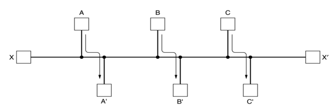
Chapter 5:

Routing Algorithms: ~Shortest Path Routing ~Distance Vector Routing ~Link State Routing ~Hierarchical Routing ~Broadcast Routing ~Multicast Routing ~Routing for Mobile Hosts ~Routing in Ad Hoc Networks

[The main function of the network layer is routing packets from source to destination. Routing algorithm decides which output line an incoming packet should be transmitted on]

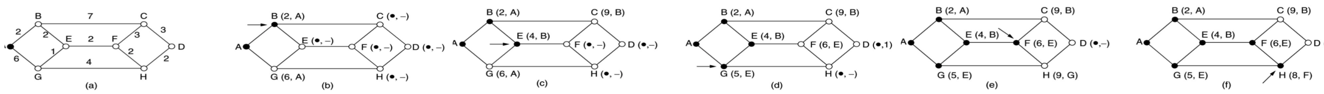


Traffic between A and A', B and B', and C and C' saturate the horizontal link. This conflicts with the traffic between X and X'.

Routing algorithms can be grouped into two classes: [Non-adaptive – do not change routing decision] [Adaptive – change routing decision if topology or traffic changes]

Property of GOOD Routing Algorithm: ~Correctness & simplicity ~[Robustness: Ability to cope with changes in topology & traffic from hard/software failure] ~[Stability: Converge to equilibrium & stay there after some time] ~Fairness ~Optimality

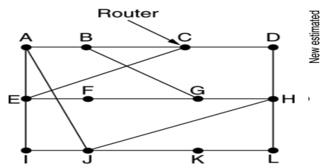
Shortest Path Routing: The labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measure delay & other factors.



~The Dijkstra's algorithm: The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node. (a) Marking node A as permanent. (b) Examining each neighbor of node A and relabeling the distance. Making the node with the smallest label the new working node. (c) Examining each neighbor of node B. Relabeling node E and C with the shortest distance.

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

Distance Vector Routing (Bellman-Ford): Dynamic routing algorithm. Each router maintains a table (vector) giving best known distance to each other router & which line to use to get there. The tables are updated by exchanging information w/ neighbors



E.g. J→G

J→A→G: 8 + 18 = 26 | J→I→G: 31 + 10 = 41 | J→H→G: 6 + 12 = **18** | J→K→G: 31 + 6 = 37

~The first 4 columns show the delay vectors received from the neighbors of J. ~Assume each router can estimate the distance to every neighbor. ~Now J can compute the updated distance to each router and the new route.

Link-State Routing: Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing for two reasons: 1.Line bandwidth was not considered. 2.Distance vector routing took too long to converge.

Each router must do the following: 1.Discover neighbors, learn their network address. 2.Measure delay or cost to each neighbor 3.Construct packet telling all it has just learned 4.Send packet to all other routers 5.Compute shortest path to every other router.

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.

Multicasting: Sending a message to a group of nodes Requires group management: create, destroy group, allow process to join & leave groups. Routers must know which hosts belong to which groups. Each router computes a spanning tree covering all other routers

When a packet is sent to a mobile host, first it is routed to the host's home LAN | Mobile Ad hoc Networks (MANETs) ~Assume no fixed base station ~Moving nodes (e.g. mobile computers) form the network ~Multi-hop connections ~Link broken when neighbor moves away/is out of power ~Nodes serve as host as well as router ~Topology changes dynamically ~No fixed infrastructure [Use of MANETS include: tactical military; online class; police]

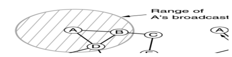
AODV (Ad hoc On-demand Distance Vector) routing algorithm. Two phases: **1.**Route discovery – E.g., node A→node I. A floods Route Request (RR) packets **2.**Route maintenance | ↓*(says "Range of A's Broadcast)*

ROUTE REQUEST packet:

Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

ROUTE REPLY packet:

Source address	Destination address	Destination sequence #	Hop count	Lifetime



(Example of range)

Congestion: When too much traffic is offered, congestion sets in and performance degrades sharply. Congestion Control Principles: ~Open-loop and Closed-loop solutions (principles continued on the next line…)

1.Monitor system: detect when/where congestion occurs [metrics include % of dropped packets, avg. queue length, avg. packet delay] **2.**Pass info where action can be taken: router sends a packet to traffic source **3.** Adjust system operation to fix problem: reduce traffic volume

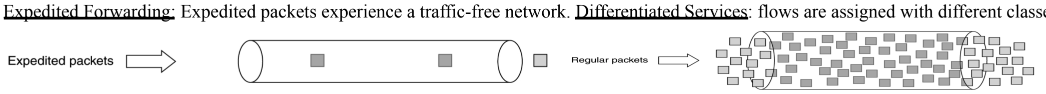
Open-loop solutions – Congestion Prevention: The systems are designed to min. congestion in the first place. Appropriate policies are used at various layers (i.e. data-link & physical)

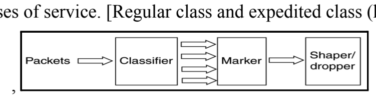
Jitter: variation (standard deviation) in packet arrival time. Audio and video streams require low jitter. High jitter: some packets take 20 msec & others take 30 msec. Each router checks to see if packet is behind/ahead of schedule, then arrange packet position in outgoing queue.

Techniques for Achieving Good Quality of Service: ~Over-provisioning: providing much router capacity, buffer space & bandwidth. Problem - expensive. ~Buffering: flows can be buffered on the receiving side before delivery; only increases delay & reduces jitter. ~Traffic Shaping: Makes the server (sender) transmit at a uniform rate. Smooths out traffic on the server side. Regulate avg. rate of data transmission. Leaky Bucket Algo. [rigid output pattern] Token Bucket Algo. [output rate may change depends on incoming traffic bursts]

Integrated Services: addressing both unicast & multicast multimedia applications RSVP: for making reservation, allows multiple senders to transmit to multiple groups of receivers, optimizes bandwidth use, eliminates congestion.

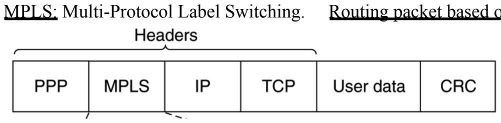
Expedited Forwarding: Expedited packets experience a traffic-free network. Differentiated Services: flows are assigned with different classes of service. [Regular class and expedited class (higher priority)]



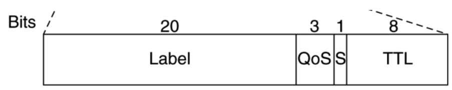


-Assured Forwarding (Four Priority Cases)

MPLS: Multi-Protocol Label Switching. Routing packet based on the label: ~**Label:** the label index ~**QoS:** Class of service ~**S:** stacking multiple labels in hierarchical networks ~**TTL:** Time to Live



MPLS:

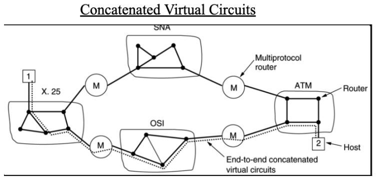


Transmitting a TCP segment using IP, MPLS, and PPP.

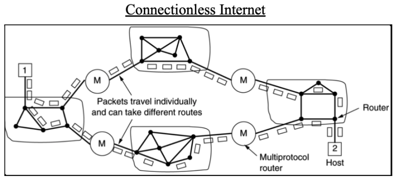
How Networks Differ: ~Service offered: Connection vs connectionless ~Protocols: IP, IPX, SNA, MPLS, etc. ~Addressing: Flat (802) vs hierarchical (IP) ~Multicasting: Present or absent (also broadcasting) ~Packet Size: Every network has its own max ~Quality of Service: Present or absent; different kinds ~Error Handling: Reliable, ordered, & unordered delivery ~Flow Control: Sliding window, rate control, other, or no ~Congestion Control: Leaky bucket, token bucket, RED, choke packets ~Security: Privacy rules, encryption ~Parameters: Different timeouts, flow specs ~Accounting: By connect time, by packet, by byte, or not at all

How Networks can be Connected: ~Ethernets connected via switch ~Ethernets connected via router

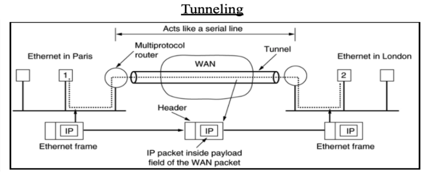
Concatenated Virtual Circuits



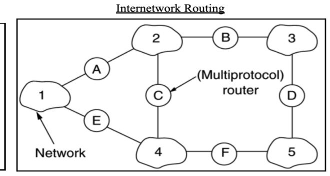
Connectionless Internet



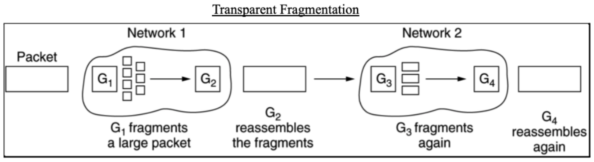
Tunneling



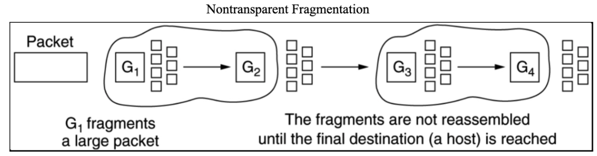
Internetwork Routing



Transparent Fragmentation

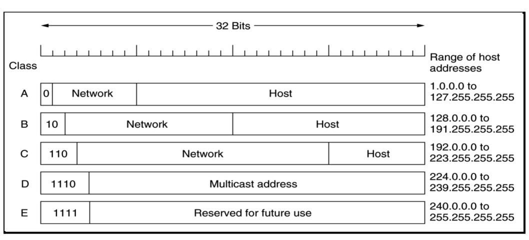


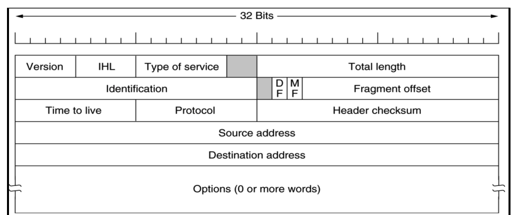
Nontransparent Fragmentation



Design Principles for Internet
1.Make sure it works **2.**Keep it simple.
3.Make clear choices **4.**Exploit modularity.
5.Expect heterogeneity.
6.Avoid static options and parameters.
7.Look for a good design; it need not be perfect.
8.Be strict when sending and tolerant when receiving.
9.Think about scalability.
10. Consider performance and cost.

IP Protocol: ~IHL – IP Header Length. ~Total length – includes both header and data ~DF – Don't Fragment. To order the router not to fragment it. ~MF – More Fragment ~Protocol – tell which transport protocol

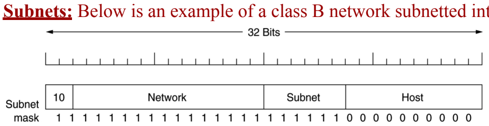




The IPv4 (Internet Protocol) header.

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Subnets: Below is an example of a class B network subnetted into 64 subnets. The practice of dividing a network into two or more networks is called subnetting. Subnet is a logical subdivision of an IP network.



Network Address Translation: ~Placement & operation of NAT box. ~IP addresses r scarce. ~A LAN uses 1 IP address. ~Each computer uses internal IP address which is converted into real IP address by NAT. ~For incoming traffic, port number is used

Internet Control Message Protocol: ~The principal ICMP message types. ~ICMP is used to report unusual events in the Internet. ~Each ICMP message type is encapsulated in an IP packet.

Address Resolution Protocol: ~Data link layer hardware doesn't understand IP address. ~LANs use data link layer address – e.g. 48-bit Ethernet address (MAC). ~Configuration profile can be used. ~ARP asks Ethernet: Who owns this IP address? Owner replies

Dynamic Host Configuration Protocol: ~DHCP allows both manual IP address assignment and automatic assignment. ~DHCP is based on a server to assign IP addresses to hosts asking for one.

OSPF: ~OSPF is a link state routing protocol. ~An Autonomous System [AS] is divided into areas. || Four types of routers: **1.**Internal routers are wholly within one area. **2.**Area border routers connect two or more areas **3.**Backbone routers are on the backbone. **4.**AS boundary routers talk to routers in other AS.