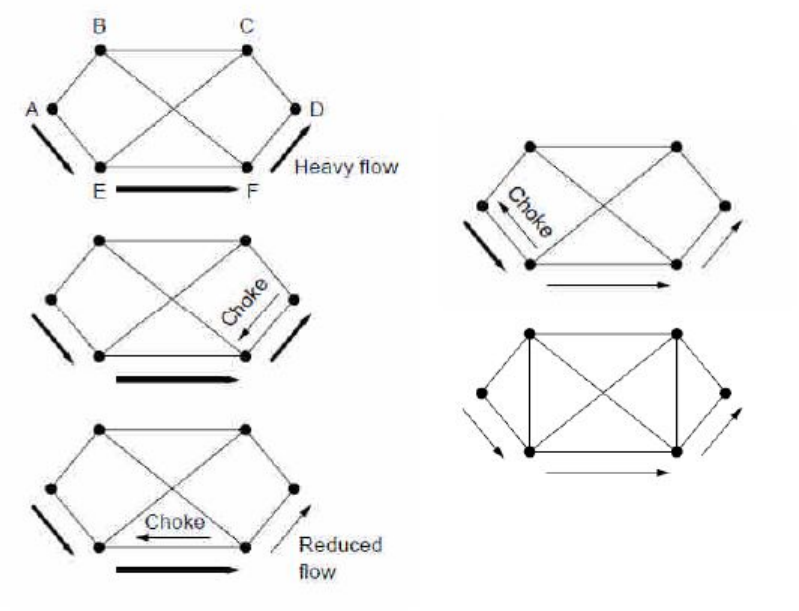


A choke packet that affects only the source..

Hop-by-hop backpressure

- At high speeds and over long distances, sending a choke/ECN packet to the source hosts does not work well because the reaction is slow.
  - An alternative approach is to have the choke packet take effect at every hop it passes through.
    - The net effect of this hop-by-hop scheme is to provide quick relief at the point of congestion at the price of using up more buffers upstream.

Hop-by-Hop Backpressure (2)



A choke packet that affects each hop it passes through.

Load Shedding

- **Load Shedding:** when routers are being inundated by packets that they can not handle, they just throw them away.
  – A router drowning in packets can just pick packets at random to drop, or
  – which packet to discard may depend on applications running:
    • For file transfer, an old packet is worth more than a new one. (**wine)**
    • For multimedia, a new packet is more important than an old one. (**milk)**
  – To implement an intelligent discard policy, applications must mark their packets in priority class to indicate how important they are.
  – Another option is to allow hosts to exceed the limits specified in the agreement negotiated when the virtual circuit was set up, but subject to the condition that all excess traffic be marked as low priority.

Random Early Detection (RED)

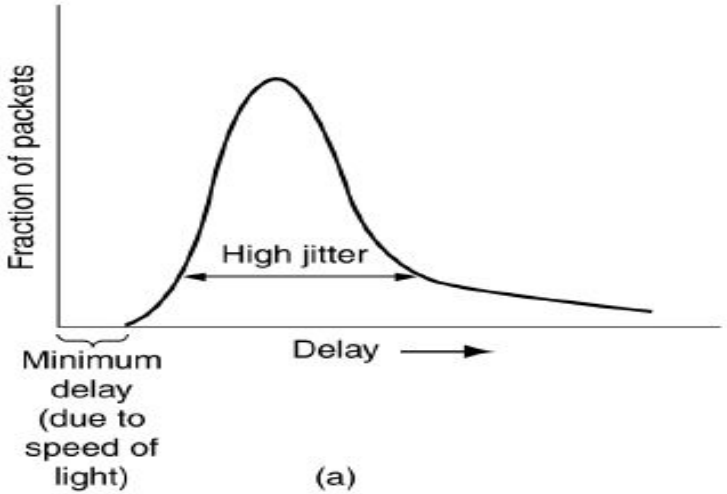
- Dealing with congestion after it is first detected is more effective than letting it gum up the networks and then trying to deal with it.
  - RED: Discard packets before all buffer space is really exhausted.
  - In some transport protocols (including TCP), the response to lost packets is for the source to slow down:
    - The reasoning is that TCP was designed for wired networks and wired networks are very *reliable, so lost packets are mostly due to buffer overruns rather than transmission errors.*
    - To work well with TCP in wireless networks, transmission errors due to *noise on the air link must recover at the data link layer.*
    - ECN (explicit signal) ↔ RED (Implicit signaling)

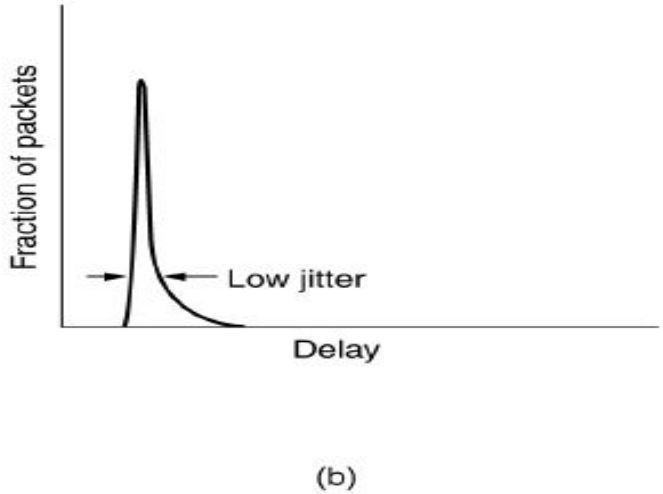
Quality of Service

- Application requirements
  - Traffic shaping
  - Packet scheduling
  - Admission control
  - Integrated services
  - Differentiated services

- A stream of packets from a source to a destination is called a **flow.**
  – In a connection-oriented network, all the packets belonging to a flow follow the same route.
  – In a connectionless network, all the packets sent from one process to another process may follow different route.
  - QoS parameters
    – Bandwidth, delay, jitter, and loss(reliability).

Jitter





(a) High jitter. (b) Low jitter.

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

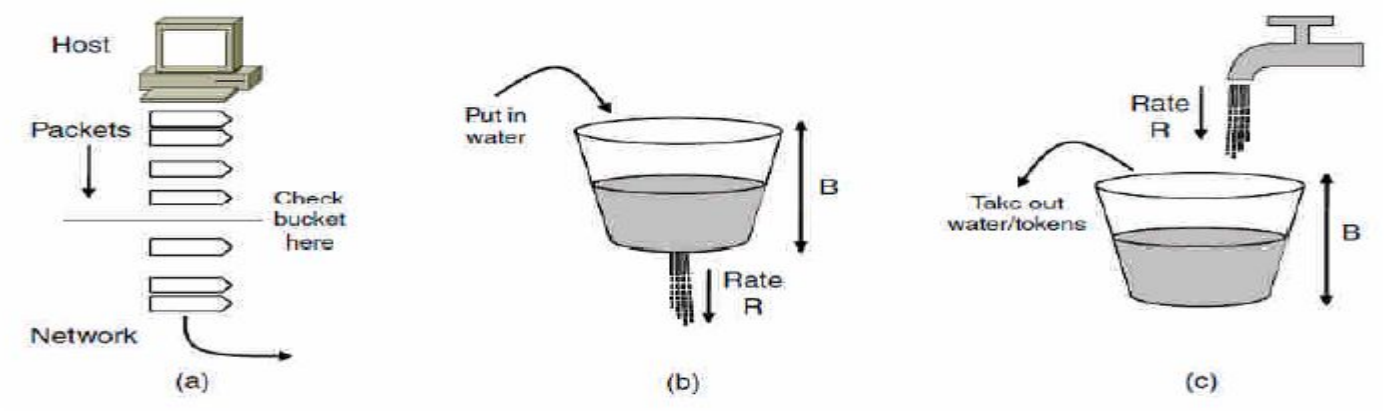
- ATM networks classify flows in four broad categories with respect to their QoS demands (These categories are also useful for other purposes and other networks) :
  - CBR: Constant bit rate (e.g., telephony)
  - rt-VBR: Real-time variable bit rate (e.g., compressed videoconferencing).
  - nrt-VBR: Non-real-time variable bit rate (e.g., watching a movie over the Internet).
  - ABR: Available bit rate (e.g., file transfer).

Categories of QoS and Examples

1. Constant bit rate
   - Telephony
2. Real-time variable bit rate
   - Compressed videoconferencing
3. Non-real-time variable bit rate
   - Watching a movie on demand
4. Available bit rate
   - File transfer

Traffic Shaping and traffic policing

- Traffic characteristics in telephone networks VS traffic characteristics in data networks.
  - Traffic in data networks is **bursty.**
  - Bursts of traffic are more difficult to handle than constant-rate traffic.
  - **Traffic shaping is about regulating the average *rate and*** burstness of a flow of data that enters the network.
    - In contrast, the sliding window protocols use one parameter to limit the amount of data in transit at once, indirectly limiting the rate.
  - Traffic shaping reduces congestion and thus helps the carrier live up to its promise of **service level agreement.**
  - Monitoring a traffic flow is called **traffic policing.**
    - Agreeing to a traffic shape and policing it afterward are easier with virtual circuit subnets than with datagram subnets. With datagram subnets, the same ideas can be applied to transport layer connections.



(a) Shaping packets. (b) A leaky bucket. (c) A token bucket

$$R_{out}(t) = \begin{cases} R & , B(t) > 0 \\ 0 & , B(t) = 0 \end{cases}$$

where $R_{out}(t)$ is the leaky bucket output rate and $B(t)$ the volume of water in the bucket at time t.

$$B(t+\Delta) = \left\{ \overline{B} \wedge \left(B(t) + \int_{t}^{t+\Delta} [R_{in}(x) - R_{out}(t)] dx \right) \right\}^{+}$$

where $R_{in}(t)$ is the input rate to the bucket at time t and $\overline{B}$ the bucket size.

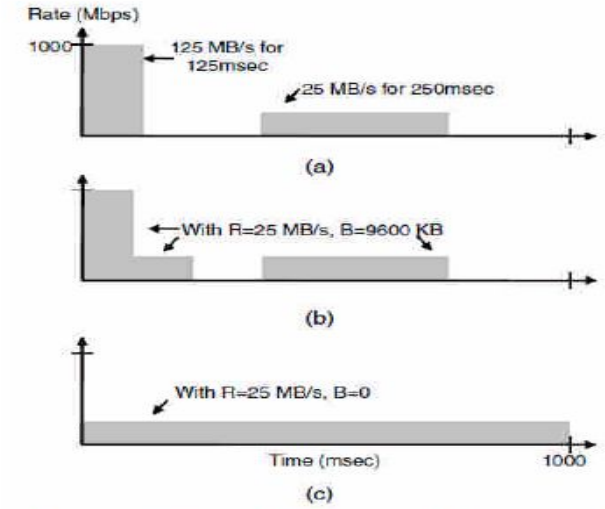
A (σ, ρ) traffic model:

σ : Burstiness

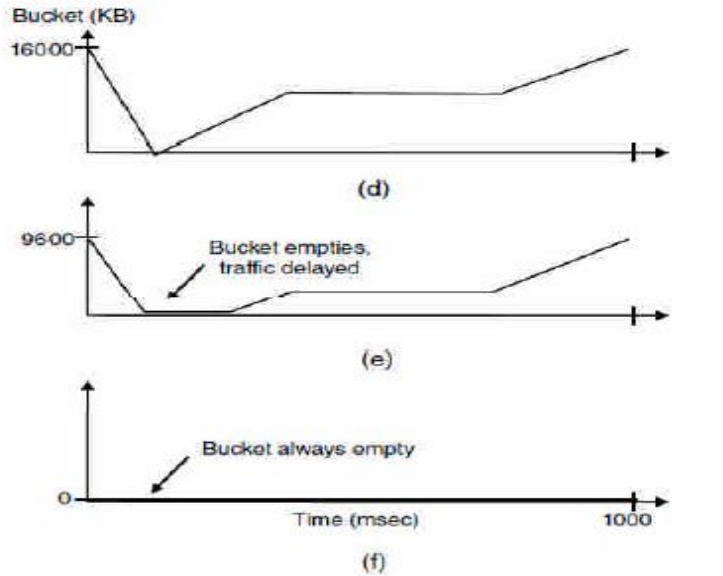
ρ : Token rate or long term average rate

$$A(t, t+s) = \int_{t}^{t+s} R_{out}(\tau) d\tau \leq \sigma + \rho \times s, \text{ for all } t > 0 \text{ and } s > 0.$$

Traffic Shaping (2)



(a) Traffic from a host. Output shaped by a token bucket of rate 200 Mbps and capacity (b) 9600 KB, (c) 0 KB.



Token bucket level for shaping with rate 200 Mbps and capacity (d) 16000 KB, (e) 9600 KB, and (f) 0KB..

- Calculate the maximum rate burst:

S: the burst length, in second

ρ: token arrival rate, in byte/sec

M: the maximum output rate, in byte/sec

C : the token bucket capacity, in byte.

Hence,

$$C + \rho S = MS,$$

giving the result $S = C/(M-\rho)$.

Admission Control

- QoS guarantee for new flows over a network.
  - QoS routing, rather than the single best path between each source and each destination.
  - Translate the parameters of a **flow spec** to the requirement of router resources (bandwidth, buffers, cpu
  - Hard guarantees (Deterministically) versus soft guarantee (Stochastically)
  - Does there exist any negotiable flow parameter?

- Flows are described in terms of **flow specification.**
  – The sender produces a flow specification proposing the parameters it would like to use.
  – As the specification propagates along the route, each route examines it and modifies the parameters as need be.
  – The modification can only reduce the flow, not increase it.
  – When it gets to the other end, the parameters can be established.
  • Five parameters of the flow specification based on RFCs 2210 and 2211
  – *Token bucket rate, token bucket size, peak data rate, minimum packet size, and maximum packet size.*
  – The size in the last two includes the transport and network layer headers.

Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

An example **flow specification**
(RFCs 2210 and 2211 for IS)

Resource reservation

- •If a specific route for a flow is available, it becomes possible to reserve resources along that route. Three different kinds of resources can potentially be reserved:
  - – Bandwidth
  - – Buffer space
  - – CPU cycles
  - • M/M/1 queueing system:
    - – Arrivals are Poisson distributed with mean rate packets/sec.
    - – The service time of each packet is exponentially distributed with mean 1/µsec.
    - – The expected delay experienced by a packet throughout the distribution

Resource reSerVation Protocol

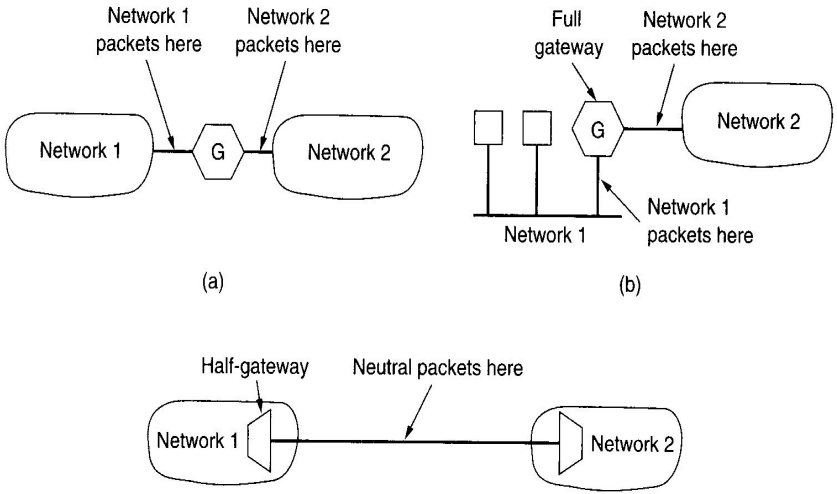
- RSVP-The Resource reSerVation Protocol
  – The IS architecture, described in RFCs 2205--2210
  – It is used for making the reservations; other protocols are used for sending the data.
  – It allows multiple senders to transmit to multiple groups of receivers, permits individual receivers to switch channels freely, and optimizes bandwidth use while at the same time eliminating congestion.
  • Its simplest form:
  – Multicast routing using spanning trees.
    • Each group is assigned a group address.
    • To send to a group, a sender puts the group's address in its packets.
    • The standard multicast routing algorithm then builds a spanning tree covering all group members. (Routing is not part of RSVP).
    • The only difference from normal multicasting is a little extra information that is multicast to the group periodically to tell the routers along the tree to maintain certain data structures in their memories.

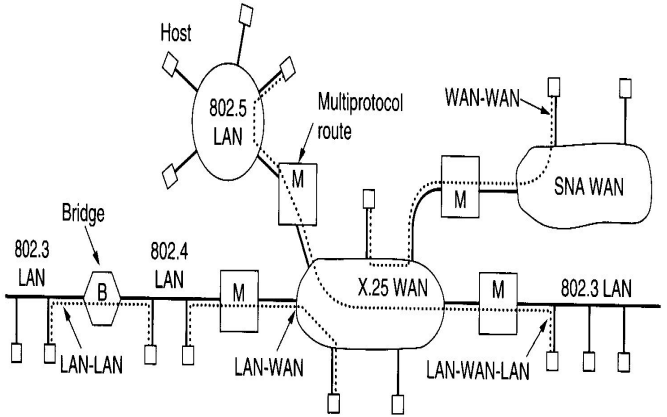
- To get better reception and eliminate congestion,
  - Any of the receivers in a group can send a reservation message up the tree to the sender.
  - The message is propagated using *reverse path forwarding algorithm.*
  - At each hop, the router notes the reservation and reserve the necessary bandwidth. If insufficient bandwidth is available, it reports back failure.
  - By the time the message gets back to the source, bandwidth has been reserved all the way from the sender to the receiver making the reservation request along the spanning tree.

- When making a reservation,
  – a receiver can (optionally) specify one or more sources that it wants to receive from.
  – It can also specify whether these choices are fixed for the duration of the reservation or whether the receiver wants to keep open the option of changing sources later.
- The routers use this information to optimize bandwidth planning.
  – Two receivers are only set up to share a path if they both agree not to change sources later on.
- Reserved bandwidth is decoupled from the choice of source.
  – Once a router has reserved bandwidth, it can switch to another source and keep that portion of the existing path that is valid for the new source.

INTERNETWORKING

- **Repeater, Bridge, Muliprotocol Router, Transport Gateways, Application Gateways**





INTERNETWORKING

- **How networks differ?**

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

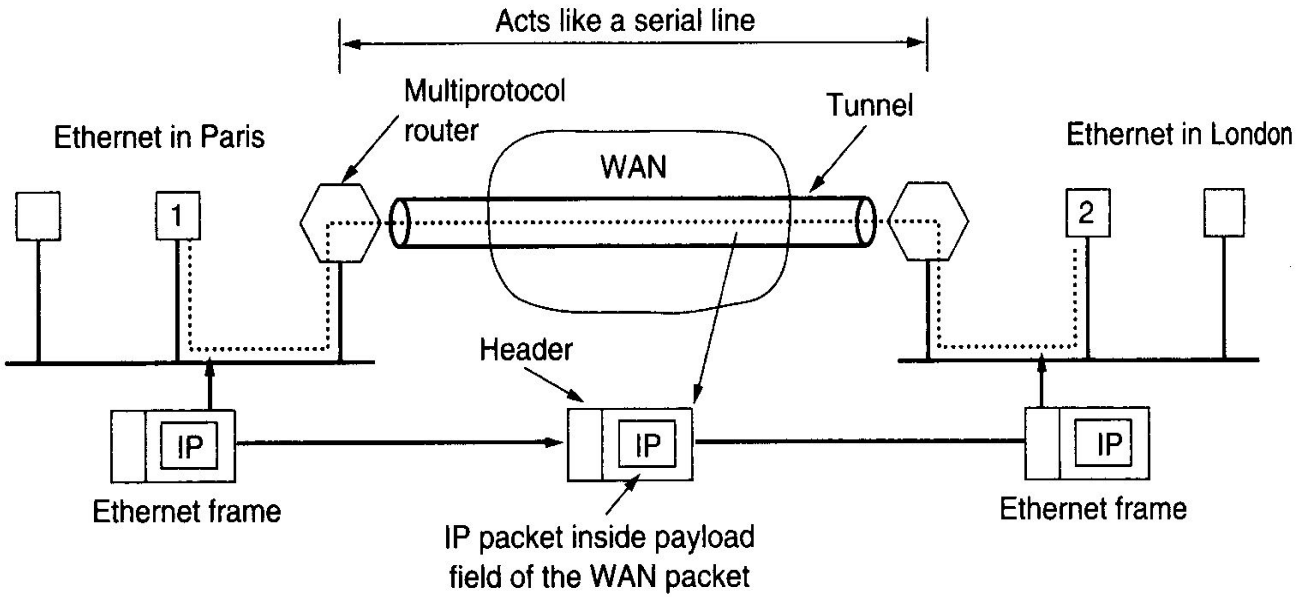
- Different devices for network interconnections:
  - In the physical layer (Layer 1):
    – Repeaters, or hubs, which move *bits from one network to* an identical network.
  - In the data link layer (Layer 2):
    – Bridges and switches, which accept *frames, examine the* MAC address, and forward the frames to a different network.
  - In the network layer (Layer 3):
    – Routers, which connect two networks.
  - If two networks have dissimilar network layers, the router may be able to translate between the *packet formats. Packet translation is* increasingly rare.
  - A router that can handle multiple protocols is called a **multiprotocol router.**

- In the transport layer (Layer 4):
  – Transport gateways, which can interface between two transport connections (by translating layer 4 *segments).*
- In the application layer (Layer 5):
  – Application gateways, which translate *message semantics.*
- An essential difference between the switched (bridged) case and the routed case:
  – With a switch (or bridge), the entire frame is transported on the basis of its MAC address.
  – With a router, the packet is extracted from the frame and the address in the packet is used for deciding where to send it

- It is very difficult to make two different networks interwork. However, there is a common special case:
  – *The source and destination hosts are on the same type of network, but there is a different network in between.*
- – An example is shown in the figures below, where the technique to the problem is called **tunneling.**
- To send an IP packet to a host in London, a host in Paris office constructs an IPv6 packet containing an IPv6 address in London and sends it to the multiprotocol router that connects the Paris IPv6 network to the IPv4 Internet.

INTERNETWORKING

- **Tunneling**



- When the multiprotocol router gets the IPv6 packet, *it encapsulates the packet with an IPv4 header addressed to the IPv4 side of the multiprotocol router that connects to the London IPv6 network.*
  - When the packet gets there, the London router removes the IPv6 packet and sends it onward to the destination host.
  - **Overlay:**
  - The above network is an example of overlay " IPv6 over IPv4"
  - *Disadvantage of tunneling:*
  - None of the hosts on the network that is tunneled over can be reached.
  - This is turned into an *advantage with VPNs (Virtual Private Networks)*

Internetworking

- A two-level routing algorithm
  - Intradomain routing, interior gateway protocol (IGP)
    - Since each network is operated independently of all the others, it is often referred as an **Autonomous System (AS).**
  - E.g., an ISP network
  - Interdomain routing, exterior gateway protocol (EGP)
    - In the Internet, the protocol is called BGP (Border gateway protocol.
    - Constrained to some *routing policy that governs the*

- Packet size issues:
  1. Hardware
  2. Operating system
  3. Protocols
  4. Compliance with (inter)national standard.
  5. Reduce error-induced retransmissions
  6. Prevent packet occupying channel too long.

- To reduce overheads, large packets are preferred for transmission.
  - Problems: When a large packet wants to travel through network whose maximum packet size is to small.
  - Solutions:

  1. Find **Path MTU (Path maximum Transmission Unit), to make sure the** problem does not occur.

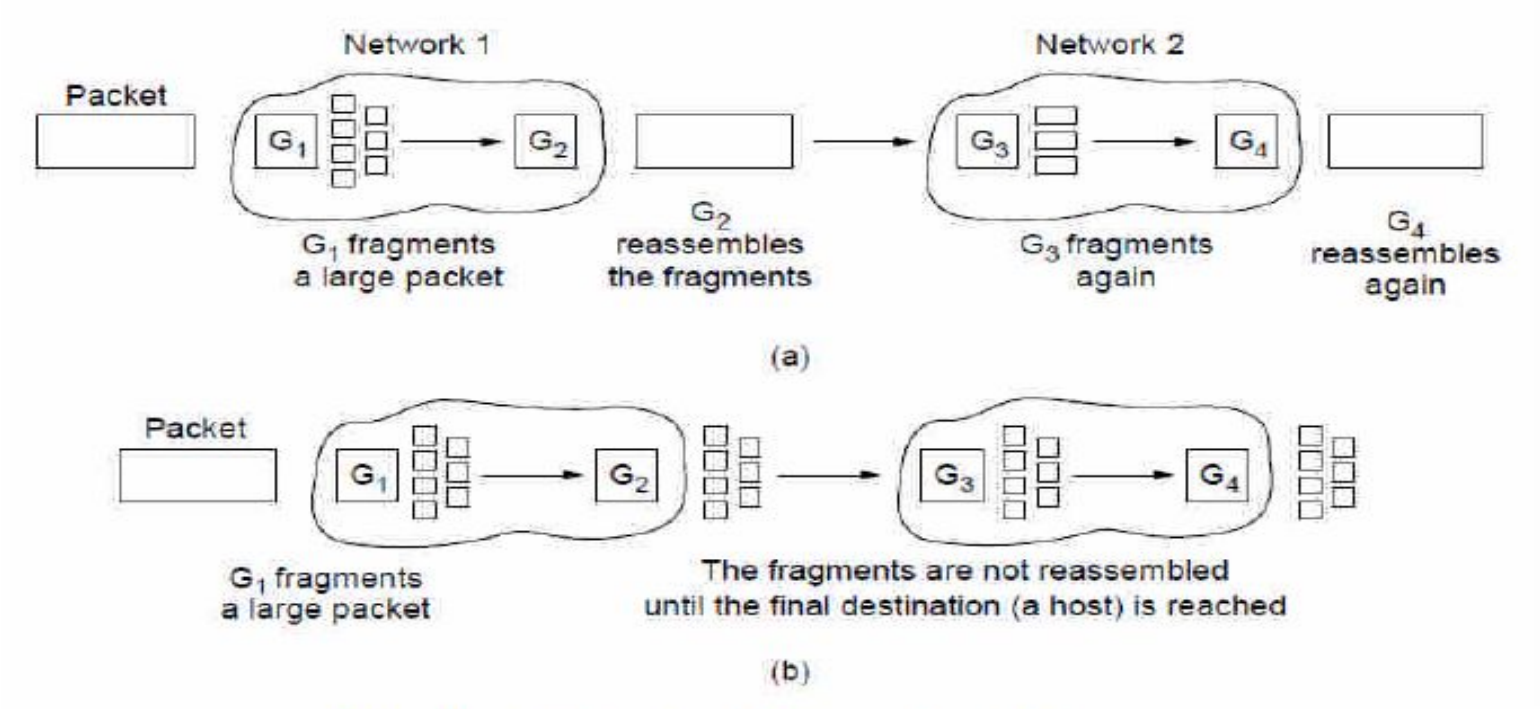
  2. Allow routers to break packet **into fragments and send each fragment** as a separate internet packet.

  - Two opposing strategies for recombining the fragments back into the original packet:

  1. Transparent fragmentation

  2. Nontransparent fragmentation.

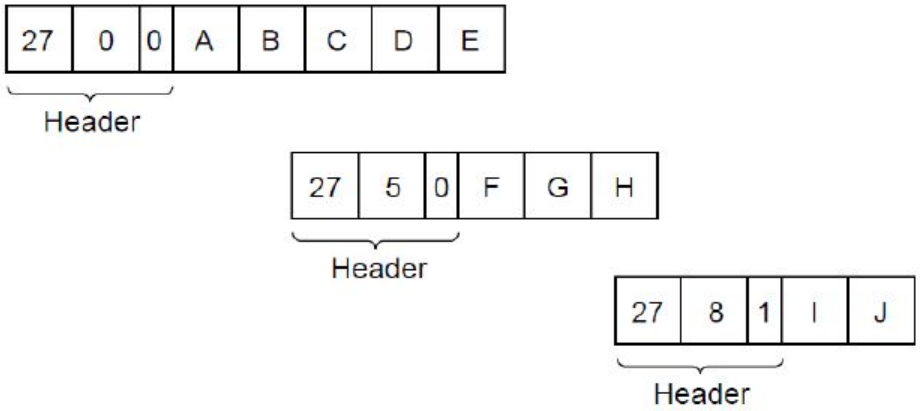
- Transparent fragmentation:
  – Make fragmentation caused by a small-packet network *transparent to any subsequent networks through which the* packet must pass on its way to the destination.
- Nontransparent fragmentation:
  – Refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet.
  – Recombination occurs only at the destination host.



(a) Transparent fragmentation.

(b) Nontransparent fragmentation

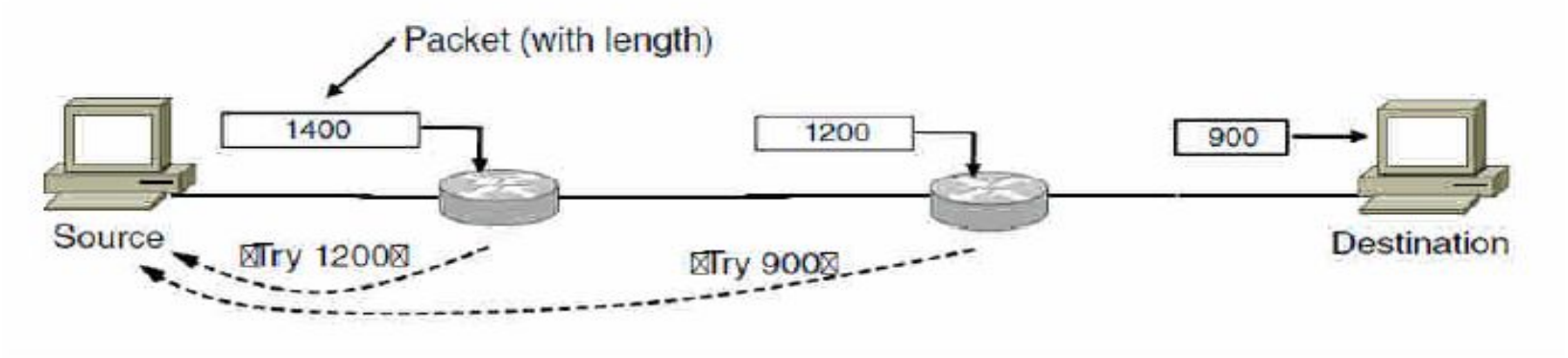
- When a packet is fragmented, the fragments must be numbered in such a way that the original data stream can be reconstructed.
  – Define an elementary fragment size small enough that the elementary fragment can pass through every network.
  –All the pieces are equal to the elementary fragment size except the last one, which may be shorter.
  – An internet packet may contain several fragments.
  – Internet header must provide the original packet number, and the number of the (first) elementary fragment contained in the packet.
  – A bit in it indicates the last elementary fragment contained within the internet packet is the last one of the original packet.
  – This approach requires three fields in the Internet header:
    - The *original packet number,*
    - The *fragment number, as an offset number*
    - A *flag, indicating whether it is the end of the packet.*



Fragmentation when the elementary data size is 1 byte

(c) Fragments after passing through a size 5 gateway.

- Path MTU discovery:
  – Each IP packet sent with its header bits set to indicate no fragmentation is allowed to be performed.
  – A router receiving a packet that is too large generates an error packet, returns it to the source, and drops the packet.
  – Source on receiving the error packet, uses the information inside and re-fragment the packet into pieces small enough for the router to handle.



Path MTU Discovery