Data Communication Networks
Network Layer

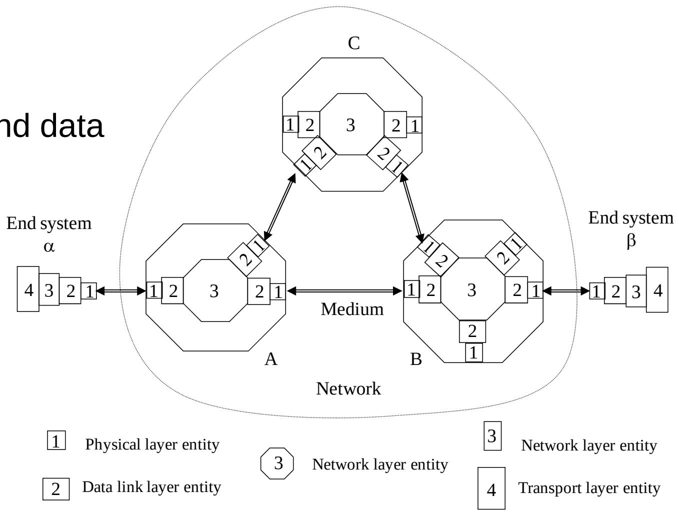
M. R. Pakravan

Department of Electrical Engineering

Sharif University of Technology

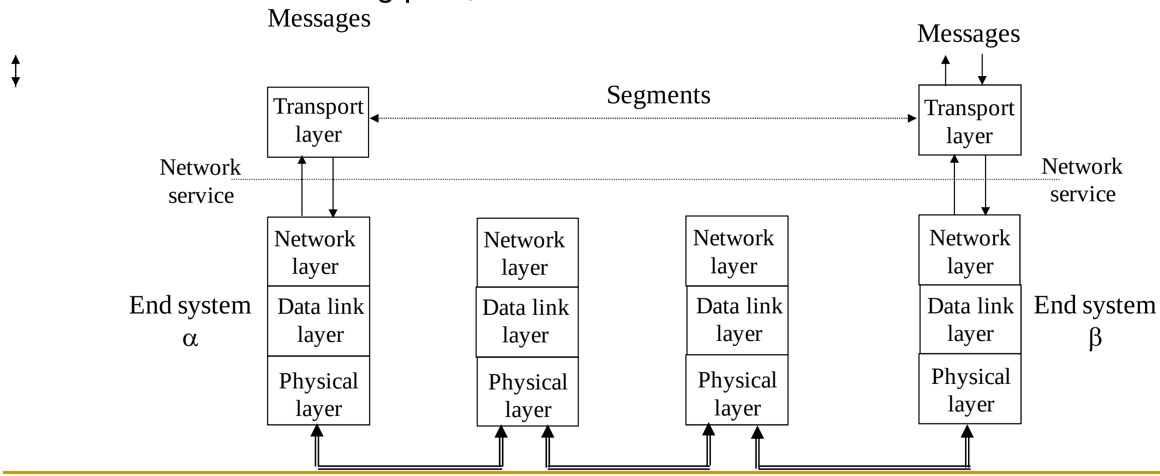
The Network Layer

- Role of network layer: ***routing packets***
  - Deals with end-to-end communication
  - Aware of network topology
  - Choose appropriate route
  - Communicates with transport and data link layers
- Network layer topics
  - Design issues
  - Routing algorithms
  - Congestion control
  - Quality of Service
  - Internetworking
  - Network layer in the Internet: IP



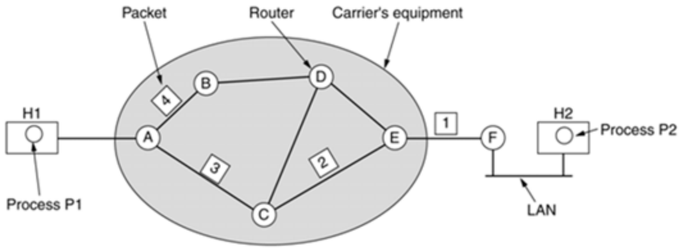
Network Layer Design Issues

- Services provided to the transport layer
  - Independent of chosen subnet technology
  - Shield transport layer from topology, type and number of subnets
  - Uniform numbering plan, across LANs and WANs



Network layer design issues

- **Connection-oriented or connectionless service?**
  - Where to put the complexity?
- Internet community: connectionless
  - Network will be unreliable anyway => handle this issue in higher layer (s)
  - Only SEND packet and RECEIVE packet needed
  - Hosts are getting powerful; so why not put the complexity there (in the transport layer) => easy to adapt
  - Speed more important than accuracy

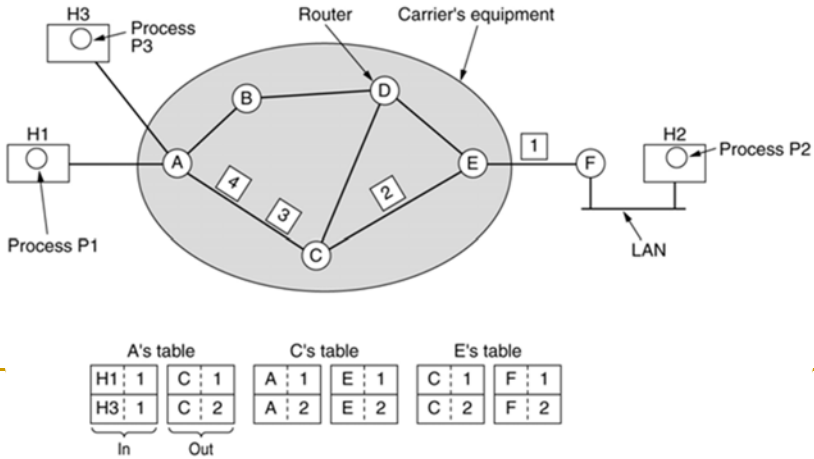


	A's table		C's table		E's table	
	initially		later			
A	–	A	–	A	A	A	C

Dest.	A's table (initially) Line	A's table (later) Line	C's table Line	E's table Line
A	–	–	A	C
B	B	B	A	D
C	C	C	–	C
D	B	B	D	D
E	C	B	E	–
F	C	B	E	F

Network Layer Design Issues

- Telephone companies: NL should be connection-oriented
  - Set up connection first
  - Each connection has unique ID
  - Negotiate parameters (quality, cost)
  - Communication in both directions
  - Use of Acknowledge gives automatic flow control



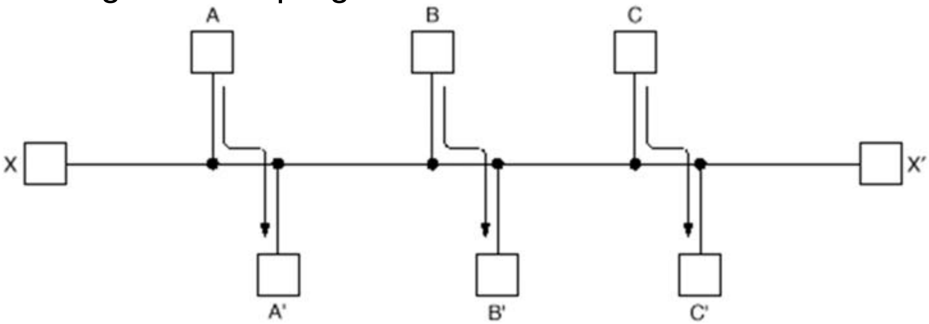
Virtual Circuits & Datagrams

- **Virtual circuits** :
  - Avoid routing every packet or cell independently
  - All packets/cells follow same route (= session routing)
  - Store # VC in routing tables (consuming some resources)
- **Datagrams**
  - Each packet contains destination and is independently routed
  - Also usable for connection-oriented service

Issue	Datagram subnet	Virtual-circuit 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	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing

- Message / packet / cell has to pass multiple hops
- Routing algorithm : decide which outgoing link to use
- Routing algorithm requirements
  - Correctness
  - Simplicity
  - Robustness: cope with changing topology
  - Stability: converge quickly
  - Fairness: guarantee progression



Routing algorithms

- Non-adaptive (static) routing:
  - Do not use measurements or estimates of current traffic and topology
  - Static routing: calculate route off-line or use deterministic algorithm
  - Examples:
    - Sheet-sign routing
    - Flooding: send packets in all directions
- Adaptive (dynamic) routing:
  - Routing decision depends on network status
  - Update state information regularly
  - What type of network information do we base the decision on?
    - Hop count
    - Distance, estimated transit time
    - Traffic rate
    - Congestion metrics

Routing Algorithms: Flooding

- Static algorithm
- Idea: forward incoming packet on all other outgoing lines
- Prevent over-flooding:
  - Hop counter (HC): Decrease HC on every Hop and discard packets with HC= 0
  - Avoid routing the same packet twice: Keep list of 'outstanding' packets using sequence numbers
  - Selective flooding: Route only in directions which are about right
- Flooding Properties:
  - Not practical in most applications
  - Generates too much traffic
  - Useful for:
    - Extreme robustness (military purpose)
    - Broadcast
    - Reference algorithm
  - It always finds the shortest delay path !!

Static Routing

- Optimality principle:
  - If B is on optimal path of AC then the whole path BC is part of this path
- Consequence: optimal routes from all sources to a given root (= sink node) form a *sink tree* $\Rightarrow$ no loops.
- Each packet delivered passes a limited number of hops
- Ideally we would like to use a sink tree for each router
- Practice: routers may go down => Should be adaptively updated

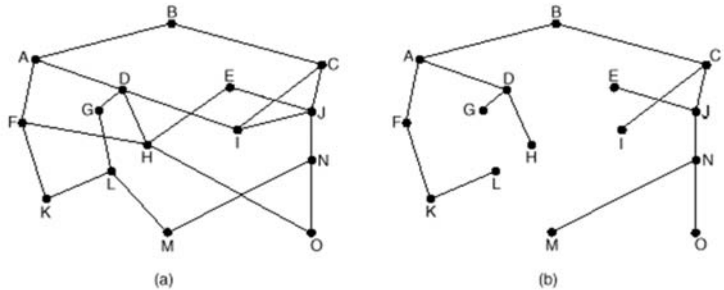
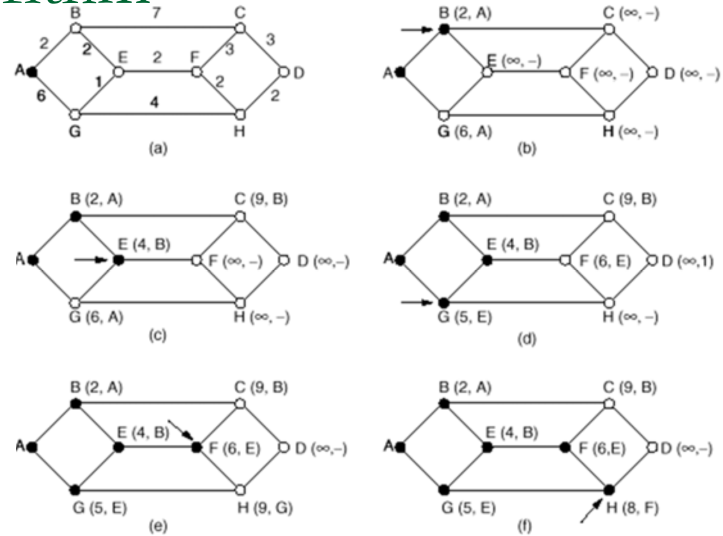


Fig. 5-5. (a) A subnet. (b) A sink tree for router *B*.

Shortest Path Algorithm

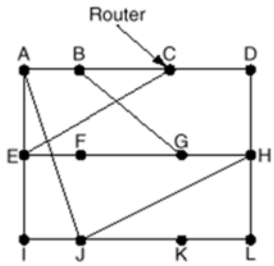
- Used as part of many routing algorithms
- Build router graph G(N, L)
  - N being nodes,
  - L being the set of links (or edges)
- Choose path metric (weight of lines L)
  - # hops
  - estimated delay (using echo packets)
  - physical distance
  - queue length (of outgoing lines)
- Apply Dijkstra's shortest path algorithm:
- Steps to find all shortest paths from A
- Initially: make A permanent: P = {A} : label all neighbors k of A with $D_k = d_{Ak}$
- Repeat until P contains all nodes
  - find next closest node and add it to P:
  - P= P + {i} , with D_i = min D_k (k not in P)
  - Updating labels of neighbours of P
  - D_k = min [D_k , $D_k + d_{ik}$] (k not in P)
- Key fact: the path to the next closest node i has to go through node from P only.



	Base Set	B	C	D	E	F	G	H
0	A	2,A	∞	∞	∞	∞	6,A	∞
1	A,B		9,B	∞	4,B	∞	6,A	∞
2	A,B,E		9,B	∞		6,E	5,E	∞
3	A,B,E,G		9,B	∞		6,E		8,F
4	A,B,E,G,F		9,B	∞				8,F
5	A,B,E,G,F,H		9,B	10,H				

Dynamic routing

- Distance vector routing
  - Used in early ARPANET
  - Each router maintains distance table indexed by destination number
  - Entry contains : outgoing link + estimated distance
- How to know distance?
  - Send echo packet to all neighbors; this packet gets back with timestamp therefore, distance to neighbor is determined
- Receive regularly updated tables from neighbors and update own table
- D (a,b) = min (D (a,x) + D (x,b)) {for all neighbors x}
- Other metrics such as hop count or cost could also be used



(a)

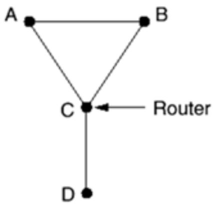
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

Distance vector routing

- Count-to-infinity problem:
  - Good news propagates much faster than bad news
- Split Horizon hack: (one of many efforts to reduce the impacts of this problem)
  - C does not report B the distance of node X for which packets have to flow through B
  - C tells D the truth about the distance to A, but tells B that CA= ∞
  - When AB fails, the bad news propagates as quickly as the good news
- Failure of Split Horizon
  - Init: CD=1, AD=2, BD=2, AB=1
  - CD fails, A and B tell C they can't reach D, so C sets CD=∞
  - Unfortunately, A hears from B that BD=2, so it sets AD=3 => wrong!



A	B	C	D	E	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	∞	∞	∞	∞	

(b)

Link State Routing

- Solves slow convergence problem of the distance vector approach
- Idea:
  - Send all nearest neighbor information to all other routers
  - Each router uses this information to know the topology and determines shortest path to all nodes
- Link State Routing: 4 steps
  - Discover neighbors (and their IDs)
  - Measure delay (or cost) to reach neighbours
  - Send link state packets to all other routers
  - Compute topology and shortest path to every router
- Perform last three steps at
  - Regular intervals
  - When major event happens (link / node going down or coming up)

Link State Routing

- Discover neighbor :
  - After booting : send HELLO packet; neighbor should respond telling its identity
  - When multiple routers are connected to LAN treat LAN as separate node
- Measure delay:
  - Send ECHO packet; neighbors respond immediately
  - Calculate trip time = delay/2
  - Average of multiple measurements may be used
  - Queuing delay on outgoing lines can be used in calculations or ignored
  - Oscillations may occur if the delay is considered
- Alternatively use bandwidth as cost metric

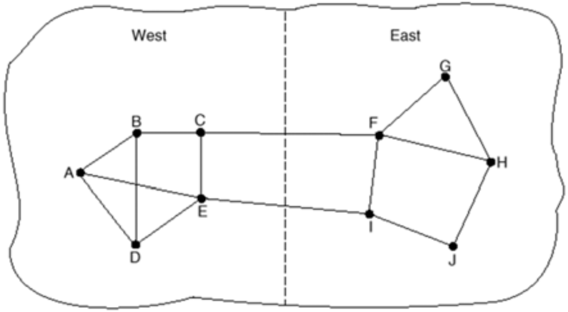


Fig. 5-14. A subnet in which the East and West parts are connected by two lines.

Link State Routing

- Sending link state packets:
  - Make link state packet
  - Use flooding to distribute packets to all routers
  - Use sequence numbers to reduce traffic
  - Routers keep (source, seq#) pairs and discard lower sequence numbers
  - Use age fields to handle corrupted sequence numbers
  - Age is decremented in regular intervals
  - Information is discarded when age is zero

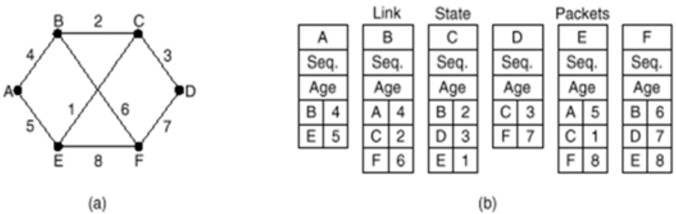


Fig. 5-15. (a) A subnet. (b) The link state packets for this subnet.

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

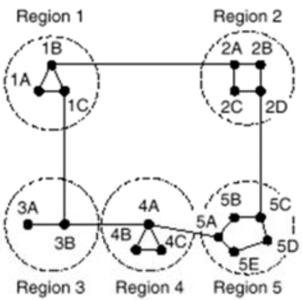
Fig. 5-16. The packet buffer for router *B* in Fig.5-15.

Link State Routing

- Route Calculation
  - Compute new routes (when packets from all sources have been received):
  - use shortest path algorithm on each node (e.g. Dijkstra's algorithm)
- Problem with Link State Routing:
  - large memory requirements ~ O (NK), where: N=number of routers K=network node degree
- Link state routing used widely. For example in OSPF (open shortest path first) routing used in the Internet

Hierarchical Routing

- Purpose: reduce routing table size and routing computation time
- Solution: use multi-level topology
  - For example in a 3-level network having:
    - k1 clusters
    - k1* k2 regions
    - k1 * k2 * k3 nodes
  - Instead of requiring k1*k2*k3 entries we need only k1+k2+k3 entries
- Increase in mean path length is sufficiently small



(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 Routing Methods

- Send lots of individual packets
  - Lots of traffic
  - Source needs to know all destination addresses
  - Generally not used
- Flooding
  - Generates too many packets
- Multi-destination routing
  - Packet contains list of destinations
  - Router splits packet among output lines and updates destination list
  - Again all destination addresses have to be known!

Broadcast Routing Methods

- Sink tree (= *spanning tree*)
  - Each packet is sent along the lines of the sink tree of the sender
  - Most efficient method for broadcast to all
  - Sink tree is not always known (e.g. not when distance vector routing is used; link state routing does have this knowledge!)
- Reverse path forwarding
  - Does not require network knowledge
  - Easy to implement
  - Transmission automatically finishes
  - Reasonably efficient
  - No list of destinations needed
  - No special mechanism needed to stop the forwarding

Broadcast Routing: Reverse Path Forwarding

- Each router knows the normal line for sending packets to all other routers
- When a packet is received from a router, it is checked to see if it is coming from the "normal" line.
- Send packet to all neighbours, but discard incoming packets from 'wrong' lines
- Packets from A arriving at B on line L is 'wrong' if B does not send to A at line L
- The idea is that if a packet arrives at the wrong line it probably did not follow the shortest route
- The overall total generated traffic is slightly more than required, but efficiency is reasonable

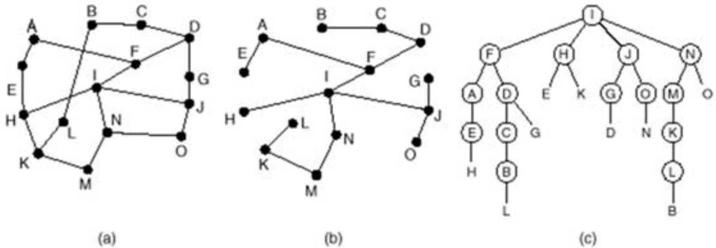
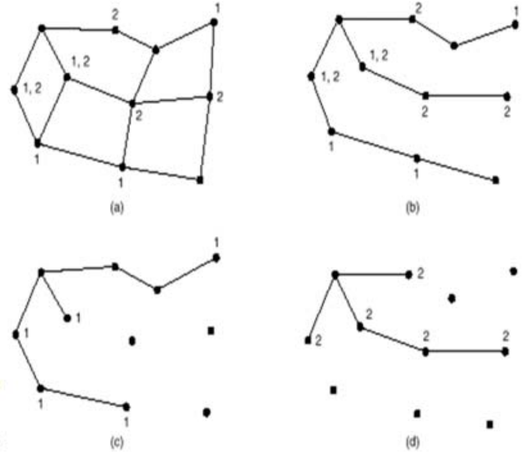


Fig. 5-20. Reverse path forwarding. (a) A subnet. (b) A spanning tree. (c) The tree built by reverse path forwarding.

Multicast Routing

- Multicast: Sending packets to a group of nodes => Group management required
- Routers must know to which groups their hosts belong
- Each router computes spanning tree for each supported group
- Multicast packets are then forwarded to appropriate spanning trees only
- When multiple spanning trees share some paths, trees need to be pruned to limit the packet forwarding only to the relevant routers

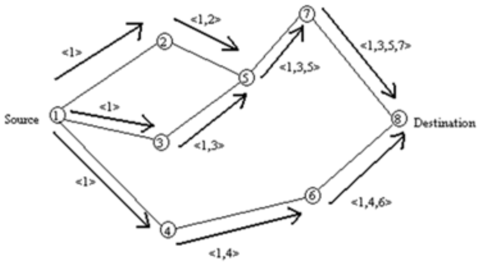


Ad-Hoc Routing

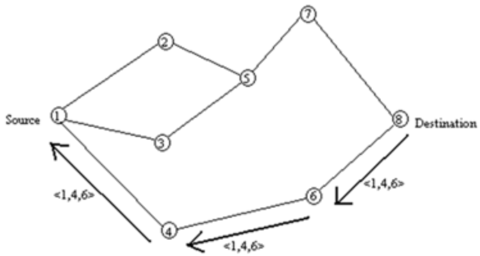
- Ad Hoc Networks: Network of nodes that just happens to be near each other. Nodes can move and start/stop working arbitrarily. Examples:
  - Military vehicles on battlefield. (No infrastructure.)
  - A fleet of ships at sea. (All moving all the time)
  - A gathering of people with notebook computers.
- Fixed-wire Routing techniques can’t be used anymore
- Several methods have been proposed for ad hoc routing. Examples are:
  - Dynamic Source Routing (DSR)
  - Ad hoc On-demand Distance Vector (AODV) Routing
- Key issue for Ad Hoc routing methods is that routes are calculated on-demand to save bandwidth and battery life

Dynamic Source Routing (DSR)

- On Demand routing protocol designed to restrict the routing overhead in ad hoc networks
- Route Request (RRQ) packets are generated by source and flooded in the network to inquire about a possible route to destination
- Each Route Request carries a sequence number generated by the source node and the path it has traversed.



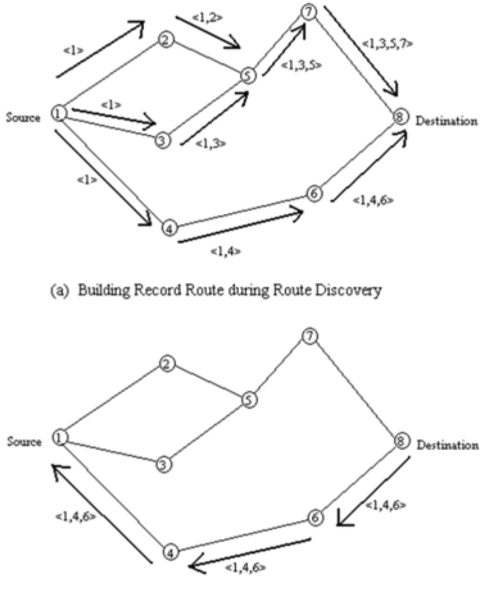
(a) Building Record Route during Route Discovery



(b) Propogation of Route Reply with the Route Record

Dynamic Source Routing (DSR)

- Each node, upon receiving a RRQ packet:
  - Checks the sequence number, records it and checks it to avoid forwarding duplicates
  - Checks time to live (TTL) of the packet to avoid forwarding old packets.
  - Adds its own ID to the packet
  - Rebroadcasts the packet to its neighbors
- Destination node replies to the source node using the trace ID of the first arrived RRQ packet.
- Source sends its data packets with a header containing the entire routing path. (That is why it is called source routing)



(b) Propogation of Route Reply with the Route Record

AODV

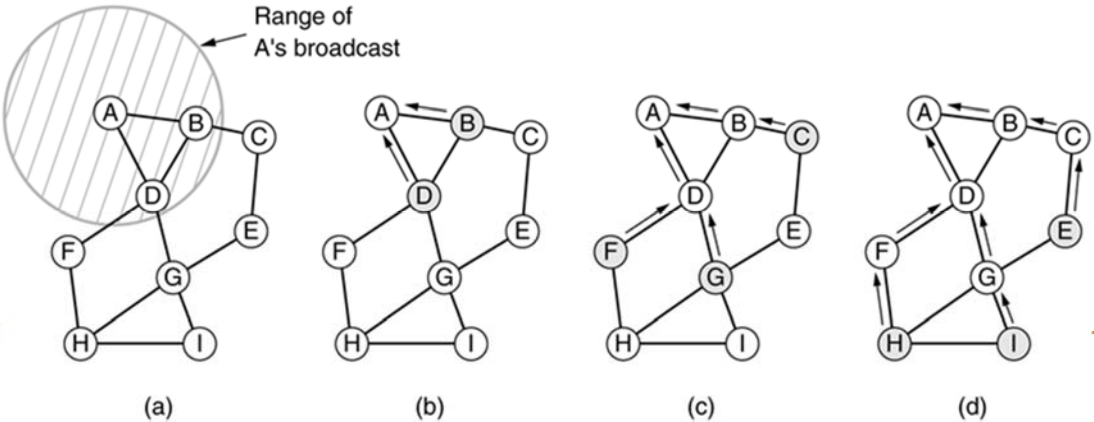
- Node A constructs a ROUTE REQUEST (RRQ) packet and broadcasts it.
- RRQ Contains:
  - Request ID: a local counter maintained separately by each node, incremented for each RRQ packet it sends
  - {Source Address, Request ID} uniquely identify the RRQ packet
  - Destination Sequence Number (Dest_Seq#): Number identifying the last known value of Route Reply (RRP) from the desired destination
- Each node that hears the RRQ discards it if it is a duplicate
- Then it looks in the route table. If there is a fresh route to the destination, it sends a RRP to the source with the info (Saying: Send the packet to me and I'll route it!)
- Fresh route: Dest_Seq# of the RRQ is larger than the Dest_Seq# that exist in its routing table
- If it does not know the route, it increments the Hop count and re-broadcasts it.
- It also extracts the source data and uses it for reverse routing (Everybody knows where is A)

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

Source address	Destination address	Destination sequence #	Hop count	Lifetime

AODV

- The message goes up to the destination (In this example, node I)
- I builds a RRP, where the Hop count, source and destination address are copied from RRQ,
- Dest_Seq# is copied from its current local counter value
- Life Time: Controls how long the route is valid
- Packet is sent back to the source address.
- In the reverse path, every node knows how to get back to the source.

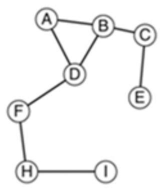


AODV Route Maintenance

- Routing info should be updated regularly because of node's changing status
- Periodically, each node sends a Hello message to its neighbors
- If no response to Hello or no response to a data packet, assume the neighbor is dead
- Active neighbors of node D: The neighbors that depend on D for routing their packets.
- When a neighbor goes down, the affected active neighbors are informed of the event. Local routing table is modified to delete the route passing through the dead node.
- Each of those nodes in turn would notify their affected active neighbors of the event.
- The routing tables are modified all over the nodes and invalid routes are purged.

Dest.	Next hop	Distance	Active neighbors	Other fields
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	
G	G	1	A, B	
H	F	2	A, B	
I	G	2	A, B	

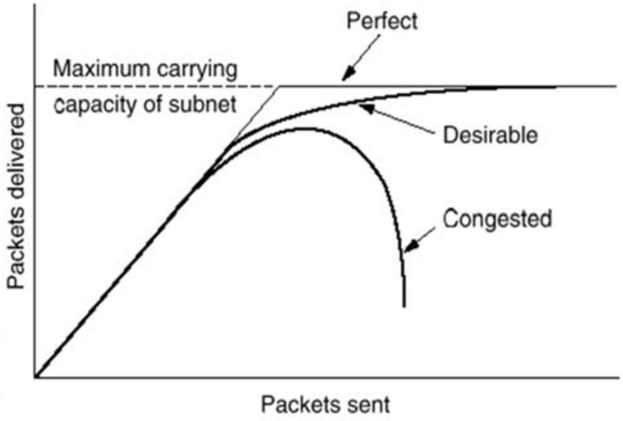
(a)



(b)

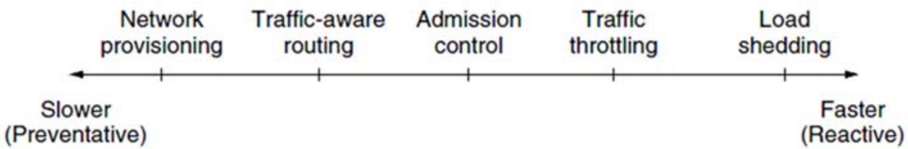
Congestion control

- Problem: Network performance severely degrades at high load
- Reasons:
  - Mismatch between parts of system
  - Router CPUs are too slow
  - Lines too slow
  - Sudden burst of traffic
  - N to 1 communication: ***Hot-spot***
  - Bottleneck in network: e.g. root of tree network
- Result: packets get lost or get timed out => retransmissions => collapse of network throughput



Congestion Control

- What is the difference between congestion control and flow control
  - Congestion control
    - Global issue (network)
    - Indirect feedback
    - Sometimes slowdown messages are used
  - Flow control
    - Local issue (line, point-to-point connection)
    - Direct feedback from neighbor
- Time Scale of applying congestion control techniques



Principles of Congestion Control

- Systems can be controlled by an open loop or a closed loop mechanism
- Open loop: no feedback; solve problems at design time (to make sure they never occur; worst case design)
- Open loop issues:
  - No regard to current state of network
  - When to accept or discard new traffic? (Admission Control)
- Closed loop: Monitor the network on congestion
  - Metrics: % discarded packets, queue length, # timed out packets, average delay
- Pass this information to right place
  - Feedback packets: increase load
  - Reserved bit field in header of every packet
  - Probe packets can be sent periodically (ask about congestion)
- Adjust system operation
  - Avoid oscillation (proper response time needed)
  - Increase number of resources
  - Decrease load

Congestion Prevention

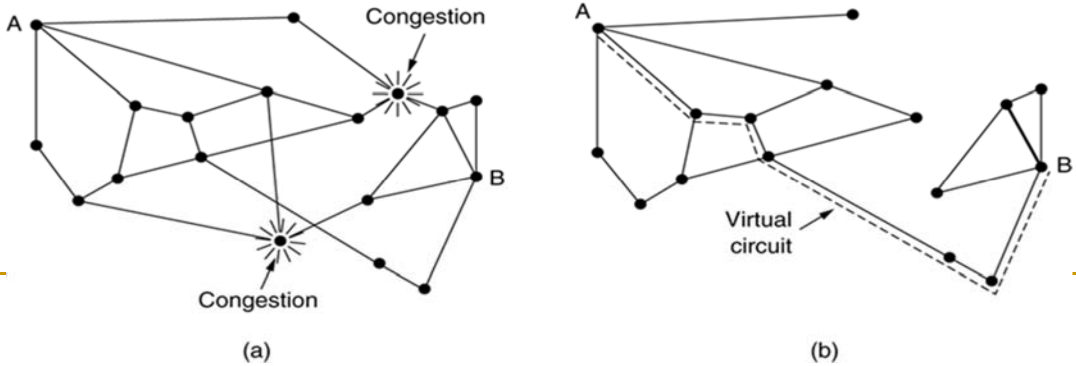
- Open loop policies can be applied at different layers
- ***Data link layer policies***
  - Retransmission creates extra packets
    - Tune time out time
    - Use selective repeat instead of go back n
  - Acknowledgement policy
    - Use piggyback
  - Tight flow control (small window)
- ***Network layer policies***:
  - Any congestion control policies work only with virtual circuits, e.g.. reserving space
  - Routing
    - Spread traffic
    - Avoid hot spots
  - Discard policies
    - Remove long life-time packets
- ***Transport layer policies***:
  - Similar as previous policies
  - Choose proper time-out value for sender

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Fig. 5-23. Policies that affect congestion.

Congestion Control

- Congestion Control in Virtual Circuit subnets:
  - Use Admission Control to prevent congestion
    - If there is congestion, don't accept new traffic.
    - Reserve the necessary resource along the data path to prevent congestion
  - Negotiate a new VC that does not pass through the congested area



Congestion Control in Datagram subnets

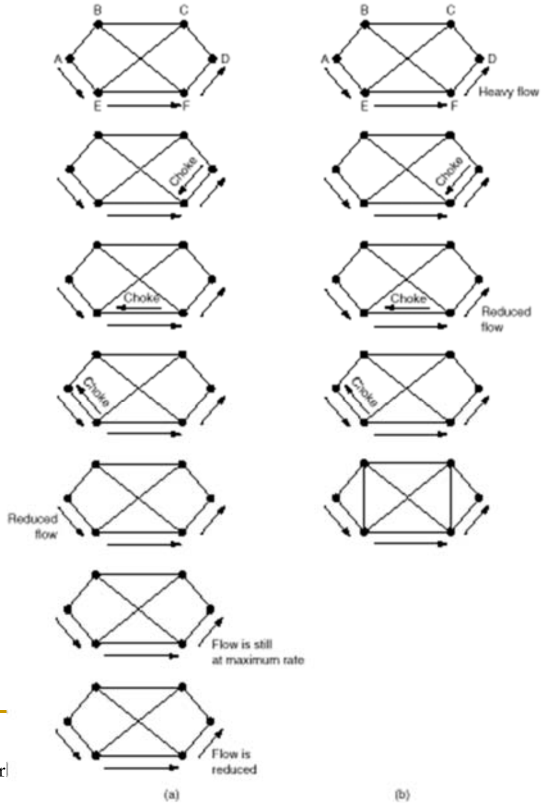
- Estimate congestion u by sampling certain activity *f* (e.g. Line utilization) on each outgoing line
- Calculate u using:

$$u_{new} = au_{old} + (1-a)f$$

- Warning bit
  - Set a bit in the ip payload indicating that there is congestion along the data path. (Give warning when u > threshold)
  - The transport entity copies this bit in the acknowledge and send it back to the source
  - The source can estimate the congestion by measuring the fraction of ACKs it receive with a congestion bit indication

Congestion Control in Datagram subnets

- Choke packets
  - Don’t wait for the destination to act. It may be too late!
  - Each router sends a choke packet to source when in warning state.
  - Source reduces the load by %X. waits for a period of time and if there was no more choke packets, goes back to normal operation
- Hop-by-hop choke packets
  - Choke packets may take a long time to cause effect
  - Solution: Choke packets should not effect the source host only, but also intermediate routers
  - Routers reduce traffic in certain direction
  - More buffer space required at routers !!



Buffer Management

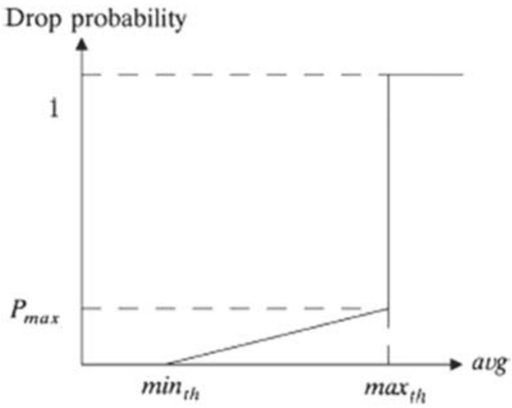
- Load Shedding
  - Allow routers to discard packets
  - What‘s the selection criterion ?
  - Discard old packets: good for video (no retransmission)
  - Discard new packets: good for file transfer (with go back n protocol)
  - Use packet priority
    - ATM supports a 1-bit priority field
    - Users should pay for high priority
  - Remark: measurements show that routers should not wait too long with discarding packets

Buffer Management

- It is best to drop packets as soon as there is congestion
- If there is congestion in a particular port, it is difficult to know whose fault it is (Or who is the source that has caused the congestion)
- Don’t send choke packets as it can add to the congestion
- TCP would slow down if packets are dropped (Reduced traffic rate from the source). This is implicit feedback from the network about its status.

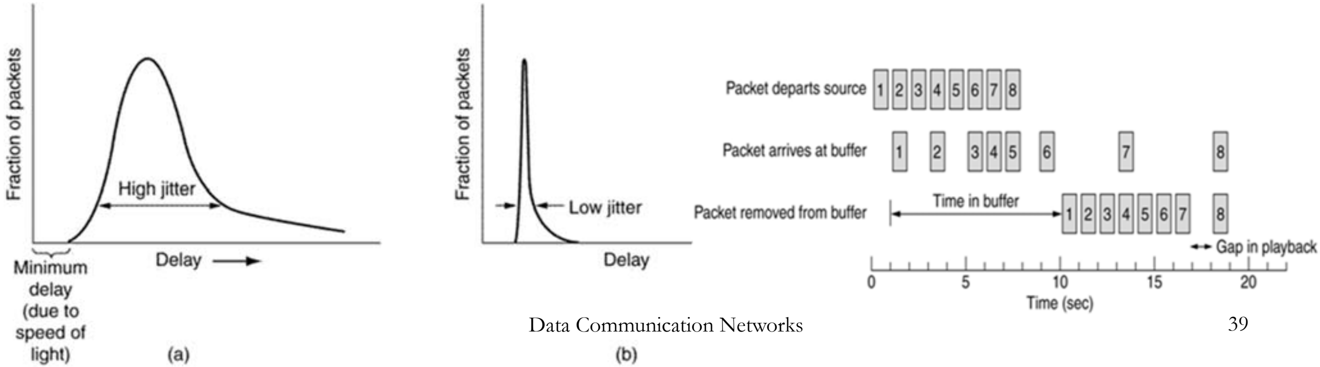
Random Early Detection

- This is a proactive approach in which the router discards one or more packets before the buffer becomes completely full
- Average Queue Length (*avg*): Moving average of the Queue length in the past
- Packets are marked with a probability P as a function of *avg*
- The probability that a packet is marked from a particular connection is roughly proportional to that connection's share of the bandwidth at the router.
- The minimum and maximum thresholds min_{th} and max_{th} are determined by the desired average queue size.
- The average queue size that makes the desired tradeoffs such as the tradeoff between maximizing throughput and minimizing delay depends on network characteristics



Jitter Control

- For audio/video streaming applications, variance of delay (Jitter) should be controlled
- To control the jitter within the network
  - Packet should contain timing information
  - At any router, the time stamp of the packet is checked to see how much the packet is ahead or behind its schedule
  - If ahead of schedule, the packet is held for a while. If late, the packet is given processing priority.
- Jitter can also be controlled at the destination (much more common)
- Buffering at the destination can be used to reduce the effects of network-induced Jitter on the packet arrival time.



Quality of Service (QoS)

- Multimedia traffic requires guarantee on the quality of service
- Flow: a stream of packets from a source to a destination
- Flow QoS parameters:
  - Reliability
  - Rate/Throughput
  - Delay/Jitter
- How to achieve QoS?
  - No single solution.
  - Multiple techniques at multiple layers
- Over-provisioning:
  - Design the network with extra capacity.
  - Very expensive
  - Common among relaxed wealthy organizations!

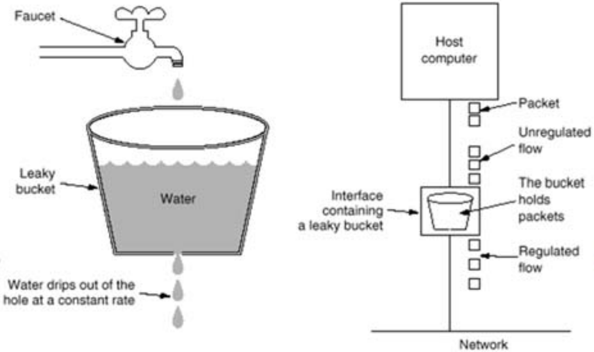
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Traffic Shaping

- Traffic shaping is used to avoid bursty traffic
  - Regulate the average rate on server side
  - Negotiate a contract between sender and network. (Service Level Agreement (SLA))
  - Specially important for real-time applications
- ***Traffic control***
  - Leaky bucket
  - Token bucket
  - Hybrid solution

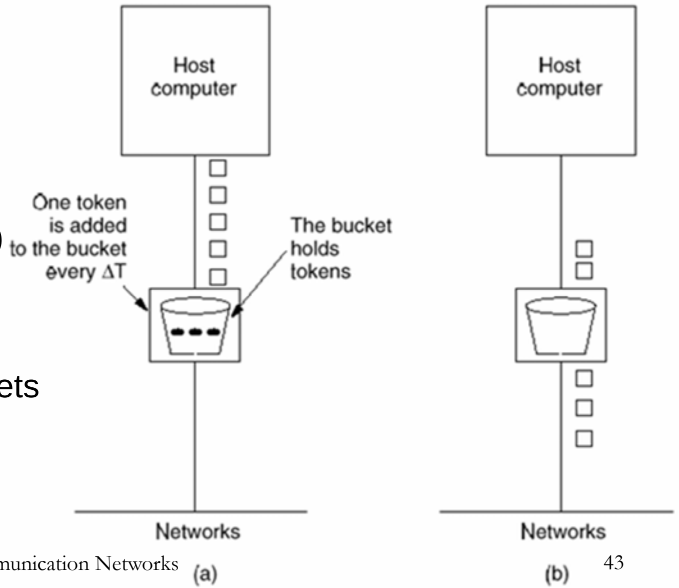
Traffic Shaping – Leaky Bucket

- Leaky bucket algorithm
  - Sender puts packets / bytes in a queue
  - When queue is full the input is discarded
  - Queue output data at constant packet or byte rate
  - Equal to a single-server queuing system with constant service time
- Implementation: use saturating byte counter
  - Every tick increase counter X (up till max value)
  - If packet with length L comes in and X>L then X = X-L, else wait



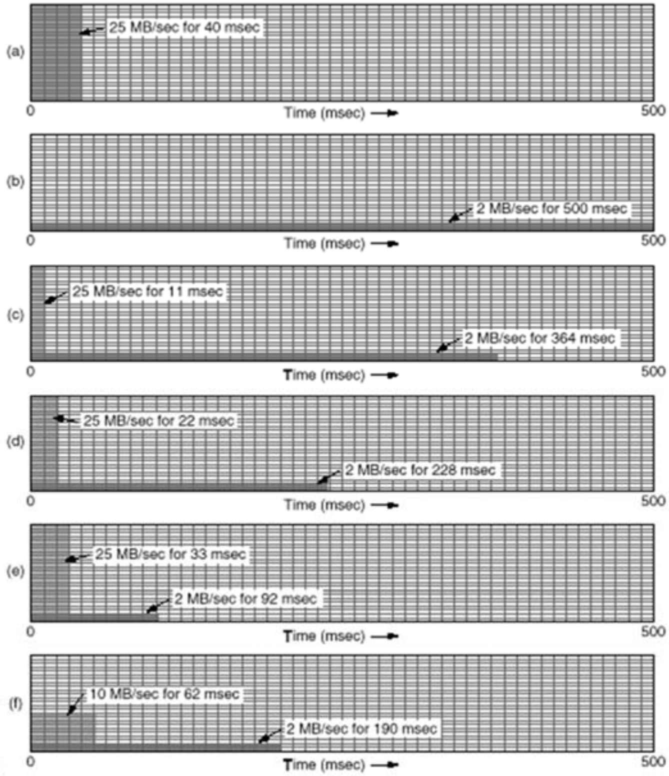
Traffic Shaping -Token Bucket

- Leaky bucket algorithm is too rigid
  - Does not allow temporary larger output rate
  - Constant output rate
- Token bucket algorithm
  - Bucket holds tokens, generated 1 per T sec. up till certain max. value
  - Idle host time builds up 'transmission rights' (= tokens)
  - Packets are never discarded
- *Hybrid solution:*
  - Leaky bucket after token buckets



Traffic Control Example

- (a) Input to a leaky bucket.
- (b) Output from a leaky bucket.
- Output from a token bucket with capacities of
  - (c) 250 KB,
  - (d) 500 KB,
  - (e) 750 KB,
  - (f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket.



Resource Reservation

- Resource Reservation
  - To provide QoS, some sort of fixed path for the flow seems to be necessary
  - The idea: Reserve critical resources along the path of a flow
  - Critical Resources:
    - Bandwidth
    - Buffer space
    - CPU processing cycle (This can be tricky)

- Example for resource relation:
  - Mean processing capacity of the processor, $\mu=10^6$ packets/sec
  - Mean arrival rate of packets to the processor (Poisson distribution), $\lambda=0.95\times 10^6$ packets/sec
  - Mean delay experienced by a packet is:

$$\rho = \frac{\lambda}{\mu} \Rightarrow T = \frac{1}{\mu} \times \frac{1}{1-\rho}$$

$$T = 1\mu\sec \times \frac{1}{1-0.95} = 20\mu\sec$$

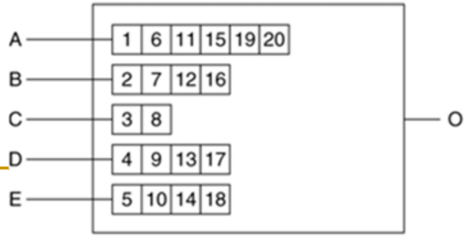
  - So, the port bandwidth, buffer space requirement and processing capacity all are related and impact QoS parameters.

Admission Control

- The flow can be specified in multiple ways.
- Sender generates the flow spec and sends it to the destination
- Routers along a flow-path will study the flow parameters, reserve enough resources and reduce the requested level of service if they can’t meet it.
- At the other end, a decision will be made on the provided QoS to the flow (control the flow admission by reservation of resources)
- An example of flow specification (RFC2210):
  - Token Bucket Rate (Bytes/sec) (Maximum sustained transmitter rate)
  - Token bucket size (Bytes) (Maximum burst size)
  - Peak data rate (Bytes/sec)
  - Minimum packet size (Bytes)
    - CPU cycles are used for each processed packet regardless of its size. This will help the router in its CPU allocation decision
  - Maximum packet size (Bytes)
    - Used for buffer allocation decision

Quality of Service (QoS)

- Packet Scheduling
  - If multiple sources compete for the same destination port of a router, then the most aggressive sender can block all others
  - To remedy that, Fair Queuing algorithm was proposed:
    - Go round robin around among senders, send one packet of each source on the destination port.
  - What if packet sizes are different?
    - Go round robin, byte by byte on packets. When a packet got all its credits, send it out
  - What if traffic sources are not all equal?
    - Assign priority to some flows and give them extra credit in queue processing.



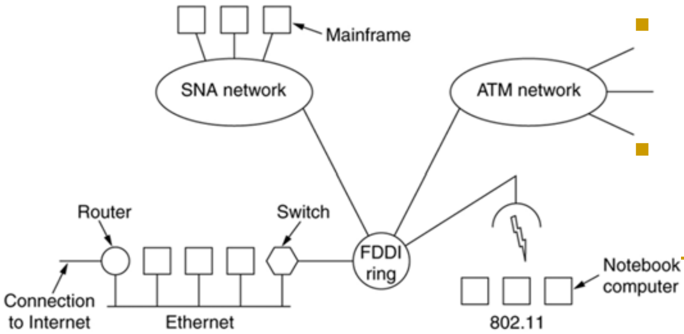
(a)

Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

Internetworking

- How to connect and route on networks using different protocol stacks
  - Different protocols will always exist (vendor policies)
  - Cheaper networks encourage variety (it becomes a departmental decision)
- Interconnection of different networks is called an **internet**



- Various boxes connecting networks or segments are used:
- Physical layer: Repeaters
  - Copy individual bits between cable segments
- Data link layer: Bridges, L2 Switches
  - Store and forward data link frames between LANs
- Network layer: Multi-protocol routers
  - Forward packets between dissimilar networks (often called gateways)
- Transport layer: Gateways
  - Connect byte streams in the transport layer e.g. TCP socket connection
- Application gateways:
  - Proxy servers
  - Firewall

Internetworking

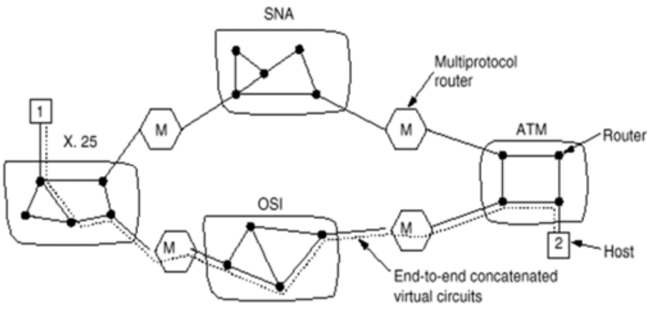
- How networks differ?
  - Connectionless service may change the order of packets
  - Protocol conversion: e.g.. From IP to IPX
  - Address conversion (range, hierarchy)
  - Quality of service may or may not be provided
  - Different packet length
  - Different methods (if at all) for flow control, security, accounting, etc.

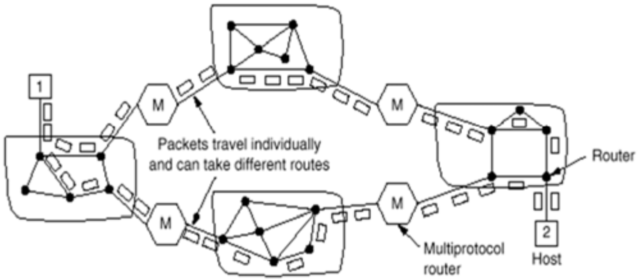
Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present 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, token bucket, RED, 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

Internetworking

- **Concatenated Virtual Circuits**
  - Connection is a concatenation of virtual circuits over different networks
  - Gateway records VC number and builds new VC to router in next subnet
  - All packets traverse same path
  - Works only if protocols do not differ too much e.g. a reliable connection can only be guaranteed if all subnets handle reliable communication
  - Concatenation of virtual circuits is also common in the transport layer e.g.. a bit pipe between OSI and TCP

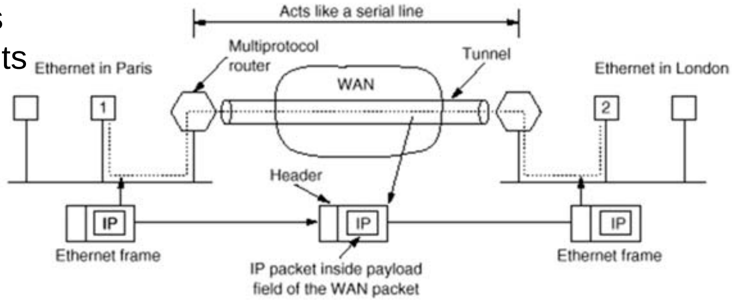
- **Datagram service spanning an internet (multiple networks)**
  - Packets may travel different routes
  - Problem:
    - Connect different network protocols: almost not possible
    - Format conversion is rarely attempted
    - Addresses difficult to re-map





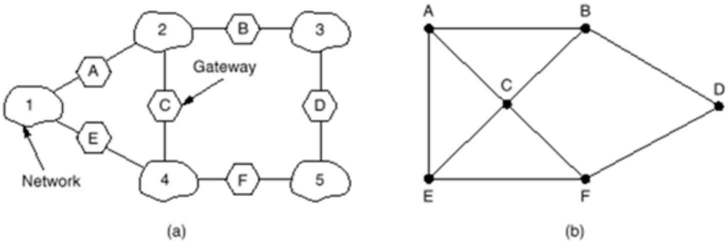
Connectionless internetworking

- Advantages of concatenated VC
  - Buffer space reservation
  - Guaranteed sequencing
  - Short headers
- Advantages of datagram internetworking services
  - Adaptive routing
  - Can be used over subnets that do not support virtual circuits (Many LANs & mobile subnets do not support VCs)
- Tunnelling
  - If source and destination hosts are on same type of network but there is a different network in between, *tunnelling* can be used
  - Packet encapsulation performed by multi protocol router



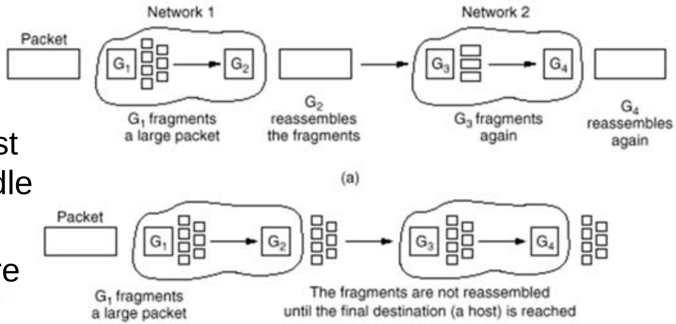
Internetwork Routing

- Distinguish single subnet and internetwork routing
- In the latter multi protocol routers are ‘talking’ to each other
  - Subnet routing uses interior gateway protocol
  - Internetwork routing uses exterior gateway protocol
  - Each network in an internet is called Autonomous system (AS)
- Packet goes from host via local router and LAN to multi protocol router (on same network)
- This router uses routing tables to choose the next multi protocol route
- Either; forward packet (if possible) or; tunnel packet
- Repeat steps 2 and 3 until final multi-router reached
- Route on local destination LAN to its destination
- Notes: countries may implement ‘forced rules’ e.g.. A packet from Canada to Canada may not cross the border



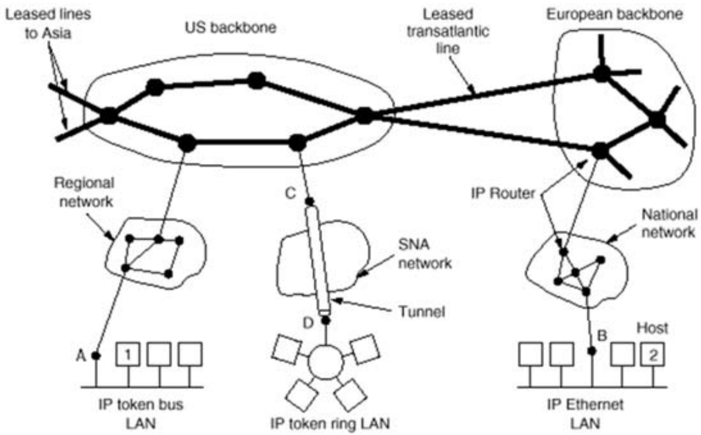
Fragmentation

- All networks impose maximum packet payload size (48 bytes in ATM, 65,515 bytes in IP)
- Multi protocol routers (gateways) may fragment packets. Two options:
  - Transparent :
    - Packet leaves network unfragmented and fragments have to use same route
  - Non-transparent
    - Fragments reassembled at destination host
    - All hosts must support this assemble feature
    - Added overhead for all packets
- Problem: if one fragment lost, end-to-end retransmission required
- Another approach: choose smallest unit which every network can handle
- Always chop large packets to this size and follow the same procedure



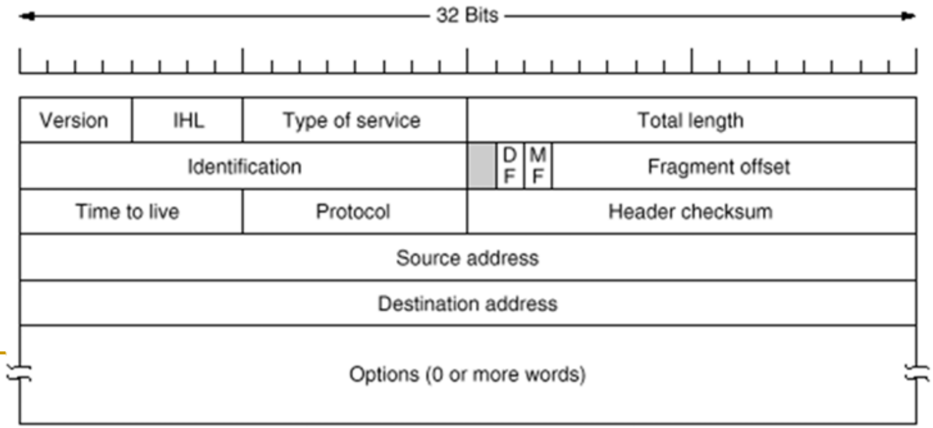
The Internet Network Layer: IP

- Internet: collection of autonomous systems running IP
- IP = Internet Protocol (provides datagram service)
- Transport layer breaks messages up in datagrams of about 1500 bytes
- Datagrams are reassembled at destination host (by Transport layer)



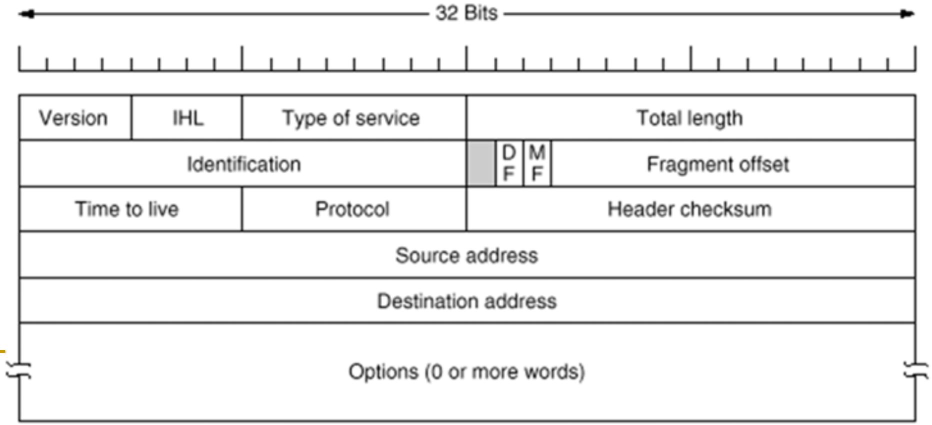
The Internet Network Layer: IP

- IP datagram contains a header of minimal 20 bytes, containing fields for:
  - Version (= 4 for IPv4)
  - IHL: internet header length
  - Type of Service
    - Priorities for delay, throughput and reliability
  - Total length
  - Identification
    - Label fragments belonging to same datagram
  - DF (do not fragment) bit
    - Note: datagrams < 576 bytes should be supported



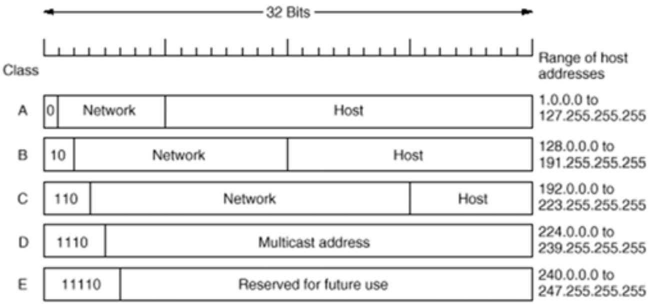
The Internet Network Layer: IP

- MF (more fragments) bit: is this the final fragment?
- Fragment offset: Location of the fragment within the datagram
- TTL (time to live)
  - Counter decremented at each hop
  - Discard datagram when zero: send control message back to source
- Protocol field:
  - Indicate which transport layer protocol to use: e.g.. UDP or TCP
  - Allows direct delivery to right TL protocol handler
- Header checksum
- Source and Destination IP addresses

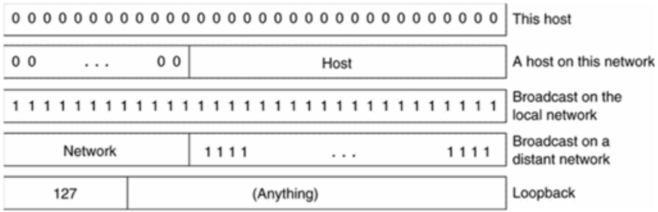


The Internet Network Layer: IP

- Every IP-host (on the net) needs unique 32-bit address
- Five classes: A - E
- Assignment by ICANN (Internet Corporation for Assigned Names and Numbers)
- Dotted decimal notation: e.g. 129.245.96.122

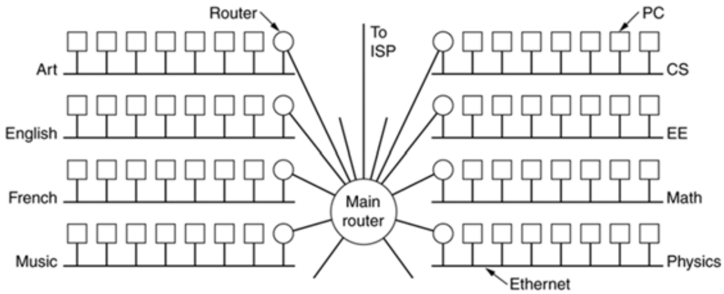


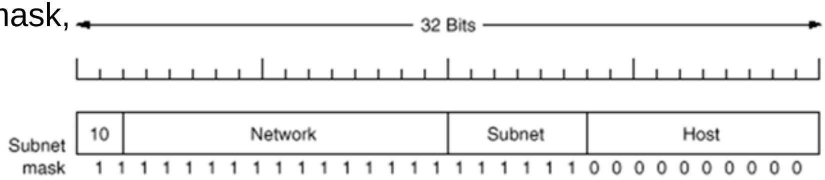
- Special addresses:
  - 0.0.0.0 used during boot-up
  - Network# = 0: own network
  - All ones: broadcast (local or on other network#)
  - 127.x.y.z : loop back for internal testing purposes



Subnets

- A single class A, B or C address refers to one network.
- A company may have different sub-networks (Different departments of Sharif University)
- From outside, it looks like one network. Internally, the network is divided into subnets.
- The main router does not need to know all host addresses. (It requires huge lookup tables, 65,536 in this example)
- Router just looks at the “Subnet” part of the address and forwards it to the appropriate local LAN routers. (Apply subnet mask, lookup the result)





Subnet routing: How is routing performed:

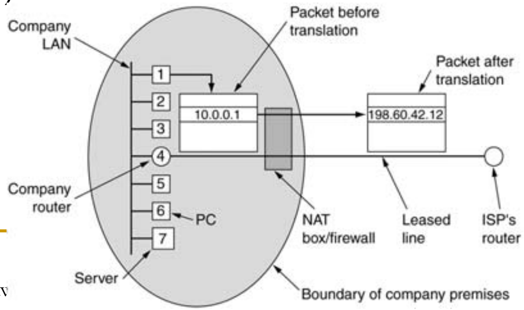
- Normally router contains table with following type of addresses
  - (network#, 0)
  - (this network#, host#)
  - This is a 2-level hierarchy
  - If incoming network address not listed sent packet to default router (with more extensive table)
- With subnetting (for internal use only):
  - Router table contains additional entries:
  - (this network#, subnet#, 0)
  - (this network#, this subnet#, host#)
  - 3-level hierarchy
  - Use host masking to quickly determine subnet
- Decision on how many bits to use for subnet and how many bits for Host addresses is an internal design parameter for each company or university

Classless Inter Domain Routing

- Problem: Internet is running out of addresses
- C class is often too small, but B class too large
- Solution: CIDR
  - Idea: assign the remaining addresses on a variable-sized blocks, without regard to classes
  - Use Masking to quickly determine outgoing link
- Class C address space divided into 4 parts, e.g. Europe gets 194.0.0.0 to 195.255.255.255 (C addresses) = 32 Million addresses
- Masking example:
  - Assume company needs 2k addresses, and gets: 194.24.0.0 - 194.24.7.255 (8 C blocks)
  - Mask = 255.255.248.0 (248 = 1111.1000b)
- Routing tables contain (base address, mask) triples
- Incoming address is masked with each entry
- If: (address AND mask) = base address, send packet to appropriate router

Network Address Translation

- An ISP with dial up users can dynamically assign IP to users
- ADSL users or business LAN users expect to be on-line all the time, so they need permanent IP addresses
- Network Address Translation:
- Allow each host in the internal network to maintain a permanent IP address.
- Replace the {TCP Source, IP source} addresses with {virtual TCP source, Virtual IP source} of each outgoing IP packet
- On the reply, use the combination of TCP and IP addresses to lookup the local host
- Reserved NAT addresses
  - IP packets with these addresses should not appear on internet
  - 10.0.0.0 – 10.255.255.255/8 (16,772,216 hosts)
  - 172.16.0.0 – 172.31.255.255/12 (1,0478,576 hosts)
  - 192.168.0.0 – 192.168.255.255/16 (65,636 hosts)



Internet Control Protocols

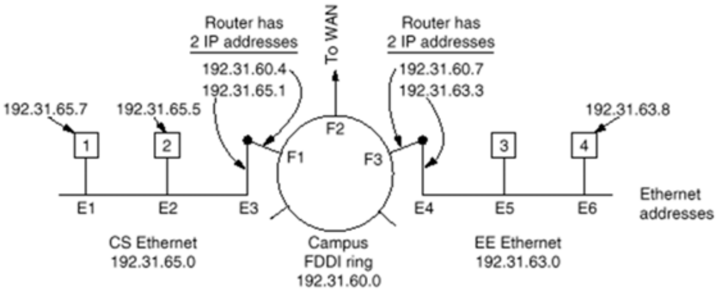
- Control protocols: ICMP, ARP, and DHCP
- ICMP : internet control message protocol
- Report unexpected events (e.g.. discovered routers)
- Messages embedded in IP packet
- Message types:
  - Destination unreachable
  - Time exceeded
  - Redirected : inform source that message seems to follow wrong route
  - Echo: are you alive ?

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

Fig. 5-50. The principal ICMP message types.

Internet Control Protocols

- ARP: How to map IP <=> Ethernet addresses (4 bytes to 6 bytes)
- All Ethernet cards have a unique address
- Host sends broadcast on LAN : give me Ethernet address of W, X, Y, Z
- Address mappings are cached temporarily
- Used by almost every internet host



- Send message to e.g. test@sharif.edu
- Lookup name using DNS (domain name server)
- DNS returns IP address
- Two possibilities:
- Local address:
  - Use ARP to get Ethernet address
- Remote address:
  - Use ARP message; gateway responds
  - send packet directly to gateway. Host has to know default (gateway) Ethernet address.
  - Intermediate network encapsulates packet
  - On destination LAN ARP is used again to get destination Ethernet address

Dynamic Host Configuration Protocol (DHCP)

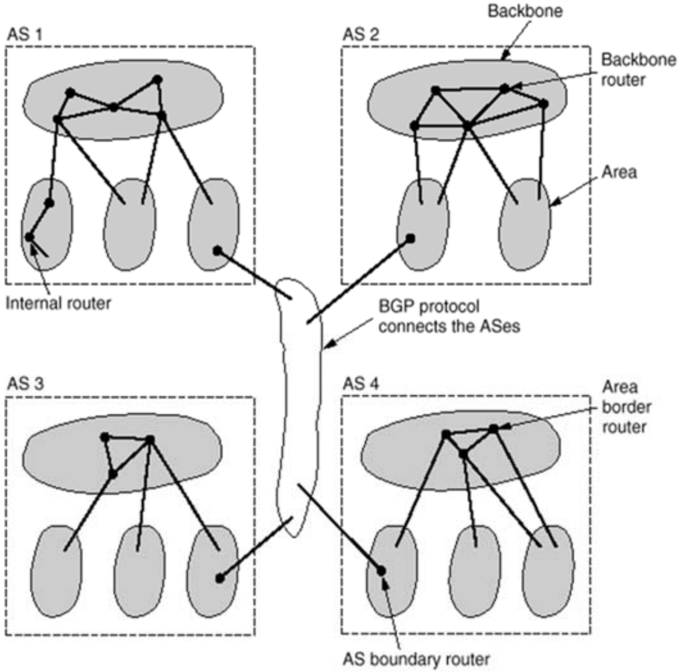
- DHCP: Dynamic Host Configuration Protocol (RFC 2131, 2132)
  - A client-server architecture for configuration of host IP addresses
  - A newly booted workstation broadcasts a DHCP DISCOVER packet
  - The DHCP server replies by giving it an IP address
  - The IP address is leased for a period of time. Host must ask for renewal of the lease before it expires. If the lease expires, the IP is returned to the pool of available IP addresses.
  - If DHCP server is not on the same LAN, a DHCP proxy can relay the DHCP messages over network to the DHCP server.
  - DHCP proxy should know the IP address of the DHCP server.

Routing on the internet

- Routing categories:
  - Interior routing: within AS (autonomous subsystem; managed by particular organization)
  - Exterior routing: between ASes
- Interior gateway protocol
  - Till 1979: distance vector protocol
    - Slow convergence (count to infinity problem)
    - Not usable for large systems
  - Since 1979: link state protocol
  - New standard (1990) OSPF: open shortest path first
- Open Shortest path First
  - Open standard
  - Support variety of distance (cost) metrics
  - Dynamic routing (adaptive)
  - Support different 'service types' e.g. real-time traffic
  - Load balancing: split load over multiple lines
  - Support hierarchical systems: a router cannot know the whole topology
  - Security support
  - Tunnelling support

Open Shortest Path First

- OSPF supports 3 types of communication
  - 1. Point-to-point
  - 2. Multi-access with broadcast (LANs)
  - 3. Multi-access without broadcast (WANs)
- (Multi-access = can have multiple routers on network)
- OSPF divides AS into areas
  - An area is a generalization of a subnet
  - Outside an area its details are not visible
  - Every AS has a backbone area (#0)



Open Shortest Path First

- OSPF uses protocol similar to Link State Protocol
- Use 'Hello' packets to learn neighbours
- Periodically flood link state update messages
- Interior routing to other area always via backbone
- OSPF supports multi-metric routing. It uses multiple graphs, one for each metric and finds the shortest path of each graph
- Most often used metrics:
  - Throughput
  - Delay
  - Reliability

Open Shortest Path First

- Exterior Routing
- BGP = Border Gateway Protocol
  - Implies ‘politics’
  - policies are manually configured into each BGP router
  - Uses variant of distance vector protocol
  - Tables exchanged certain (distance, path) tuples
  - By exchanging the whole path, the ‘count to infinity’ problem can be solved

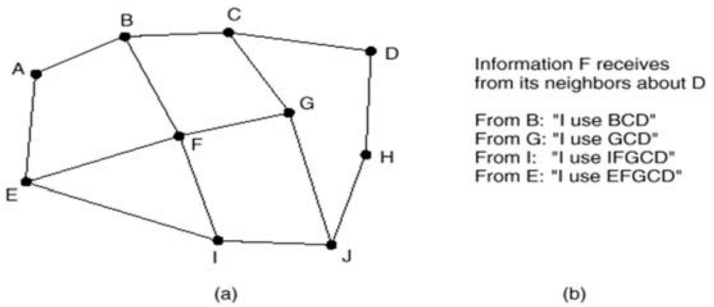


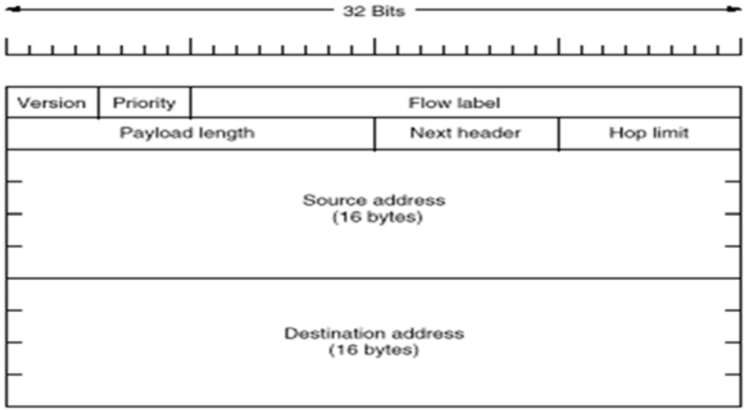
Fig. 5-55. (a) A set of BGP routers. (b) Information sent to *F*.

IPv6 Protocol

- To expand the feature set and capabilities of IP, in 1990 work on new protocol started with these requirements:
  - Support billions of hosts
  - Reduce routing table size
  - Simplify protocol (reduces processing power requirement in routers)
  - Better security (authentication and privacy)
  - Multicasting
  - Mobile hosts
  - Allow protocol to evolve (room for expansion)
  - Coexist with IPv4
- Important Features:
  - Longer addresses than IPv4: 16 byte addresses
  - Header simplification (fixed size of 40 bytes)
  - Better support for options
  - Improved Security capabilities
  - Support for packet priority
  - Support for pseudo connections (using the flow label)
  - Support very long datagrams

IPv6 Protocol Features

- Version: Always set to 6
- Priority: Handle packets with different classes of service
  - 0-7 non-real-time (flow controlled)
  - 8-15 real-time (no flow control)
- Flow: flow label which is used for flow-driven routing
- Payload Length: packets with size >64k are supported
- Next Header: Headers can optionally be extended over multiple packets.

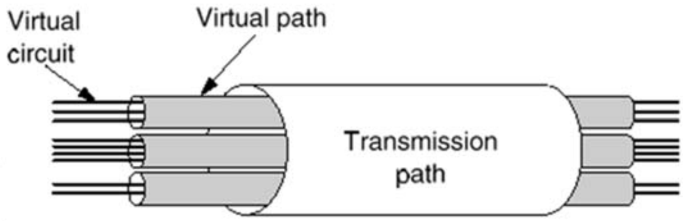


IPv6 Features

- Fragmentation:
  - All routers should support packet size of at least 576 bytes
  - For larger packets, if router can not support it, it is dropped and a notice is sent to the host.
  - It is better if fragmentation is done at host rather than the network

ATM

- Discussion: where to put ATM?
- Internet (IP) community: ATM is just a carrier for us (= IP), so it should be below the network layer
- However: ATM involves routing, therefore a network layer protocol
- ATM protocol
  - Connection oriented
  - Virtual circuits are unidirectional
  - Connection is pair of VCs with same number
  - Unreliable (no acknowledgement)
  - In-order arrival of cells
  - Two-level connection hierarchy
  - level-1: virtual path (=bundle of VCs);
    level-2: virtual connection



ATM Cell header

- Two interfaces with slightly different header formats:
  - UNI : user-network interface
  - NNI : network-network interface
- Header fields
  - GFC : (UNI only)
    - Flow control between host and network (not really used)
  - VPI (virtual path ID) 8 bits
  - VCI (virtual circuits ID) 16 bits
  - PTI (payload type)
    - 4 types are user supplied
    - 4 types are network supplied e.g., a 000 PTI may change (by network) into 010 to warn for congestion
  - CLP bit (cell loss priority)
  - HEC (header checksum)
    - Corrects all single bit errors
    - Detects 90% of multi-bit errors

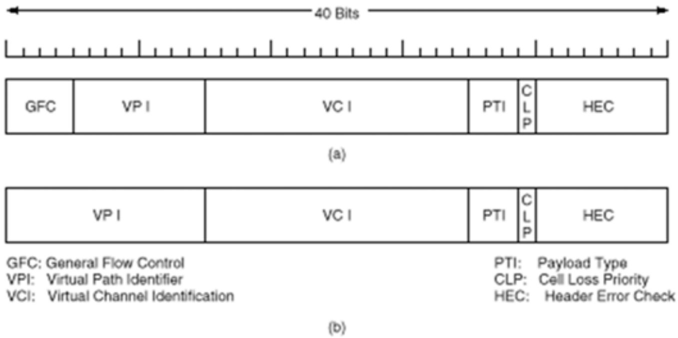


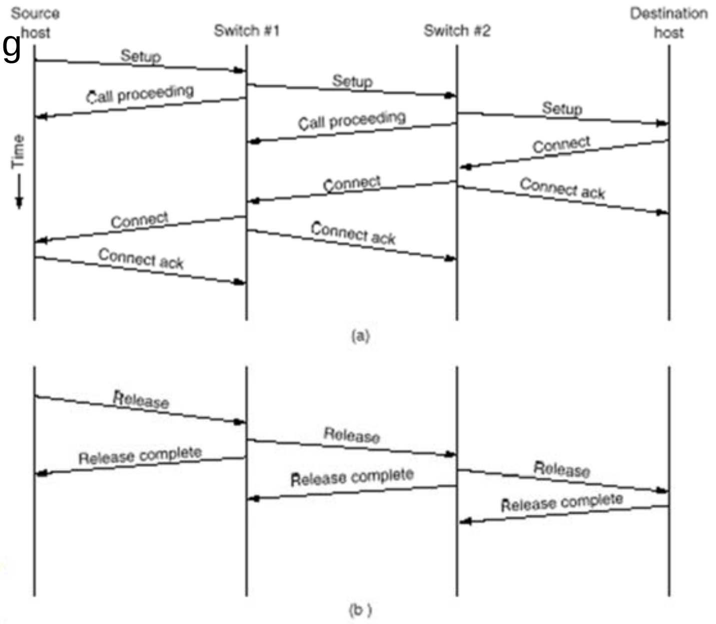
Fig. 5-62. (a) The ATM layer header at the UNI. (b) The ATM layer header at the NNI.

Payload type	Meaning
000	User data cell, no congestion, cell type 0
001	User data cell, no congestion, cell type 1
010	User data cell, congestion experienced, cell type 0
011	User data cell, congestion experienced, cell type 1
100	Maintenance information between adjacent switches
101	Maintenance information between source and destination switches
110	Resource Management cell (used for ABR congestion control)
111	Reserved for future function

Fig. 5-63. Values of the *PTI* field.

ATM Connection setup

- How to set up a connection?
  - Put request on VP 0, VC 5
  - Network returns new VC which can be used for setting up the connection
  - Send SETUP message to destination using new VC channel
  - Destination responds with CONNECT Addresses: 20 bytes
  - Release: send RELEASE message: propagates to destination



ATM Routing

- Standard does not specify particular routing algorithm. The idea is:
  - VPI based routing between switches
  - VCI based routing between switch and final host
- Reason:
  - Additional VC can use same VPI based routing channel ; no new table entries needed
  - Smaller tables (2^{12} instead of 2^{28} entries)
  - Faster routing
  - Quicker re-routing
  - A whole set of VCs can be re-routed by changing a single entry
  - Support for private network (with company owned VPI number)

ATM Routing

- A switch (router) gets requests of the form: (source, destination, line#, VPI)
- Using the destination the switch determines the outgoing line and chooses a free outgoing VPI number
- This information is kept in tables (one for each outgoing link)

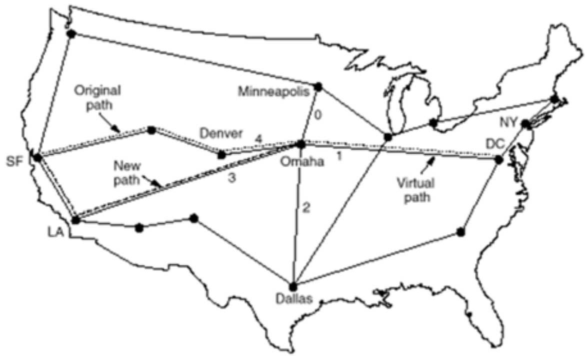


Fig. 5-66. Rerouting a virtual path reroutes all of its virtual circuits.

Incoming VPI	VPI_table for Minn. Outgoing Line	VPI_table for Minn. Outgoing VPI	VPI_table for DC Outgoing Line	VPI_table for DC Outgoing VPI	VPI_table for Dallas Outgoing Line	VPI_table for Dallas Outgoing VPI	VPI_table for LA Outgoing Line	VPI_table for LA Outgoing VPI	VPI_table for Denver Outgoing Line	VPI_table for Denver Outgoing VPI
0										
1	3	1	4	1			0	1	1	1
2	4	5	4	2			1	3	1	2
3			3	2					1	4
4			4	3					1	5
5			4	4					0	2
6										
7										
8										
4095										
	Line 0		Line 1		Line 2		Line 3		Line 4	

Source	Incoming line	Incoming VPI	Destination	Outgoing line	Outgoing VPI	Path:
NY	1	1	SF	4	1	New
NY	1	2	Denver	4	2	New
LA	3	1	Minneapolis	0	1	New
DC	1	3	LA	3	2	New
NY	1	1	SF	4	1	Old
SF	4	3	DC	1	4	New
DC	1	5	SF	4	4	New
NY	1	2	Denver	4	2	Old
SF	4	5	Minneapolis	0	2	New
NY	1	1	SF	4	1	Old

Fig. 5-67. Some routes through the Omaha switch.

ATM Service Categories

- ATM offers 4 types of service:
- CBR : constant bit rate
  - Used to support existing voice channels (like T1, E1)
  - Real-time application may also require this service
- VBR : variable bit rate (guaranteed bandwidth)
  - Real time
  - Average cell delay and its variation must be tightly controlled
  - Non-real time (some jitter can be tolerated) e.g.. multi media email

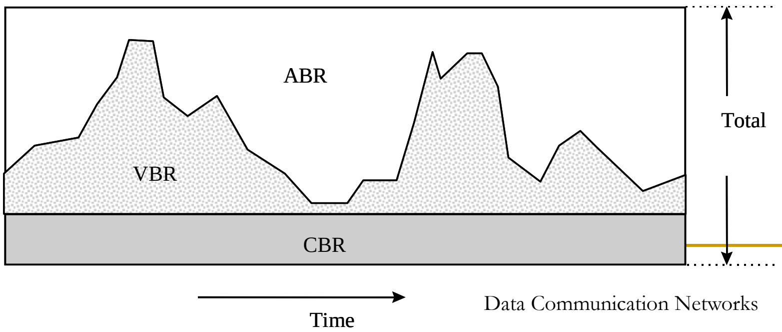
Service characteristic	CBR	RT-VBR	NRT-VBR	ABR	UBR
Bandwidth guarantee	Yes	Yes	Yes	Optional	No
Suitable for real-time traffic	Yes	Yes	No	No	No
Suitable for bursty traffic	No	No	Yes	Yes	Yes
Feedback about congestion	No	No	No	Yes	No

Class	Description	Example
CBR	Constant bit rate	T1 circuit
RT-VBR	Variable bit rate: real time	Real-time videoconferencing
NRT-VBR	Variable bit rate: non-real time	Multimedia email
ABR	Available bit rate	Browsing the Web
UBR	Unspecified bit rate	Background file transfer

Fig. 5-69. The ATM service categories.

ATM Service Categories

- ABR : available bit rate
  - For bursty traffic with known average bandwidth
  - Network provides congestion feedback
  - Used to replace leased lines
- UBR : unspecified bit rate
  - No guarantees or promises about bandwidth
  - In case of congestion UBR cells are discarded without feedback
  - Applications have to do own error and flow control



ATM quality of service QoS

- Standard defines a number of QoS parameters
- For each parameter the worst case value is specified
- The network is expected to meet these values

Parameter	Acronym	Meaning
Peak cell rate	PCR	Maximum rate at which cells will be sent
Sustained cell rate	SCR	The long-term average cell rate
Minimum cell rate	MCR	The minimum acceptable cell rate
Cell delay variation tolerance	CDVT	The maximum acceptable cell jitter
Cell loss ratio	CLR	Fraction of cells lost or delivered too late
Cell transfer delay	CTD	How long delivery takes (mean and maximum)
Cell delay variation	CDV	The variance in cell delivery times
Cell error rate	CER	Fraction of cells delivered with error
Severely-errored cell block ratio	SECBR	Fraction of blocks garbled
Cell misinsertion rate	CMR	Fraction of cells delivered to wrong destination

Fig. 5-71. Some of the quality of service parameters.

ATM Quality of Service (QoS) Categories

- Time/Delay:
  - Cell Transfer Delay: the average transmit time
  - Cell Delay Variation: how uniformly cells arrive
  - Cell Delay Variation Tolerance: CDVT
    - Leaky bucket used to control this Cell Loss Ratio = fraction of non-delivered cells

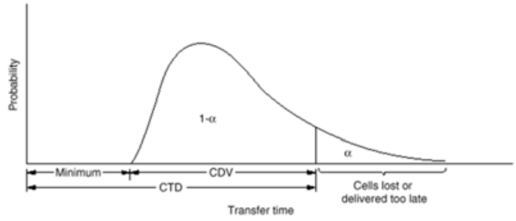
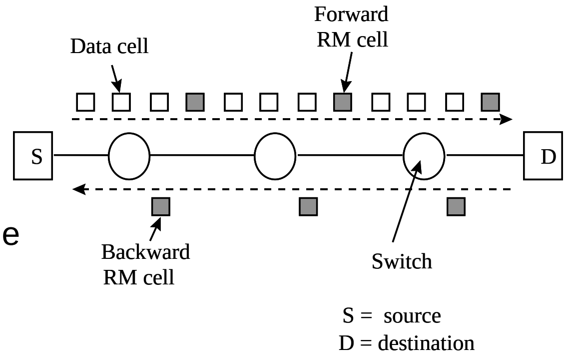


Fig. 5-72. The probability density function for cell arrival times.

- Quality
  - Severely Errored Cell Block Ratio: fraction of N-cell blocks of which M or more cells are delivered with one or more bits wrong
  - Cell Error Ratio : fraction of cells with wrong bits in them
  - Cell Mis-insertion Rate : cells per second delivered at wrong destination
- Rate
  - Peak Cell Rate : PCR
  - Sustained Cell Rate : SCR
  - Minimum Cell Rate : MCR
    - If network can not guarantee this, the connection is not granted
    - Note: ABR with MCR = 0 is equal to UBR

ATM Congestion Control



- ATM Congestion Control Strategies :
  - Admission control
  - Resource reservation: Capacity can be reserved using the SETUP message
- Rate based congestion control
  - For ABR traffic type only
  - A special cell used for resource management cell (RM) is sent by sender and contains the required bandwidth
  - RM cell travels from source to destination and back and during this trip, each switch may reduce this value
  - Upon return, sender knows available bandwidth
  - Incentive : if sender does not reduce transmission rate, cells may be discarded