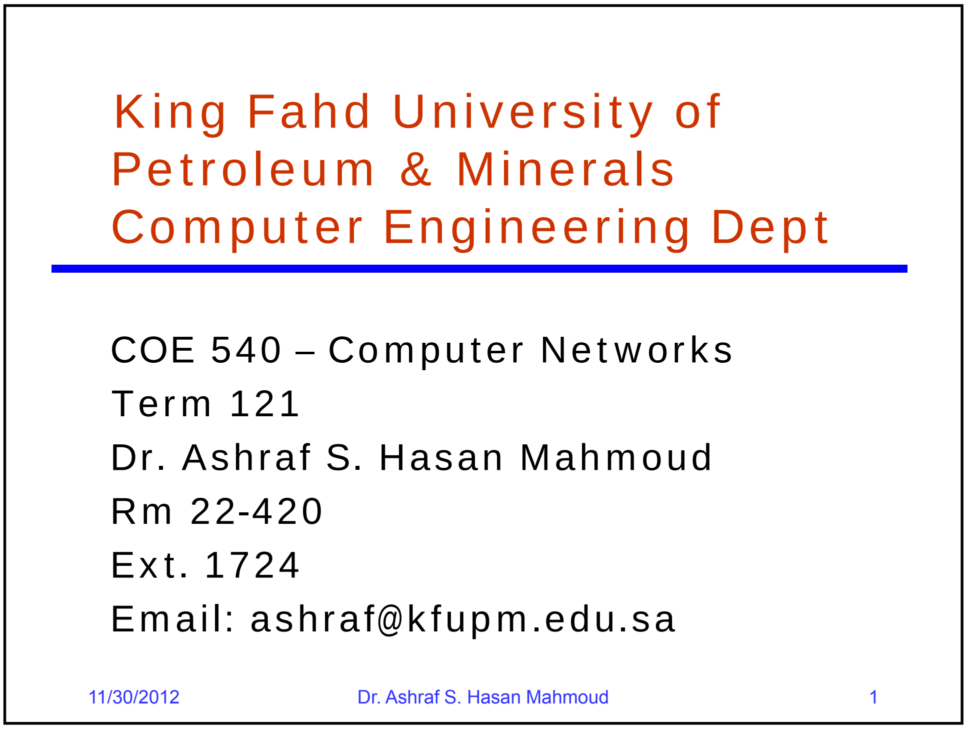
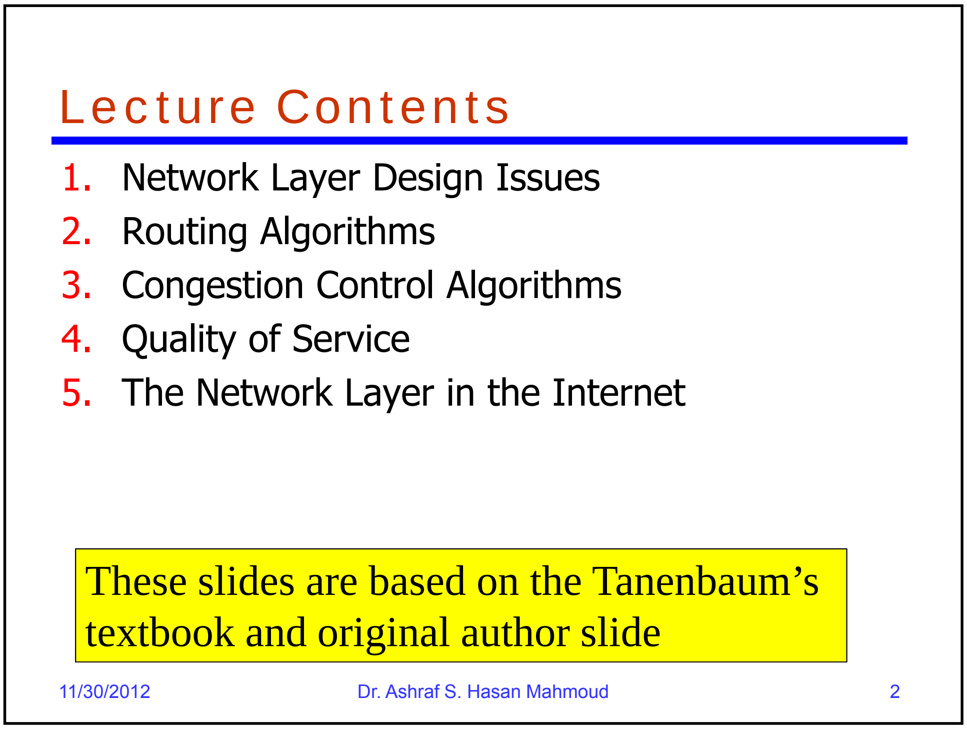


King Fahd University of Petroleum & Minerals Computer Engineering Dept

COE 540 – Computer Networks
Term 121
Dr. Ashraf S. Hasan Mahmoud
Rm 22-420
Ext. 1724
Email: ashraf@kfupm.edu.sa



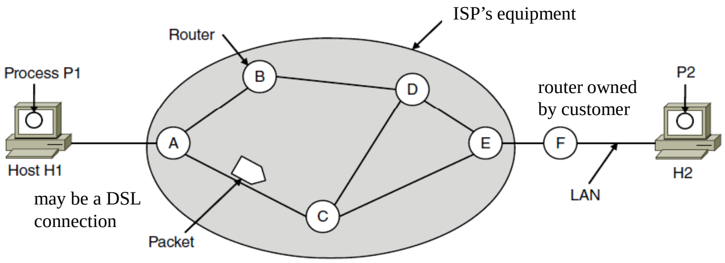
Lecture Contents

1. Network Layer Design Issues
2. Routing Algorithms
3. Congestion Control Algorithms
4. Quality of Service
5. The Network Layer in the Internet

These slides are based on the Tanenbaum's textbook and original author slide

Network Layer Design Issues

- Store-and-forward Packet Switching
  - A host with a packet to send transmits it to the nearest router
  - The packet is stored, verified (checksum), and then routed to the next router along the path leading to the destination
  - Until the packet is delivered to the destination host



The environment of the network layer protocols.

Services Provided to the Transport Layer

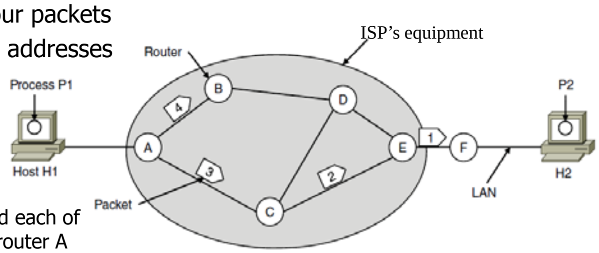
- Transport layer works end-to-end therefore the services should be:
  - Independent of the router technology
  - Shielded from the number, type, and topology of the routers preset
  - The network addresses made available to the transport layer should use uniform numbering plan even across LANs and WANs
- End-to-End argument
  - End hosts must perform error control
  - Network layer provides "PACKET SEND" and "PACKET RECEIVE" primitives only
- ATM/Telephony point of view
- Connection-oriented evolving features of the internet – e.g. MPLS and VLANs

Types Of Networks

- Datagram network
  - Packets (or Datagrams) injected and routed in the network independently
  - Connectionless model
- Virtual-circuit network
  - Path from source router to destination router is determined before data transmission commences – virtual circuit
  - Similar to physical circuits

Datagram Network

- Message from H1 to H2
- Message divided into four packets
- Packets has src and dst addresses



- The PPP layer in H1 will send each of the four packets to the ISP router A
- Routing table per router – (Dest, output port)
- Role of the routing algorithm
  - Packets 1, 2, and 3 are sent through port C of router A – Path ACE
  - Packet 4 is sent through port B of router A – Path ABDE

A's table (initially)

Dest.	Line
A	⊠
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	⊠
B	B
C	C
D	B
E	B
F	B

C's Table

A	A
B	A
C	⊠
D	E
E	E
F	E

E's Table

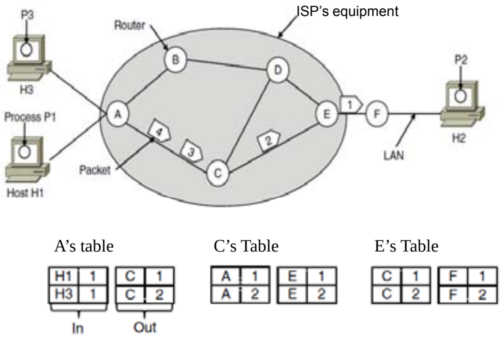
A	C
B	D
C	C
D	D
E	⊠
F	F

Routing within a datagram network

Implementation of Connection-Oriented Service

- H1 to establish connection H2
- Phases: connection setup, data transfer, connection tear-down
- Connection setup – determine a route from source router to destination router

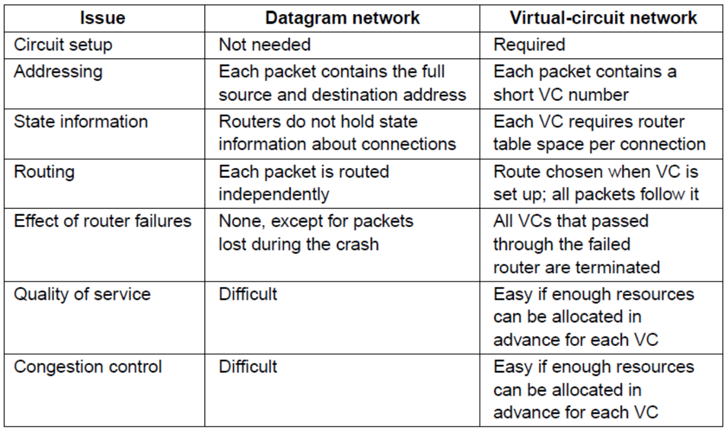
- Route used for all packets belonging t the same connection
- Route/connection stored in routing tables in every router along the path
- Every packet carries a virtual circuit (connection) identifier
- Label (connection identifier) switching at router A for H3's connection
- MultiProtocol Label Switching (MPLS) is used within ISP networks where IP packets are wrapped in an MPLS header having a 20-bit connection identifier



Routing within a virtual-circuit network

Comparison of Virtual-Circuit and Datagram Networks

- Tradeoffs between virtual circuits and datagrams



Issue	Datagram network	Virtual-circuit network
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

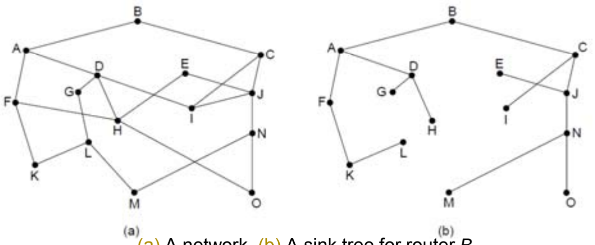
Comparison of datagram and virtual-circuit networks

Routing Algorithms

- Desired properties: correctness, simplicity, robustness, stability, fairness, and efficiency.
- Nonadaptive versus adaptive
  - E.g. adaptive algorithms base their routing decisions on measurements or estimates of the current traffic load and/or topology
- Nonadaptive - Static routing – does not respond to failures
  - Used in situations where the routing decision is very clear (e.g. node E to node F in previous figure)
- Adaptive – Dynamic routing
  - Steps of operation
    - Get information (locally, adjacent node, all routers)
    - Compute new routes
    - Communicate to other routers
  - Different metrics for optimization: distance, number of hops, estimated transit time, etc.

The Optimality Principle (Bellman 1957)

- 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: refer to textbook page 383
- Sink tree – set of all optimal paths from all sources to a given destination.
  - See example for sink tree for router B – number of hops is the used metric
- The goal of the routing algorithm is to compute and use the sink trees for all routers
- The sink tree is not necessarily unique
  - Other tree with the same path lengths may exist
- If we allow ALL possible paths to be chosen ➔ Directed Acyclic Graph (DAG)
  - No loops
- Spanning tree - is a subset of the network that includes all routers but with no loops
  - Sink tree is a spanning tree

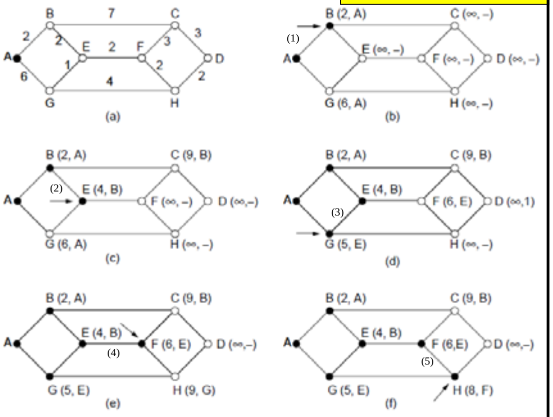


(a) A network. (b) A sink tree for router *B*.

Shortest Path Algorithm (Dijkstra 1959)

Label: (X, Z)
X – distance from source
Z – preceding node

- Finds the shortest path from a source to all possible destinations
- Global view of the network
- Source – node A
- Every node is labeled in (X, Z) where X is the distance from source, Z is the preceding node
- Initially no paths are known – as algorithm proceeds, paths are found and labels are updated
- Example: Find shortest path from A to D



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

A Link-State Routing Algorithm

Slides by Kurose
To use the prescribed notation

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives *forwarding table* for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Dijsktra's Algorithm

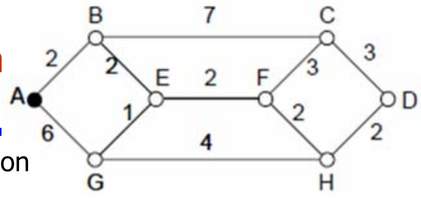
Slides by Kurose
To use the prescribed notation

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

Shortest Path Algorithm (Dijkstra 1959) – cont'd

- Example redone using Kurose's notation



Step	N'	B	C	D	E	F	G	H
0	{A}	(2,A)	(∞,-)	(∞,-)	(∞,-)	(∞,-)	(6, A)	(∞,-)
1	{A, B}	(2, A)	(9, B)	(∞,-)	(4, B)	(∞,-)	(6, A)	(∞,-)
2	{A, B, E}	(2, A)	(9, B)	(∞,-)	(4, B)	(6, E)	(5, E)	(∞,-)
3	{A, B, E, G}	(2, A)	(9, B)	(∞,-)	(4, B)	(6, E)	(5, E)	(9, G)
4	{A, B, E, G, F}	(2, A)	(9, B)	(∞,-)	(4, B)	(6, E)	(5, E)	(8, F)
5	{A, B, E, G, F, H}	(2, A)	(9, B)	(10, H)	(4, B)	(6, E)	(5, E)	(8, F)
6	{A, B, E, G, F, H, C}	(2, A)	(9, B)	(10, H)	(4, B)	(6, E)	(5, E)	(8, F)
7	{A, B, E, G, F, H, C, D}	(2, A)	(9, B)	(10, H)	(4, B)	(6, E)	(5, E)	(8, F)

Flooding

- Every incoming packet is sent on every outgoing link
  - Vast number of duplicate packets
- Measures to limit duplicate packets
  - Hop count
  - Router remembers packets that have already been sent - list of (src router, sequence #) – can be summarized
- Main features:
  - Effective broadcast mechanism - The packet is delivered to EVERY node
  - Very robust – will find a path if one exists
  - Requires very little set up
  - Always chooses the shortest path
- Used a reference when comparing performances of routing algorithms

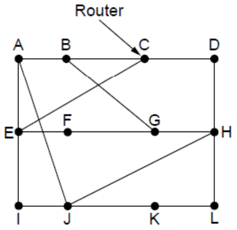
Distance Vector Routing

- AKA distributed Bellman-Ford routing algorithm
- Used initially in ARPANET and the Internet (RIP)
- Each router maintains a routing table indexed by and containing one entry for EACH router in the network
  - Entry (Y, Z) at row X – where the estimate of the distance to node X is Y through the output port Z

Distance Vector Routing – cont'd

cost = delay estimate

- The update process
- J's cost estimates to its neighbors A, I, H, and K are 8, 10, 12, and K, respectively.
- J receives cost vectors from its neighbors
- J updates its cost to G (for example): min (cost_AG + cost_JA, cost_IG + cost_JI, cost_HG + cost_JH, cost_KG + cost_JK) = min(18+8, 31+10, 6+12, 31+6) = 18 – through node H
- Same calculation is performed for all other destinations.



(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	

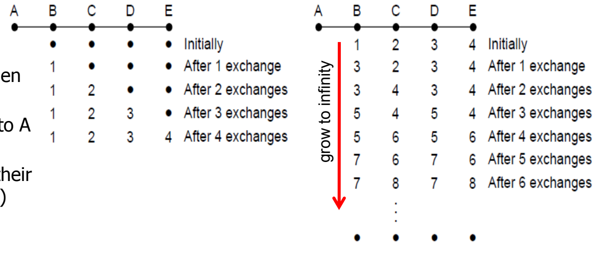
Vectors received from J's four neighbors

(b)

(a) A network.
(b) Input from *A, I, H, K, and the new routing table* for *J.*

Distance Vector Routing – Count-to-infinity Problem

- Distance vector routing algorithm may converge slowly
- Reacts rapidly to "good" news but very slowly to "bad" news
- Consider the five-node (linear) network shown
  - Delay metric is number of hops
- Good news travels fast
  - Initially, A is down – All other routers (B, C, D, and E) know this; i.e. delay to A is infinity
  - Assume A comes up
  - Step 1 - B updates its route to A to be 1
  - Step 2 – C updates its route to A to be 1 + 1 = 2
  - Etc.
- Bad news travels slow
  - Initially all links are up, then A goes down
  - Step 1 – B updates entry to A (3, C)
  - Step 2 – B and C update their entries to (5, C) and (4, B)
  - Other nodes start also updating their entries
  - Etc



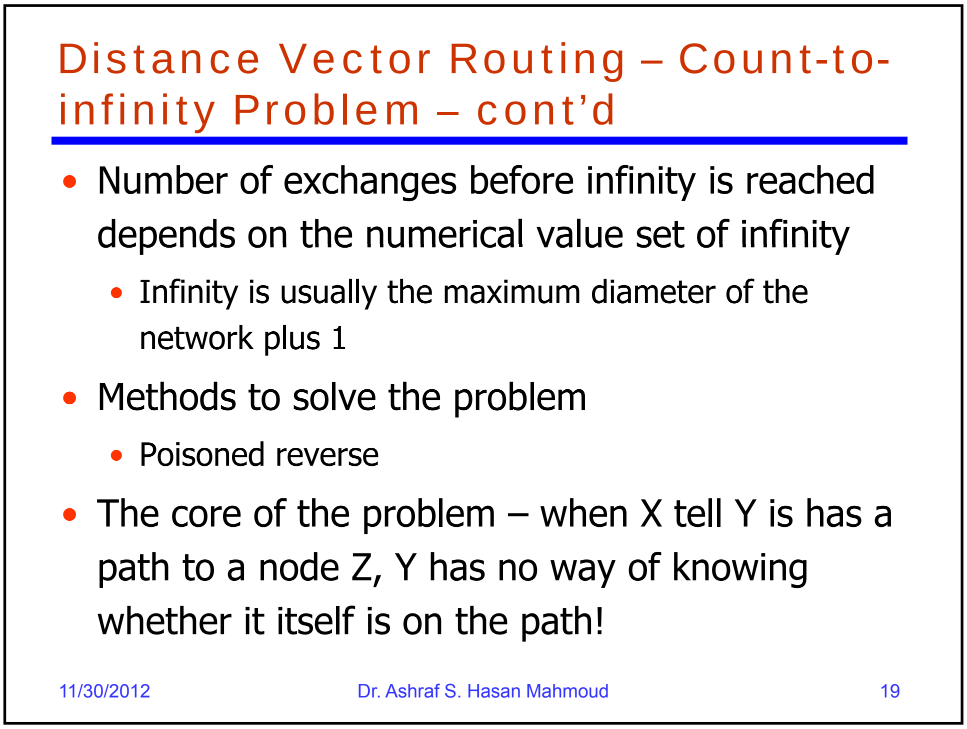
A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(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
		⋮			
	•	•	•	•	

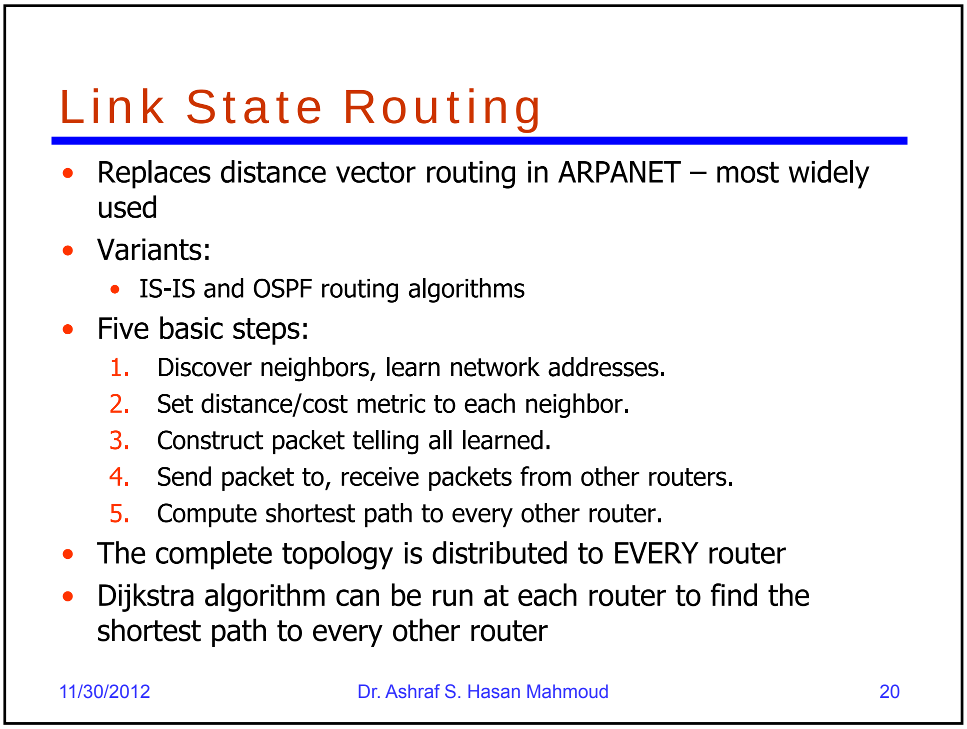
(b)

The count-to-infinity problem



Distance Vector Routing – Count-to-infinity Problem – cont'd

- Number of exchanges before infinity is reached depends on the numerical value set of infinity
  - Infinity is usually the maximum diameter of the network plus 1
- Methods to solve the problem
  - Poisoned reverse
- The core of the problem – when X tell Y is has a path to a node Z, Y has no way of knowing whether it itself is on the path!

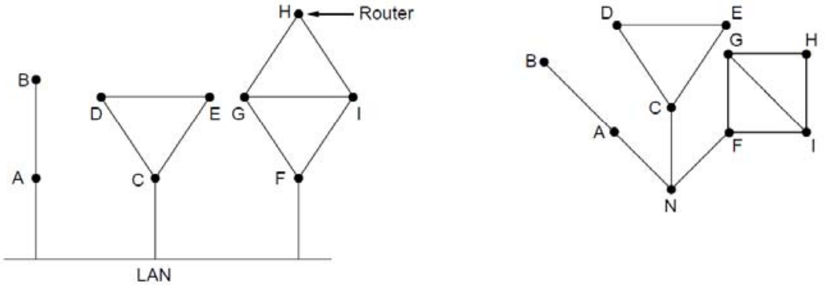


Link State Routing

- Replaces distance vector routing in ARPANET – most widely used
- Variants:
  - IS-IS and OSPF routing algorithms
- Five basic steps:
  1. Discover neighbors, learn network addresses.
  2. Set distance/cost metric to each neighbor.
  3. Construct packet telling all learned.
  4. Send packet to, receive packets from other routers.
  5. Compute shortest path to every other router.
- The complete topology is distributed to EVERY router
- Dijkstra algorithm can be run at each router to find the shortest path to every other router

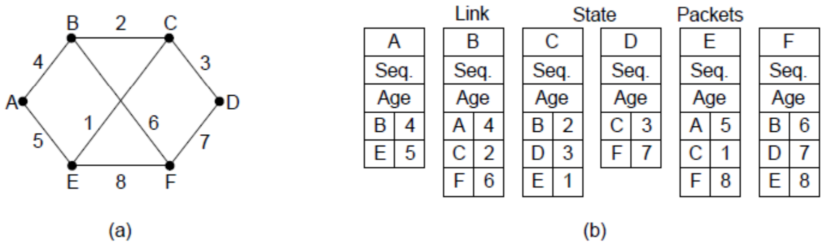
Link State Routing – Learning About Neighbors

- Routers discover/learn their neighbors
  - HELLO packets – receiving router responds with its unique name
  - Simple for point-to-point connections
- For broadcast links (e.g. switch, ring, or classic Ethernet)
  - One designated router plays the role of the consolidated LAN point



Link State Routing – Building Link State Packets

- Packet containing
  - Source address (sender)
  - Sequence number and age
  - List of neighbors and the corresponding cost
- When to build link state packets?
  - Periodically
  - Event wise (e.g. when line or neighbor going up or down)



A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

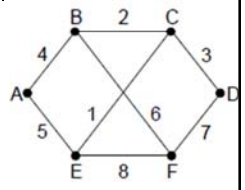
(a) A network. (b) The link state packets for this network.

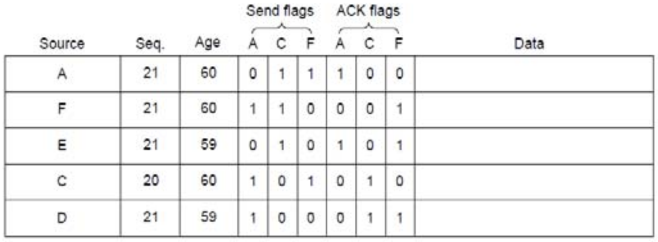
Link State Routing – Distributing Link State Packets

- All of the routers MUST get ALL of the link state packets quickly and reliably
  - If different routers use different versions of the network topology ➔ inconsistent routes (loops, unreachable machines, etc.)
- The basic distribution algorithm is flooding
  - Each packet contains a sequence number that is increased for each new packet sent
  - Packets arriving for the first time are forwarded and packets already seen are discarded
- Potential problems and fixes:
  - Sequence wrap around – use 32-bit sequence number
  - Router crashing and having to start again – sequence number 0 → packet rejected as old
  - Corrupted sequence number → packets to be sent will be rejected because of a corrupted old sequence number
- Solutions to problems 2 and 3 is to include the *age field* – decremented once per second and also by each router during the initial flooding process
  - If zero, info from router is discarded
  - No packet can get lost and live for ever

Link State Routing – Distributing Link State Packets – cont'd

- Adding more robustness
  - Arriving flooded link state packets are not transmitted immediately but held for some time
  - Processing of arriving link state packets
  - All link state packets are ACKed
- Data structure for packet buffer at node B





Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags 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	

immediate neighbor

immediate neighbor

Arrived through EAB and EFB ➔
Must be sent to C - A and F must be ACKed

immediate neighbor

Arrived through DCB and DFB ➔
Must be sent to A - C and F must be ACKed

The packet buffer for router *B* in in shown network

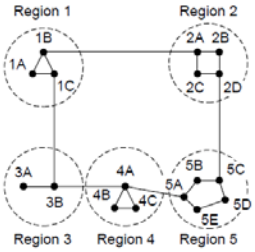
Link State Routing – Computing New Routes

- Every link is represented twice – one per direction
  - One can find a path from A → B different than the path from B → A
- Dijkstra's algorithm can run locally on each router to compute shortest path to all other nodes
- For n routers where each has k neighbors ➔ memory space required is proportional to kn
- Computation time grows fast with kn
- IS-IS – Intermediate System-Intermediate System link state protocol
  - Designed for DECnet, later adopted by ISO for the OSI model
  - Modified later to be used the IP protocol
  - Can carry information about multiple network layer protocols
- OSPF – Open Shortest Path First Protocol
  - Designed by the IETF after the IS-IS and adopts many of its innovations

Hierarchical Routing

- If all routers belong to one region – then every router must know the path to every other router
  - Very resource consuming for large networks
- For Hierarchical routing – the router knows the internal topology of its region; knows how to route to a particular region
  - Does not know the topology of the other regions

- Routing table for 1A:
  - Flat structure case – 17 entries
  - Hierarchical case – 7 entries
- Note that path from 1A to 5C goes through 1C to regions 3 and 4 and then 5 – NOT OPTIMUM – Tradeoff!
- For N routers network – the optimal number of levels is log(N) requiring e log(N) entries per table
  - Effective increase in mean path length is 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

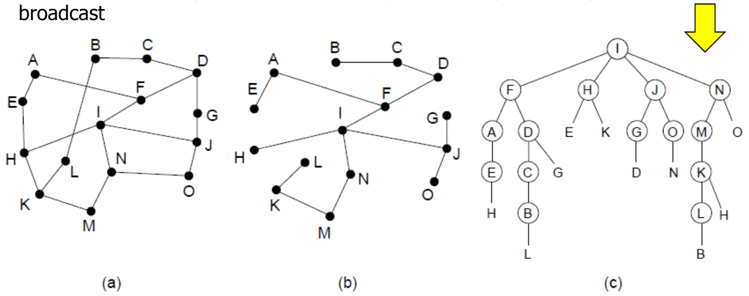
(c)

Broadcast Routing

- Sending a packet to all destinations simultaneously
- Simple/inefficient – the source sends distinct packets for each node in the network
- Solutions:
  - Multidestination routing – each packet contains a list of a bit map indicating the desired destinations
    - One packet per output port of a router
    - Must know the destinations
  - Flooding – can be made efficient with sequence number per source; simple
  - Reverse path forwarding
  - Spanning Tree broadcast algorithm

Broadcast Routing – Reverse Path Forwarding

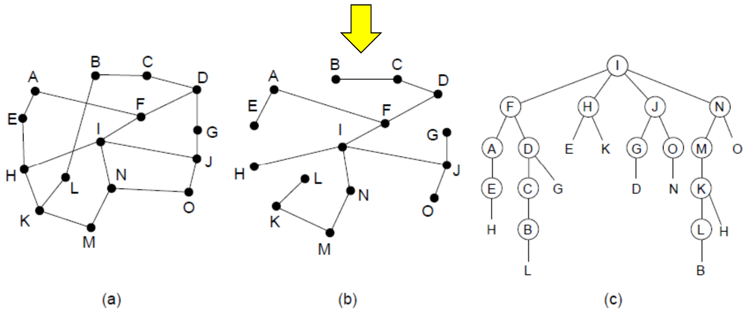
- Simple rules
  - When broadcast arrives at router
  - Router checks if packets arrived on a link that is normally used for sending packets towards the source
    - Yes – there is an excellent chance that the broadcast packet followed the shortest path from the source (i.e. first copy to arrive); copy to all outgoing links except the one it arrived on
    - No – likely to be a duplicate – discard
- Example of reverse path forwarding – requires five hops and 24 packets to terminate the broadcast



Reverse path forwarding. (a) A network. (b) A sink tree.
(c) The tree built by reverse path forwarding.

Broadcast Routing – Spanning Tree

- Spanning Tree (e.g. the sink tree) – achieves broadcast in four hops and 14 packets!



Reverse path forwarding. (a) A network. (b) A sink tree.
(c) The tree built by reverse path forwarding.

Multicast Routing

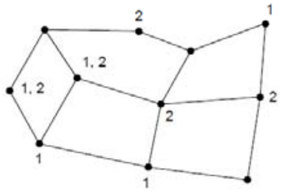
- Application: multiplayer gaming, live video streaming, etc.
  - Sending distinct packets to each of the receivers is inefficient
- Multicasting – a mechanism to send packet to a well-defined group
- User are grouped in groups – each group has a designated multicast address – routers know the groups to which they belong
- Builds on broadcast routing – employ spanning tree for multicast
  - Case I: dense group distributed all over the network
  - Case II: sparse and very few nodes compared to the rest
- Refer to textbook slides on the subject

copied from authors’ slides

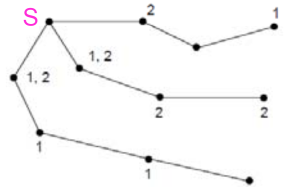
Multicast Routing (1) – Dense Case

Multicast sends to a subset of the nodes called a group

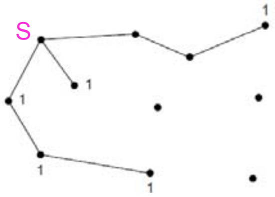
- Uses a different tree for each group and source



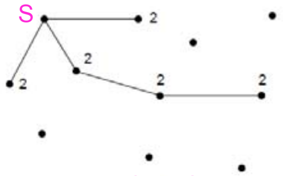
Network with groups 1 & 2



Spanning tree from source S



Multicast tree from S to group 1



Multicast tree from S to group 2

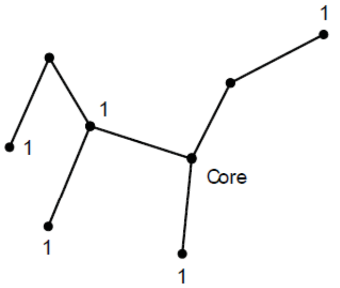
CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011

copied from authors’ slides

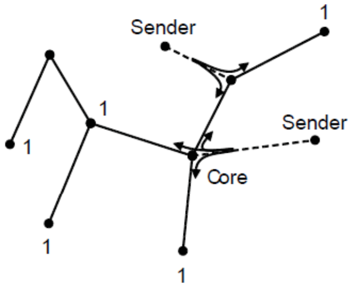
Multicast Routing (2) – Sparse Case

CBT (Core-Based Tree) uses a single tree to multicast

- Tree is the sink tree from core node to group members
- Multicast heads to the core until it reaches the CBT



Sink tree from core to group 1



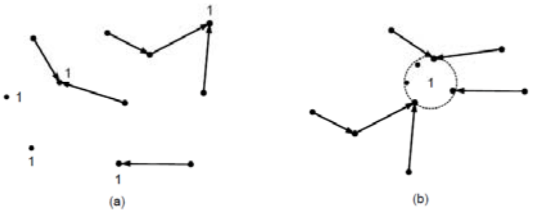
Multicast is send to the core then
down when it reaches the sink tree

CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011

Anycast Routing

- Packet delivered to nearest member of a group
- Anycast routing algorithm is an algorithm that finds all paths to the nearest member of all groups
- Internet DNS service – more in Chapter 7
- Regular distance vector routing algorithms and link state routing algorithms can produce anycast routes

- Case I: Distance vector routing (DVR)
  - All members given same address – say "1"
  - DVR will distribute vectors as usual and nodes will choose the shortest path to "1"
  - ➔ Nodes will be sending to the nearest instance of destination "1"
- Case II: Link state routing (LSR)
  - ?



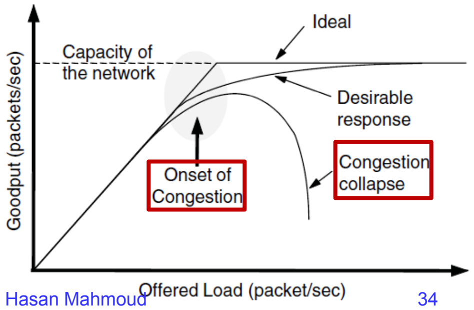
(a) Anycast routes to group 1.

(b) Topology seen by the routing protocol.

Congestion Control Algorithms

- Congestion – too many packets in the network → delay and loss → performance degradation
- Responsibility of network and transport layer
- Goodput versus load curve: Goodput – rate of useful packets delivered by the network
  - Onset of congestion
  - congestion collapse – goodput degrades rapidly with the increase of load (beyond capacity)

- Signs of congestions
  - Excessive delay
  - Lost packets
  - ➔ retransmissions ➔ more congestion
- The goal: avoid congestion and avoid congestion collapse

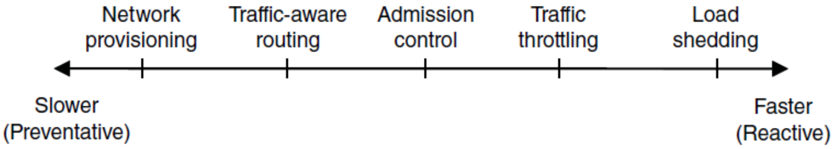


Congestion Control Algorithms – cont'd

- Congestion cannot be avoided completely – case of a stream of packets all destined for the same router output port
  - Queue build up
  - Packets dropped (finite buffer size) → would infinite memory solve the issue?
- Steps:
  1. Direct traffic to other parts of the network away from congestion
  2. Eventually all regions become congested ➔ shed some load
  3. Build faster network
- Difference between congestion control and flow control
  - Congestion control – network is able to carry offered load
  - Flow control – sender does not overwhelm receiver; 2 end points perspective

Congestion Control Algorithms – Approaches

- The two key ideas: increase resources and decrease load
- Applied on different time scales (Refer to Figure below)



Congestion Control Algorithms – Approaches – cont'd

- Provisioning
  - Scale of months
  - Based on long-term traffic trends
- Traffic-aware Routing
  - Shift traffic from heavily used paths
  - E.g. splitting traffic (load balancing)
- Admission Control
  - Intent to reduce load
  - E.g. virtual circuits – reject calls
- Traffic Throttling
  - Network assisted congestion control
  - Feedback from network
  - Sensitive to timing → oscillations
- Load Shedding
  - Last resort
  - Can help in preventing congestion collapse

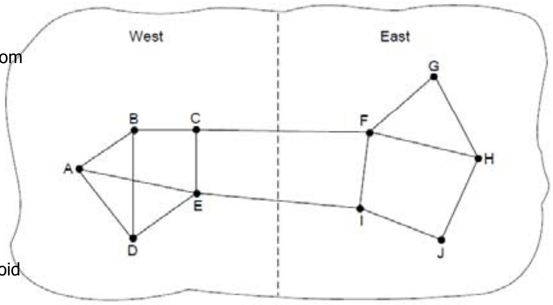
Congestion Control Algorithms – Approaches – cont'd

- Two main difficulties:
  1. How to identify the onset of congestion?
  2. How to inform the source that needs to slow down?

- **Answer 1:** routers can monitor average load, queueing delay, or packet loss → increasing numbers indicate a problem!
- **Answer 2:** Routers may need to participate in a feedback loop with the source.

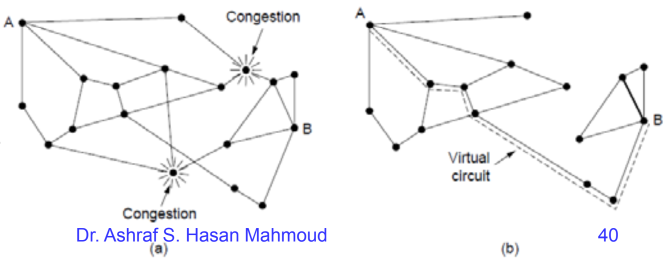
Congestion Control Algorithms – Traffic-Aware Routing

- Objective: shift traffic away from hotspots (potential congestion spots)
- Setting weights: link bandwidth, propagation delay, measured load (e.g. queueing delay)
- Oscillations!
- Solutions:
  - Multipath routing
  - Shift traffic across routs slowly to reach a stable solution
- Oscillations Example:
  - Assume initially that best route from West to East is through link CF
  - Link CF become congested
  - Routing algorithm selects EI link
  - Link EI congested (link CF lightly loaded)
  - Routing algorithm selects link CF
  - Etc.
- Traffic engineering – changing (slowly) the input to the routing algorithm to avoid congestion



Congestion Control Algorithms – Admission Control

- Widely used in virtual-circuit networks
- How to estimate the required bandwidth for bursty traffic
  - Average, Peak, etc?
  - Leaky bucket – two parameters that bound the average rate and the instantaneous burst size of traffic – to be revisited.
- Two approaches:
  - Reserve capacity along the virtual-circuit path based on traffic description – Service agreement; related to QoS
  - How many virtual-circuits to carry with given traffic descriptions?
- E.g. Peak rate for VC is 10 Mb/s – how many can pass through 100 Mb/s link?
  - 10 can be admitted without concern of congestion – wasteful of link capacity
- Admission control may be combined with traffic-aware routing
- Select routes "around" hotspots as part of the VC set up procedure



Congestion Control Algorithms – Traffic Throttling

- Feedback to transmitter to reduce its transmit rate
- Congestion avoidance
- Two problems to solve:
  1. Router must determine when congestion is approaching
  2. Routers must deliver timely feedback to the sender
- <u>Solutions to Problem 1</u>:
  - Routers monitor (a) utilization of output links, (b) the buffering of queued packets inside the router, and (c) number of packets lost due to insufficient buffering
- (a) Average utilization is not a good measure of burstiness
- (c) Packet loss in router occurs *after* congestion onset – too late
- (b) queueing delay for packet – captures congestion experience
  - Should be low most of the time
  - How to estimate this queueing delay, *d*?
- Exponentially Weighted Moving Average (EWMA):

$$d\_\text{new} = \alpha\, d\_\text{old} + (1 - \alpha)\, s$$

where *s* is the instantaneous queue length (sampled periodically)

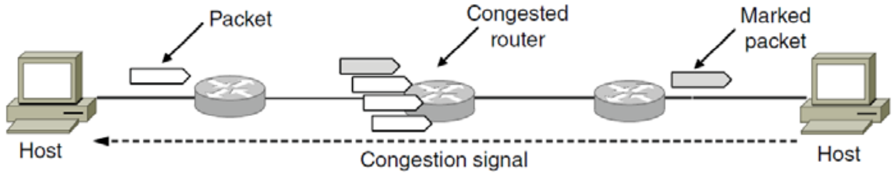
- The parameter α – determines how fast the router forgets recent history

Congestion Control Algorithms – Traffic Throttling – cont’d

- <u>Solutions to Problem 2</u>:
  - Routers must identify the appropriate senders
  - Deliver timely feedback to the appropriate senders by sending as few as possible of packet – already congestion network
- (a) Choke Packets:
  - Routers identifies packets in its buffer
  - Sends choke packet to source specifying the destination address found in the packets
  - Source reduces traffic upon receiving choke packet – may receive multiple choke packets
    - Traffic reduction requires time to take effect
- For IP-datagram – router may select packets at random ➔ causes choke packets to be sent to fast senders
- Example procedure for IP – SOURCE QUENCH message – is not clearly defined – not implemented

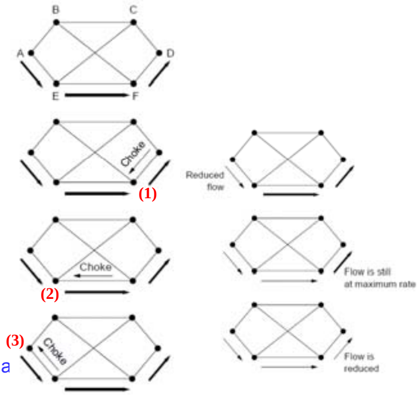
Congestion Control Algorithms – Traffic Throttling – cont'd

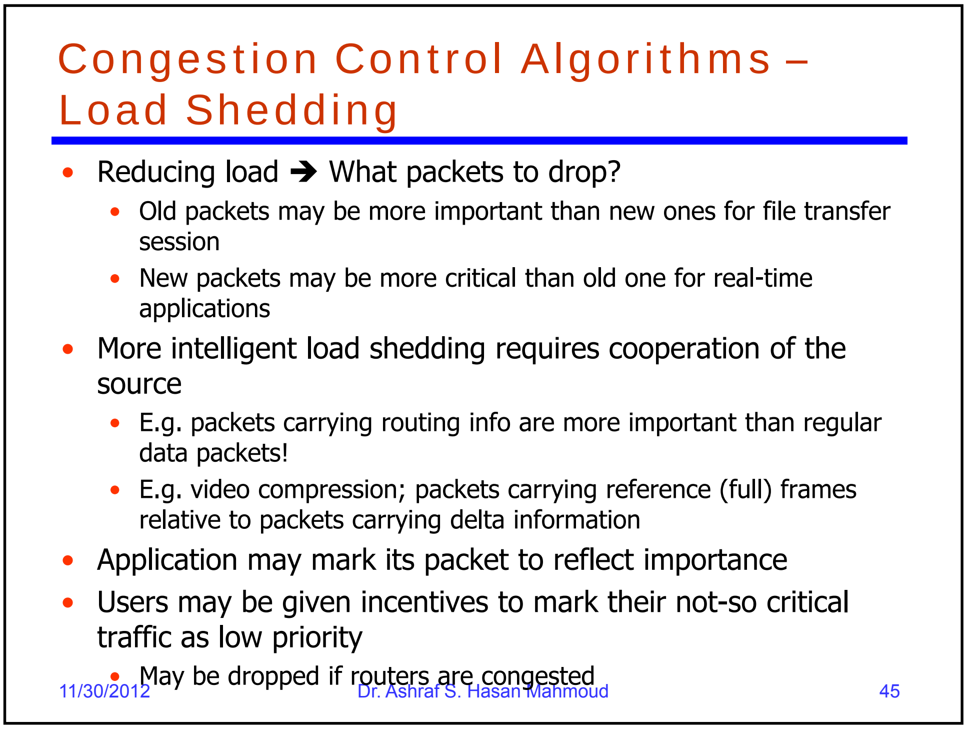
- (b) Explicit Congestion Notification (ECN):
  - Used in the Internet
  - 2 bits in the IP packet header – indicate whether packet has experienced congestion
  - Packets are marked by routers
  - Destination will echo any marks back to source



Congestion Control Algorithms – Traffic Throttling – cont'd

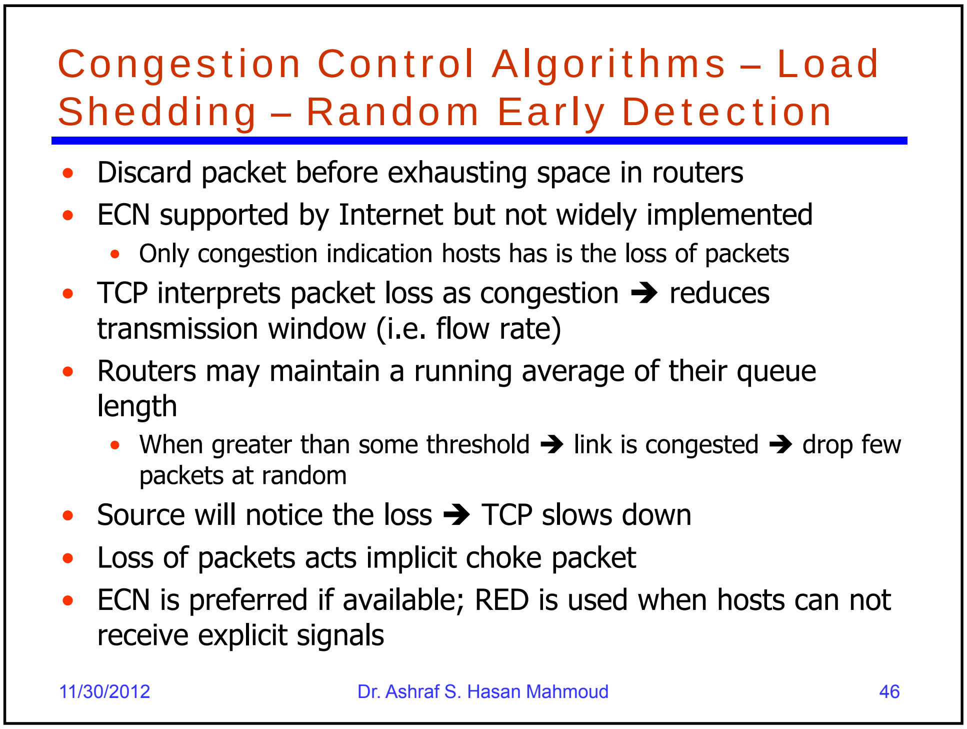
- (b) Hop-by-hop backpressure:
  - Appropriate for high-speed or over long distance
    - Many new packets may be transmitted after congestion has signaled
  - Solution: Let the choke packet **take effect at every hop**
- Example:
  1. Choke packet received by F ➔ F reduces flow to D; F devotes more buffers for session
  2. Choke packet received by E ➔ …
  3. Choke packet received by A (source) ➔ traffic flow reduced
- **Key idea:** provide quick relieve at the price of buffering upstream





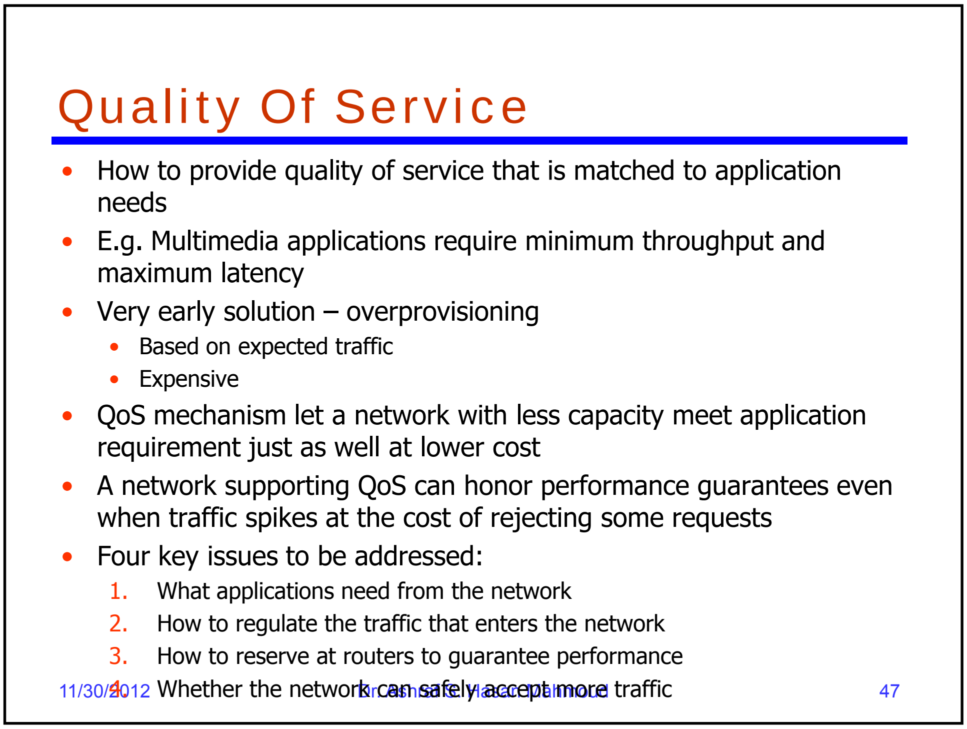
Congestion Control Algorithms – Load Shedding

- Reducing load ➔ What packets to drop?
  - Old packets may be more important than new ones for file transfer session
  - New packets may be more critical than old one for real-time applications
- More intelligent load shedding requires cooperation of the source
  - E.g. packets carrying routing info are more important than regular data packets!
  - E.g. video compression; packets carrying reference (full) frames relative to packets carrying delta information
- Application may mark its packet to reflect importance
- Users may be given incentives to mark their not-so critical traffic as low priority
  - May be dropped if routers are congested



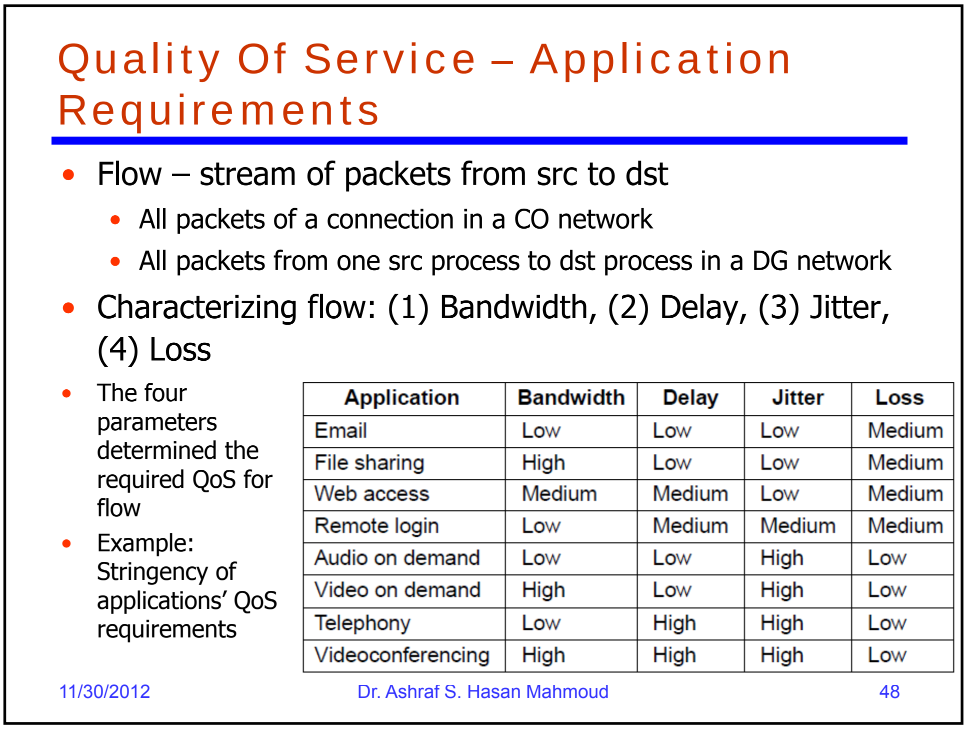
Congestion Control Algorithms – Load Shedding – Random Early Detection

- Discard packet before exhausting space in routers
- ECN supported by Internet but not widely implemented
  - Only congestion indication hosts has is the loss of packets
- TCP interprets packet loss as congestion ➔ reduces transmission window (i.e. flow rate)
- Routers may maintain a running average of their queue length
  - When greater than some threshold ➔ link is congested ➔ drop few packets at random
- Source will notice the loss ➔ TCP slows down
- Loss of packets acts implicit choke packet
- ECN is preferred if available; RED is used when hosts can not receive explicit signals



Quality Of Service

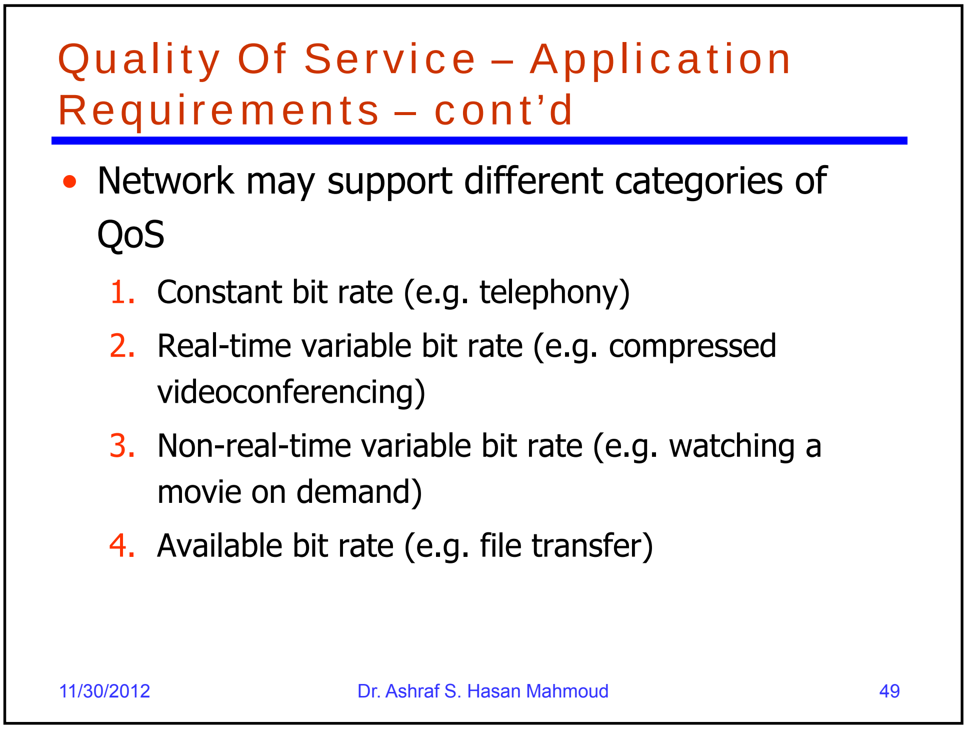
- How to provide quality of service that is matched to application needs
- E.g. Multimedia applications require minimum throughput and maximum latency
- Very early solution – overprovisioning
  - Based on expected traffic
  - Expensive
- QoS mechanism let a network with less capacity meet application requirement just as well at lower cost
- A network supporting QoS can honor performance guarantees even when traffic spikes at the cost of rejecting some requests
- Four key issues to be addressed:
  1. What applications need from the network
  2. How to regulate the traffic that enters the network
  3. How to reserve at routers to guarantee performance
  4. Whether the network can safely accept more traffic



Quality Of Service – Application Requirements

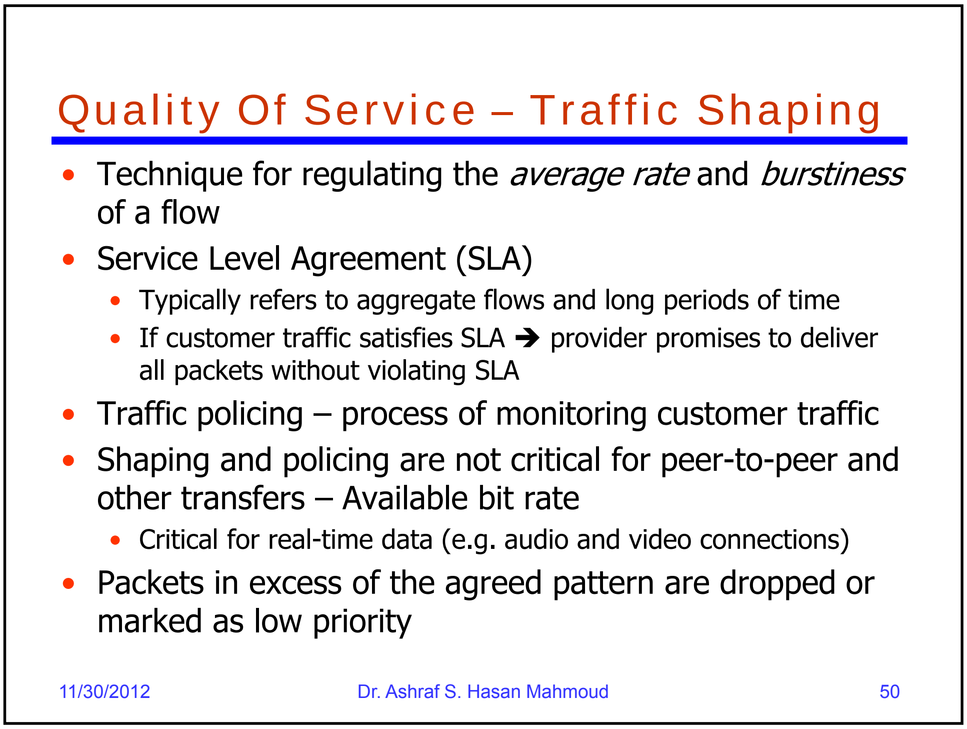
- Flow – stream of packets from src to dst
  - All packets of a connection in a CO network
  - All packets from one src process to dst process in a DG network
- Characterizing flow: (1) Bandwidth, (2) Delay, (3) Jitter, (4) Loss
- The four parameters determined the required QoS for flow
- Example: Stringency of applications' QoS requirements

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



Quality Of Service – Application Requirements – cont'd

- Network may support different categories of QoS
  1. Constant bit rate (e.g. telephony)
  2. Real-time variable bit rate (e.g. compressed videoconferencing)
  3. Non-real-time variable bit rate (e.g. watching a movie on demand)
  4. Available bit rate (e.g. file transfer)



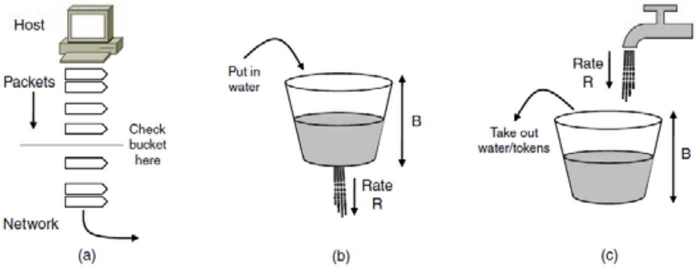
Quality Of Service – Traffic Shaping

- Technique for regulating the *average rate* and *burstiness* of a flow
- Service Level Agreement (SLA)
  - Typically refers to aggregate flows and long periods of time
  - If customer traffic satisfies SLA ➔ provider promises to deliver all packets without violating SLA
- Traffic policing – process of monitoring customer traffic
- Shaping and policing are not critical for peer-to-peer and other transfers – Available bit rate
  - Critical for real-time data (e.g. audio and video connections)
- Packets in excess of the agreed pattern are dropped or marked as low priority

Quality Of Service – Leaky and Token Bucket

- Used to shape or police packets entering the network
- Bucket of size B and outflow is R regardless of input
- Host is equipped with network interface that is responsible for shaping the traffic
- Leaky bucket algorithm:
  - Packet arriving to the less than full bucket enters the bucket and then the network
  - Packet arriving to a full bucket is either queued (at the host side of the interface) or discarded (at the network side of the interface)
- Token bucket algorithm
  - Tokens accumulate at fixed rate R in the bucket
  - Bucket maximum capacity of B tokens
  - To send traffic we must use tokens from the bucket

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



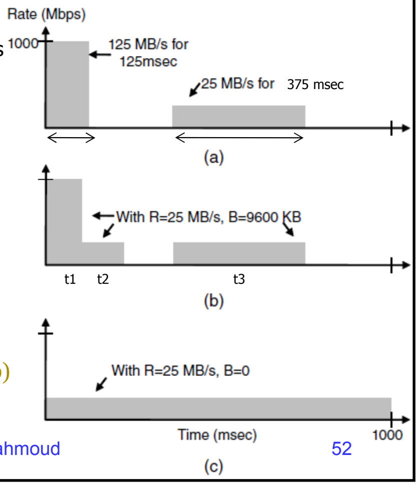
Quality Of Service – Leaky and Token Bucket – cont'd

- Mechanisms to limit the long-term rate of a flow but allow short-term bursts up to a maximum regulated length to pass through unaltered
  - Large burst will be smoothed by a leaky bucket traffic shaper
- Example:
  - A computer can produce data up to 1000 Mb/s
  - Link interface also runs at 1000 Mb/s

a) Total # of bytes: 125MB x 0.125 + 25MB x 0.375
= 15.625 MB + 9.375 MB = 25 MB

c) Total # of Bytes: 25MB x 1.0 = 25 MB

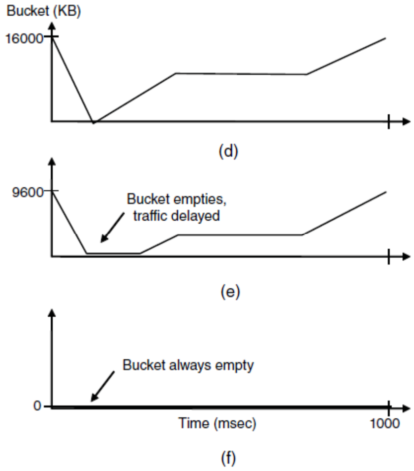
b) R = 25MB/s and B = 9600 KB ➔ S = B/(M-R) = 93.75 msec
Note that t1 = S = 93.75 msec
t2 = (15.625 – 125 x 0.09375)/25 = 0.15625 sec (156.625 msec)
t3 = 375 msec



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

Quality Of Service – Leaky and Token Bucket – cont'd

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



Quality Of Service –Token Bucket – Implementation

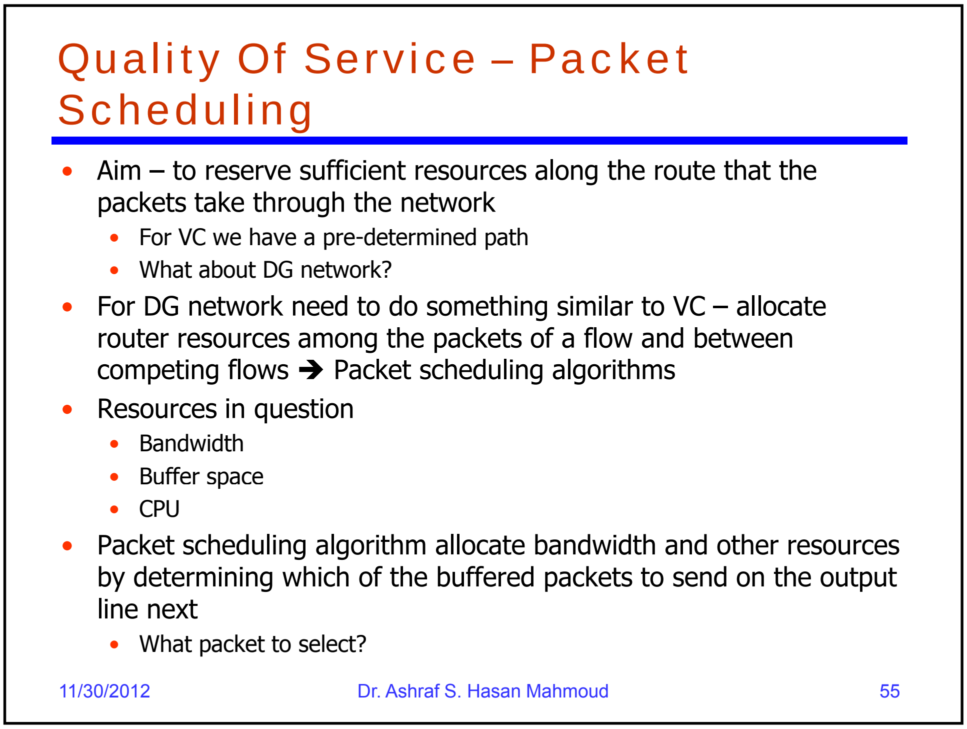
- Bucket level – counter
- Clock tick every Δt ➔ counter decremented by R/ Δt units
- Every time a unit of traffic is sent into the network the counter is decremented
- Traffic is sent till the counter reaches zero
- Unit of transmission (or bucket level)
  - Packet – what about variable packet sizes
  - Bytes
- Length of maximum burst (i.e. until the bucket empties)

$$B + R S = M S$$

where B is the bucket size, R token arrival rate in bytes/sec, M is the maximum output rate in bytes/sec, and S is the burst length in seconds

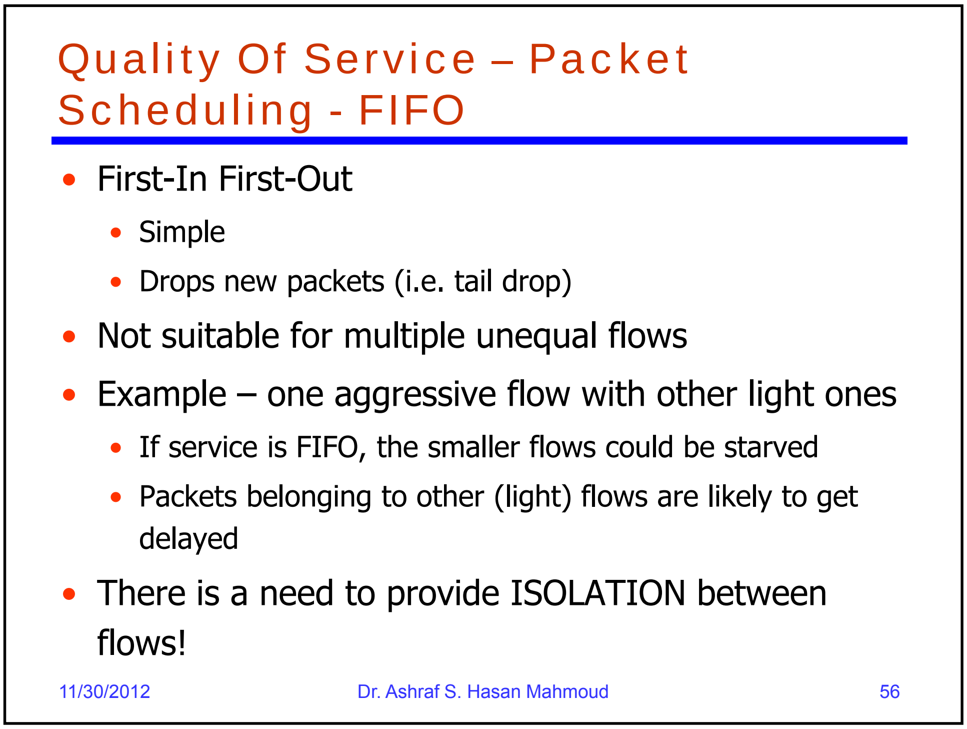
➔ Length of maximum burst S = B / (M – R);

- Token buckets implemented for shaping hosts ➔ Packets queued and delayed until the buckets permit them to be sent
- Token bucket implemented for policing at routers ➔ no more packets are sent than permitted



Quality Of Service – Packet Scheduling

- Aim – to reserve sufficient resources along the route that the packets take through the network
  - For VC we have a pre-determined path
  - What about DG network?
- For DG network need to do something similar to VC – allocate router resources among the packets of a flow and between competing flows ➔ Packet scheduling algorithms
- Resources in question
  - Bandwidth
  - Buffer space
  - CPU
- Packet scheduling algorithm allocate bandwidth and other resources by determining which of the buffered packets to send on the output line next
  - What packet to select?

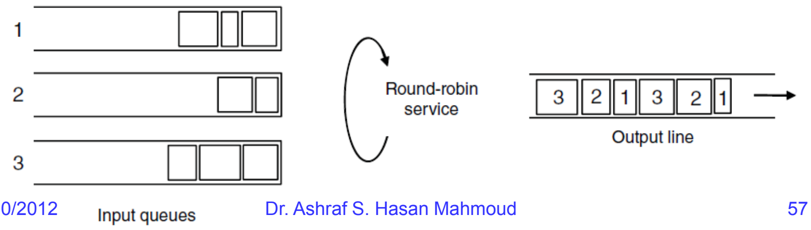


Quality Of Service – Packet Scheduling - FIFO

- First-In First-Out
  - Simple
  - Drops new packets (i.e. tail drop)
- Not suitable for multiple unequal flows
- Example – one aggressive flow with other light ones
  - If service is FIFO, the smaller flows could be starved
  - Packets belonging to other (light) flows are likely to get delayed
- There is a need to provide ISOLATION between flows!

Quality Of Service – Packet Scheduling – Fair Queueing

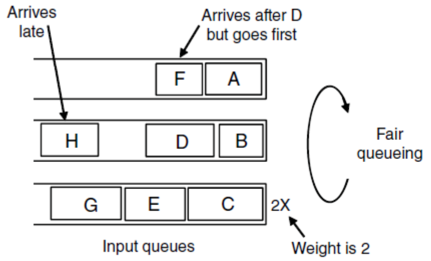
- Strong isolation of flows – no interference
- Round-robin fair queueing
- All host get send at the same rate, if they have traffic (and all packets are equal in length!)
- Byte-by-byte round-robin
  - Compute virtual time – that is the number of the round at which each packet would finish being sent
  - Each round drains a byte from all of the queues that have data to send
  - Packets are sorted in the order of their finishing time and sent in that order
- Fair queueing (as described above) does not pre-empt packets



Quality Of Service – Packet Scheduling – Weighted Fair Queueing

- Example 1: consider the first two input queues only
- Solution – Packets arrive in the order of A, B, D, F and H ➔ transmit order (using data from table) is A. B. F. and D
- Example 2: consider the three input queues – case of Weighted Fair Queueing (WFQ); third queue carries twice the weight
- Solution: In general, $F_i = \max(A_i, F_{i-1}) + L_i/W$
  where Fi finish time of ith packet, Ai arrival instant of ith packet, Li length of ith packet in bytes, and W is the relative weight (bigger is higher priority)

(a) Weighted Fair Queueing



(a)

(b) Finishing times for the packets

Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

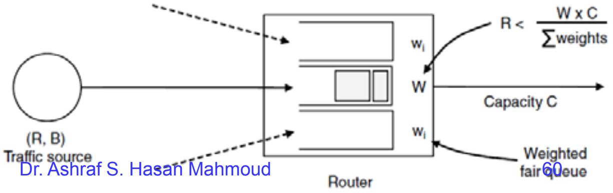
(b)

Quality Of Service – Admission Control

- Admission Control:
  - User offers a flow with specific QoS requirements to the network
  - Network decides whether to accept or reject flow
- Accept flow – reserve needed resources at routers to meet specified QoS
  - A single congested router can break the QoS guarantee
- QoS routing – choosing a different route (than best route) that has excess capacity
  - May involve also splitting the flow into multiple paths
- Key issues:
  - How would application estimate resources needed in advance
  - Different applications vary in their tolerance for delays/losses
  - Some application are willing to negotiate their flow parameters (e.g. 30 frame/sec versus 25 frame/sec video and number of bits per pixel)
- Flow specification:
  - Token bucket rate (bytes/sec) --¬
  - Token bucket size (bytes) → specify the maximum burst possible and sustained avg rate
  - Peak data rate (Bytes/sec) → sender can not exceed this rate even for short times
  - Minimum packet size (Bytes) → relevant to CPU processing time per packet
  - Maximum packet size (Bytes) → to meet network limitations on maximum packet size

Quality Of Service – Admission Control – cont'd

- How to relate flow specifications to set of resource reservations?
- Example: C = 1 Gb/s, flow has rate R of 1 Mb/s
- Solution: to admit flow, the weight of the flow must be greater than 1/1000 of the total weights
  - Guarantees minimum bandwidth
- Largest queueing delay depends on the burst size of the token bucket
- Two extremes:
  - (1) smooth traffic – no bursts ➔ no queueing; Dmin = 0;
  - (2) traffic is saved up in burst of B Bytes each, i.e. maximum burst size ➔ B Bytes arrive at once to the router; D is the time to drain B Bytes at the guaranteed rate R. ➔ Dmax = B/R
- If Dmax is not acceptable, the flow must request more bandwidth from network



Quality Of Service – Admission Control – Summary

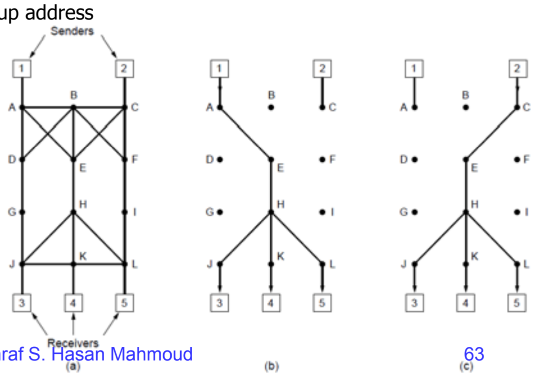
- Prescribed previous guarantees are HARD
- Two key points:
  - Token bucket bound the burstiness of src
  - FQ isolates the bandwidth given to different flows
- ➔ flow of interest WILL MEET its bandwidth and delay guarantees (in a router) regardless of how other competing flows behave
  - Even if they all burst at the same time
- The above holds even for a path through multiple routers in any network topology
  - Flow get a minimum bandwidth because that bandwidth is guaranteed at each router
  - Flow get maximum delay (i.e. Dmax) – if flow burts and traffic hit first router (worst case) – D = Dmax – but this smoothens the traffic for subsequent routers ➔ burst will not incur further delays ➔ maximum D for any router is Dmax

Quality Of Service – Integrated Services (IntServ)

- Integrated Services – IETF effort for devising an architecture for streaming multimedia
- Aimed at unicast and multicast applications
- Consider multicast scenarios (dynamic group membership)
  - Advance reservation of resources does not work well - does not scale

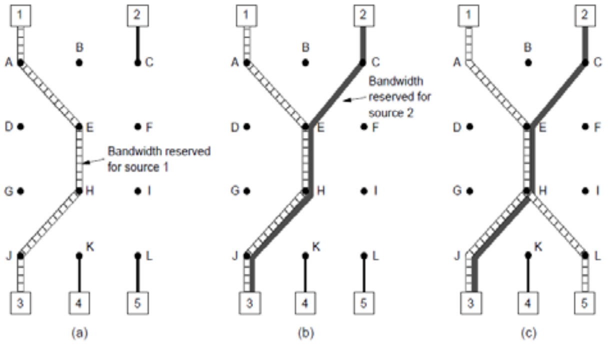
Quality Of Service – Integrated Services - RSVP

- RSVP – The Resource Reservation Protocol
  - Used to make reservations
  - Multiple senders can transmit to multiple groups of receivers
  - Individual receivers may switch channels freely
  - Optimizes bandwidth and eliminates congestion
- Utilizes multicast routing – based on spanning trees – not part of RSVP
  - Each group is designated a group address
- Example: (a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.
- Host 1 and Host 2 are multicast senders
- Hosts 3, 4, and 5 are multicast receivers
- Senders and receivers may be not disjoint!



Quality Of Service – Integrated Services – RSVP – cont'd

- Any receiver in the group can make the reservation message – travels up the tree to the sender
  - Propagated using reverse path forwarding
  - At each hop, the router reserves the needed resources
- Example:
  - (a) Host 3 requests a channel to host 1.
- (b) Host 3 then requests a second channel, to host 2.
  - Note <u>two separate channels</u> are needed from host 3 to router E
- (c) Host 5 requests a channel to host 1
  - Reservations are made as far as router H
  - The H checks – Host 3 and Host 4 may have asked for different amounts of bandwidth!

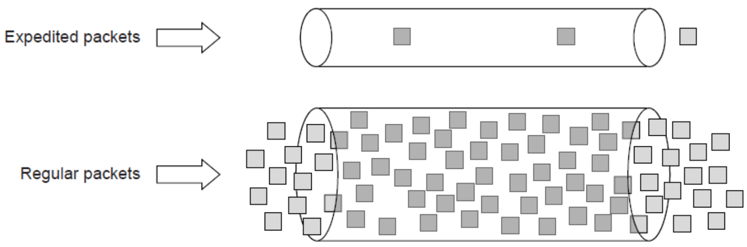

1
2
A
B
C
D
E
F
G
H
I
J
K
L
3
4
5
Bandwidth reserved for source 1
Bandwidth reserved for source 2
(a)
(b)
(c)

Quality Of Service – Differentiated Services (DiffServ)

- Flow-based algorithms (such as IntServ)
  - Require in advance setup for every flow
  - Network routers need to keep per-flow state
  - Does not scale well for thousands and millions of flows
  - Require significant change to the router code; also involve complex router-to-router exchange
- DiffServ – simpler approach – can be implemented in routers without the need for path setup
  - Class-based QoS
- May be offered by a set of router forming an **administrative domain** (e.g. an ISP or telco)
  - **Set of service classes are defined with corresponding forwarding rules**
  - Customer packets entering the domain are marked to indicate the class
  - Customer packets receive service depending on the class
- Classes define "per hop behavior"
  - Not a guarantee across the network!
- Traffic within a class may be shaped (i.e. using token bucket method)

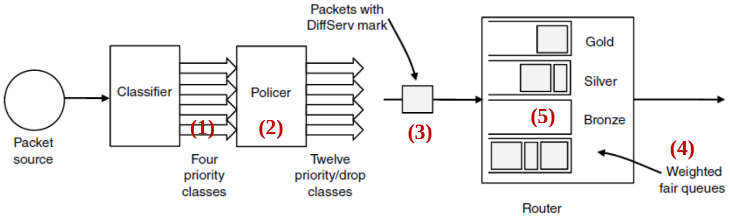
Quality Of Service – Differentiated Services (DiffServ) – Expedited Forwarding

- Choice of service classes is up to the operator
- There are network-independent service classes (defined by IETF)
  - Expedited forwarding
- Implementation
  - Packets are classified (expedited versus regular) at sending host (more info is available) or ingress (first) router
    - E.g. VOIP packets are marked expedited
    - If network does not support expedited service – no harm done
  - Routers have two queues (one per class) with priority given to the expedited traffic queue
- Ingress router may police the traffic



Quality Of Service – Differentiated Services (DiffServ) – Assured Forwarding

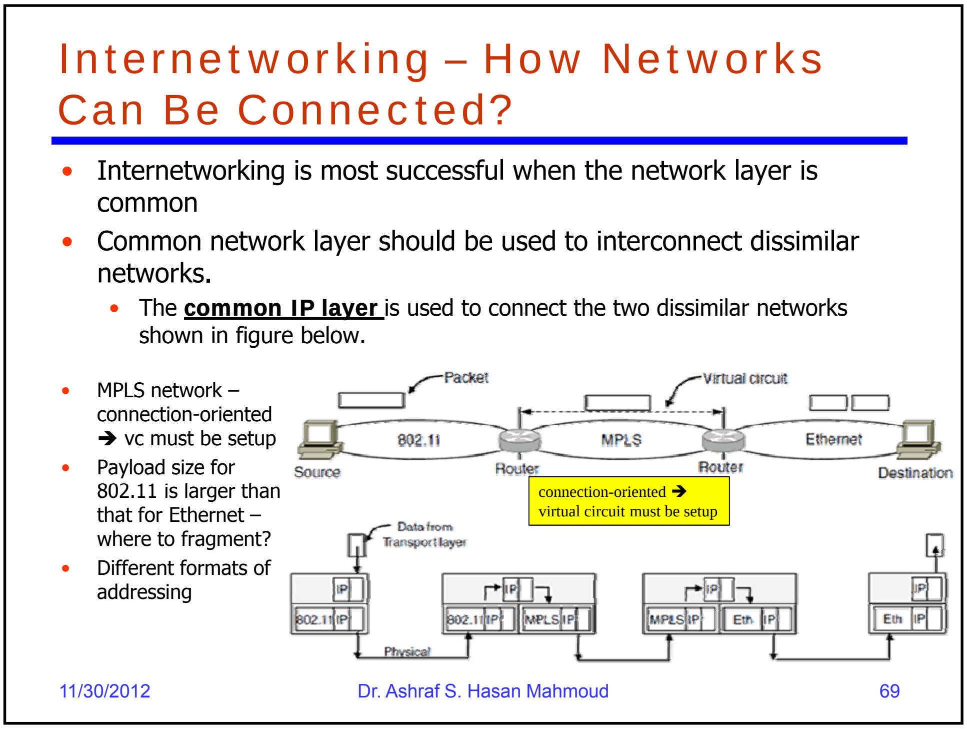
- More elaborate than expedited services
- Defines four priority classes and three discard classes
  - Top three priorities: gold, silver, and bronze
  - Discard classes for packets experiencing congestion: low, medium, and high
- ➔ 12 service classes
- Example: Refer to figure
  - Step 1: classify traffic into one of four classes
  - Step 2: determine the discard class using traffic policer
    - Packets that fit within small burst → LOW
    - Packets that exceed small burst → HIGH
  - Step 3: Encode the DiffServ mark into packet
  - Step 4: WFQ – popular choice
  - Step 5: Within each of the four priority classes, RED may start to drop packets as congestion builds



Internetworking

- Heterogeneous networks – multiple standards
- Value of network of N nodes – number of connections that may be made between the nodes ~ N^2.
  - Bigger networks are more valuable
  - Combine multiple smaller networks to get big networks
- Networks may be different – refer to table

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

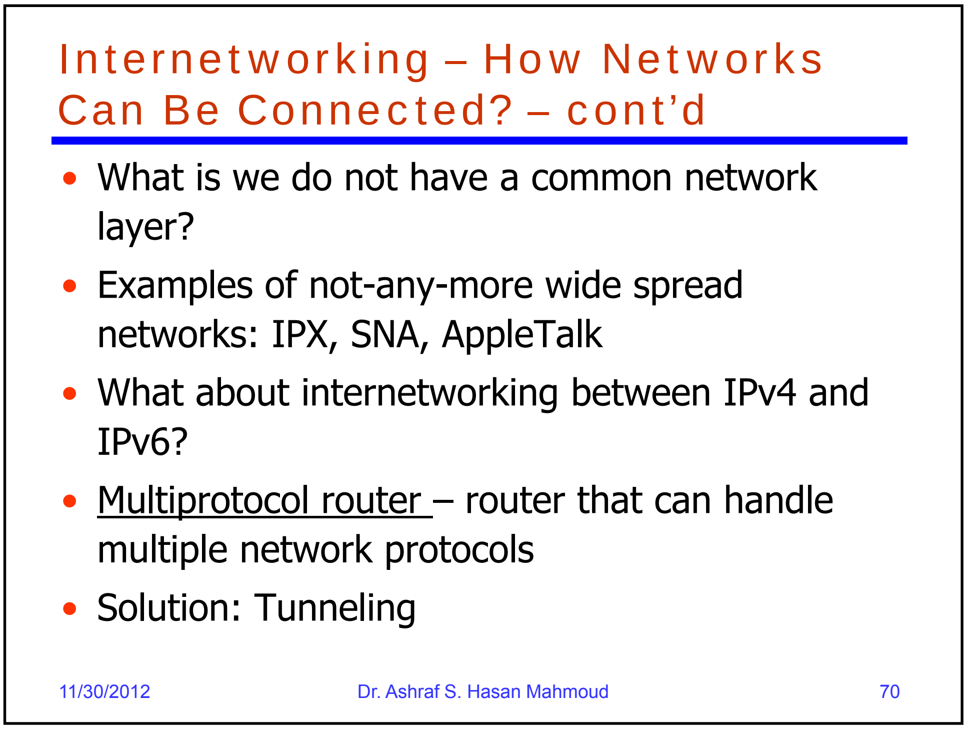


Internetworking – How Networks Can Be Connected?

- Internetworking is most successful when the network layer is common
- Common network layer should be used to interconnect dissimilar networks.
  - The **common IP layer** is used to connect the two dissimilar networks shown in figure below.

- MPLS network – connection-oriented ➔ vc must be setup
- Payload size for 802.11 is larger than that for Ethernet – where to fragment?
- Different formats of addressing

connection-oriented ➔ virtual circuit must be setup

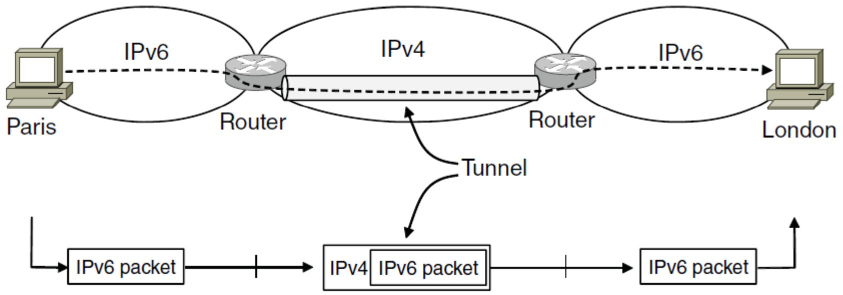


Internetworking – How Networks Can Be Connected? – cont'd

- What is we do not have a common network layer?
- Examples of not-any-more wide spread networks: IPX, SNA, AppleTalk
- What about internetworking between IPv4 and IPv6?
- Multiprotocol router – router that can handle multiple network protocols
- Solution: Tunneling

Internetworking - Tunneling

- Common solution to connect isolated hosts and networks using other networks
- The resultant network is referred to as an **overlay**
  - A network is overlaid on the base network
- **VPN** – an overlay network used to provide a measure of security



Internetworking – Interwork Routing

- Two-level routing protocol:
  - Intradomain or **Interior gateway protocol**
  - Interdomain or **Exterior gateway protocol** – for the Internet BGP is used
- **Autonomous System (AS)**: A network that is operated independently of all the others – e.g. some ISP network
- Nontechnical factors → **routing policy**: method of selecting routes amongst autonomous systems
  - Business agreements
  - Crossing international boundaries
  - Application of different laws in regard to traffic and content
  - Etc.

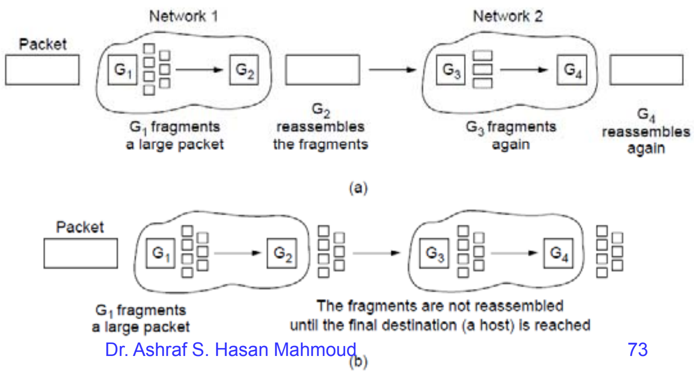
Internetworking – Packet Fragmentation

- Maximum payloads:
  - 1500 Bytes for Ethernet, 2272 Bytes for 802.11, 65,515 Bytes for IP, Etc.
- Path maximum transmission unit (Path MTU)
  - Source usually does not know the path ahead of time and does not know how small packets must be
  - Packets are routed independently
- Solution 1: allow routers to fragment the packet

(a) **Transparent** versus

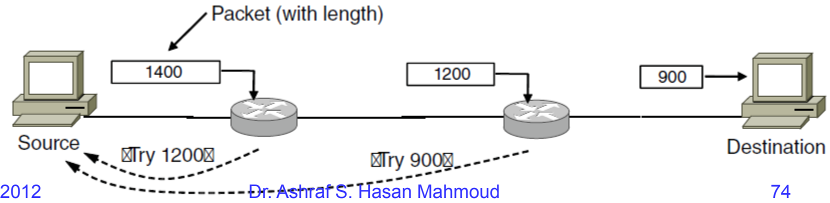
(b) **non transparent** fragmentation

- Pros and cons of each strategy


Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

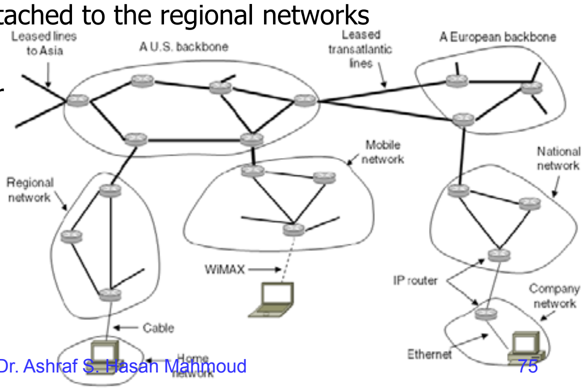
Internetworking – Packet Fragmentation – cont'd

- Solution 2: do not allow routers to fragment packets
- Path MTU discovery
  - IP packet sent with bits in header indicating no fragmentation is allowed
  - Receiving router either accepts to route the packet or sends an error packet back to the source
  - Source uses info in the error packet to design a smaller packet
  - Process is repeated



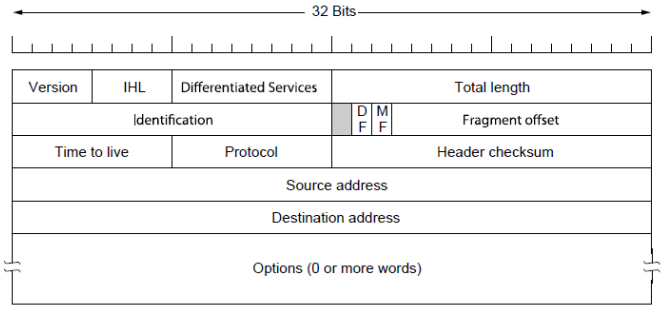
The Network Layer in the Internet

- The Internet – collection of Ases that are interconnected
  - Several major backbones – high-bandwidth lines and fast routers
  - Tier-1 networks – biggest of the backbones; other networks connect to Tier-1 to get connected to the Internet
- ISPs (mid-level) attach to the backbones - regional networks
- Third level ISPs are attached to the regional networks
- IP is the Network layer protocol – glue that holds the whole Internet



IPv4

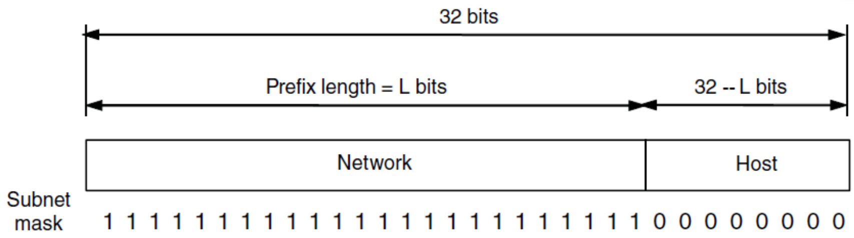
- Header: 20 bytes (fixed) + variable (optional) part
- Version – version 4
- IHL – how many 32-bit words the header is; min = 5 (no options), max = 15 (i.e. 60 bytes ➔ options = 40 bytes)
- Differentiated Services (aka Type of Service) – distinguish between different classes of service
- Total Length – datagram size in bytes – max = 65,535 bytes
- Identification – all fragments belonging to same packet have same identification
- DF – don't fragment; MF – more fragments
- Fragment offset – all fragments (except the last) are multiples of 8 bytes
- Time-to-Live – limit packet lifetime – max = 255
- Protocol – which transport protocol
- Header checksum – one's complement of the sum of the 16-bit words of the header
- Options – allow extensions or subsequent version of protocol


32 Bits

Version	IHL	Differentiated Services	Total length
Identification		D F / M F	Fragment offset
Time to live		Protocol	Header checksum
Source address			
Destination address			
Options (0 or more words)			

IP Addresses

- IPv4 uses 32-bit addresses
- Hierarchical structure: L bits (**prefix**) for network address + (32-L) bits for Host
    - Pros – scalability; all routes to same network in the same direction
    - Cons – address depends on location; may waste blocks of addresses
- **Subnet mask** – may be ANDED with the IP address to extract only the network portion.



IP Addresses - Subnets

- Internet Corporation for Assigned Names and Numbers (**ICANN**) – managing network addresses
- **Subnetting** – process of splitting an address block to multiple networks for internal use, while still acting like a single network to the outside word
- Example – block of /16 is divided into
    - A /17 block (one half) for CS department – Mask = 255.255.128.0

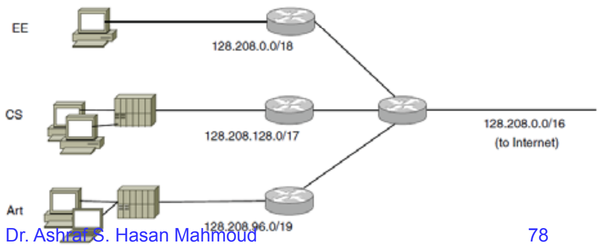
      `128.208.0.0/17 = 1000 0000  1101 0000  1xxx xxxx  xxxx xxxx`
    - A /18 block (one quarter) for EE department – Mask = 255.255.192.0

      `128.208.0.0/18 = 1000 0000  1101 0000  00xx xxxx  xxxx xxxx`
    - A /19 block (one eighth) for Art – Mask = 255.255.96.0

      `128.208.96.0/19 = 1000 0000  1101 0000  011x xxxx  xxxx xxxx`

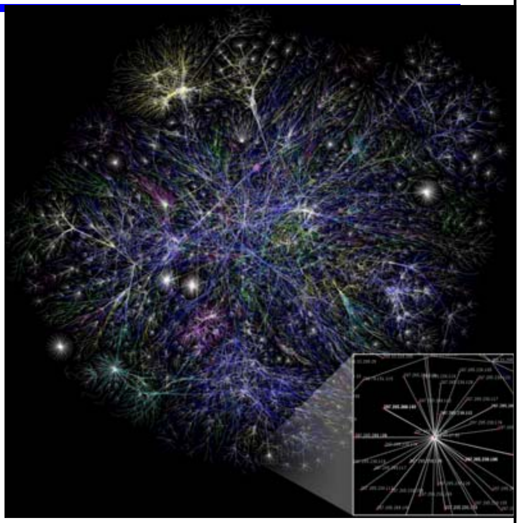
subnets

- Where to route 128.208.2.151?
- And with EE mask ➔ 128.208.0.0 address of EE network
- Subnetting is only visible within the 128.208.0.0/16 network!



IP Addresses – CIDR

- CIDR – Classless InterDomain Routing
- Default-free zone – core routers of the internet
  - About million interconnected networks!
  - **routing table explosion**
- Solution: **route aggregation**
  - Combine multiple small prefixes into a single larger prefix → **supernet**
  - Example: the same IP address that one router treats as part of a /22 (a block of 2^{10} addresses) may be treated by another router as part of a larger /20 (a block of 2^{12} addresses)
  - It is up to the router to have the corresponding prefix information



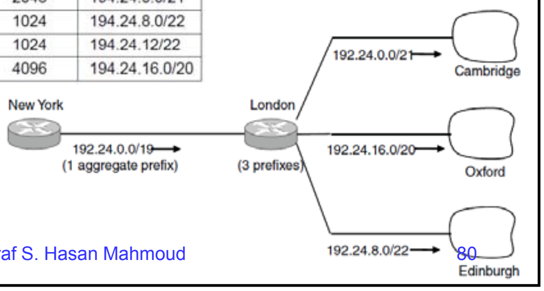
Source: http://en.wikipedia.org/wiki/File:Internet_map_1024.jpg

CIDR calculator
http://www.subnet-calculator.com/cidr.php

IP Addresses – CIDR – cont'd

- Example: address block of 8192 IP addresses available starting at 192.24.0.0 – distributed as shown in Table below →192.24.0.0/19
- All routers in the default-free zone are now told about the IP addresses in the three networks
  - Router London need to send on a different outgoing line for each of the prefixes – one entry per prefix
  - Router New York – single aggregate entry for the prefix 194.24.0.0/19 is advertized from Router London to Router New York
- Address aggregation is an automatic process

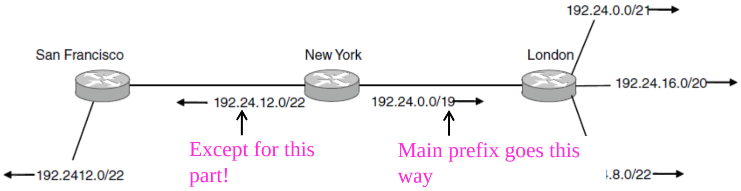
University	First address	Last address	How many	Prefix
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20



CIDR calculator
http://www.subnet-calculator.com/cidr.php

IP Addresses – CIDR – cont'd

- Prefixes are ALLOWED to overlap ➔ where to send?
- Packets are sent in the direction of the most specific route → **longest matching prefix** that has the fewest IP addresses
- When a packet arrives, the routing table is scanned to determine if the dst address lies within the prefix → use entry with longest match
  - Match for a /20 mask and a /24 mask → use the /24 mask
- Specialized hardware is used to speedup the table lookup.

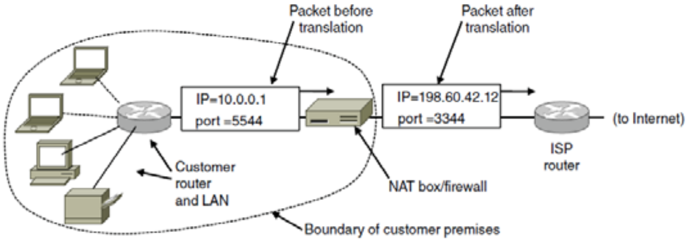


Except for this part!

Main prefix goes this way

IP Addresses – NAT

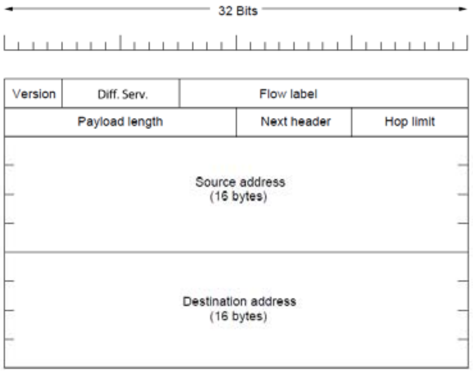
- **Network Address Translation** – NATing
- Three reserved ranges:
  - 10.0.0.0 – 10.255.255.255/8 (16,777,216 hosts)
  - 172.16.0.0 – 172.31.255.255/12 (1,048, 576 hosts)
  - 192.168.0.0 – 192.168.255.255/16 (65,536 hosts)
- Terms: NAT box, src and dst ports mapping
- 1) IP address is being reused – not unique
- 2) breaks the end-to-end connectivity model – incoming packets cannot be accepted until the mapping table is established!
- NAT box maintains a "kind" of state information



IPv6 and Header Structure

- Standardized since 1998
- Deployed and used on only a tiny fraction of the Internet (estimated 1%)
- Main features (over IPv4):
  1. 128-bit address space
  2. Simplified header – faster processing of packets
  3. Better support for options
  4. Big advancement in security

- Header (40 bytes) fields
  - Version
  - Diff. Services
  - Flow label
  - Payload length
  - Next header – which of the 6 addition headers follow this IP packet. If the IP packet is the last one, then Next header fields indicates which transport protocol is used!
  - Hop limit
  - Src/Dst address fields



Internet Control Protocol

- ICMP – Internet Control Message Protocol
- ARP – Address Resolution Protocol
- DHCP – Dynamic Host Configuration Protocol

Internet Control Message Protocol (ICMP)

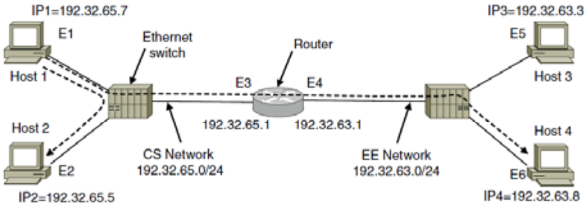
- Routers report events to sender using ICMP
- Principal ICMP messages are listed below
- Example 1– message destination unreachable is used when router cannot locate the destination or when a packet with DF bit cannot be delivered because a size is large.
- Example 2 - message Time exceeded – when a packet is dropped because its TtL counter reaches zero → tracerout!
- Example 3 – message Echo and echo reply →ping

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 and Echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

Address Resolution Protocol (ARP)

- Ethernet network interface card (NIC) has 48-bit MAC address
  - All traffic on LAN segment is addressed using MAC addresses
- How to map IP addresses to MAC addresses?
- Refer to the example -

- Example 1: Host 1 sends one packet to Host 2
  - Host 1 knows the name for the destination "eagle.cs.uni.edu"
  - Using DNS (to be studied later) it obtains the IP address for Host 2
  - Build packet with IP address of Host 2 as destination
  - Host 1 uses ARP to find the corresponding MAC for the IP of Host 2
  - Builds a MAC frame with MAC for Host 2
  - Host 1 sends MAC frame to Host 2

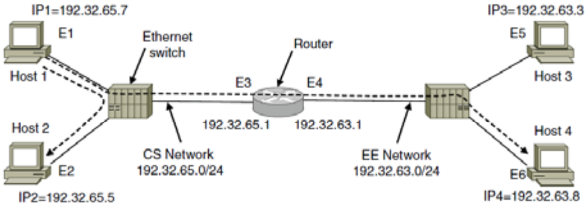


Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Address Resolution Protocol (ARP) – Optimizations

- Cashing with time-to-live entries – why?
- Host 1 sends its MAC along – not only Host 2 can update it ARP but all machines on the LAN
- Gratuitous ARP
  - When a machine is configured with a new IP
  - It issues an ARP inquiring about its IP (broadcasting its own MAC-IP mapping)
  - All machines listening to the broadcast perform cache update
  - If a response is received ➔ IP conflict, i.e. two machines with same IP address
- Example 2: Host 1 wants to send packet to Host 4

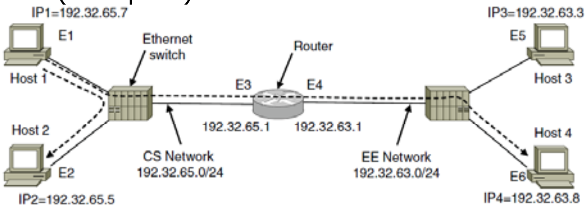
- Host 1 knows IP address of Host 4 – Wanted IP is on a different network
- Host 1 forwards packet to **default gateway**
- Host 1 uses ARP to get the MAC address of the router interface to the 192.32.65.1
- Router receives the MAC frame, realizes the dst IP is on 192.32.63.1 network
- Router uses ARP to get the MAC address of Host 4
- Router sends MAC frame to Host 4



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Address Resolution Protocol (ARP) – Optimizations – cont'd

- Proxy ARP –
  - Host 1 does not know that Host 4 is on a different network
  - Host 1 uses ARP to get the MAC address of Host 4
  - Router responds by providing its MAC address on the 192.32.65.1 network
  - Rest continues as before (Example 2)



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

DHCP

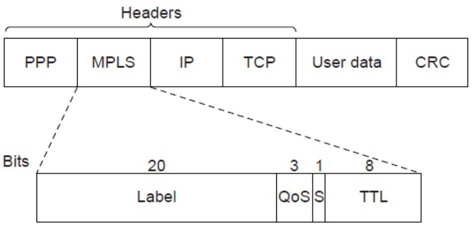
- DHCP server
- Client sends a broadcast DHCP DISCOVER packet
- Server receives broadcast and allocates a free IP – sends DHCP OFFER
- Leasing – IP address assignment is done for a fixed period of time
    - Just before the lease expires, the host may ask for a DHCP renewal
- A mechanism to configure hosts with needed parameters; e.g. network mask, IP of default gateway, IP address of DNS servers
- Defined in RFC 2131 and 2132

MultiProtocol Label Switching (MPLS)

- An alternative routing technology – starting to be widely used, especially in ISP networks
- Similar to circuit-switching
- Label added in front of each packet
    - Label acts as an index to an internal table specifying the output line – routing ➔ mere table lookup
- MPLS header – 4 bytes with four fields

- Label – holds the index
- QoS – class of service
- S – stacking multiple labels
- TtL – time-to-live

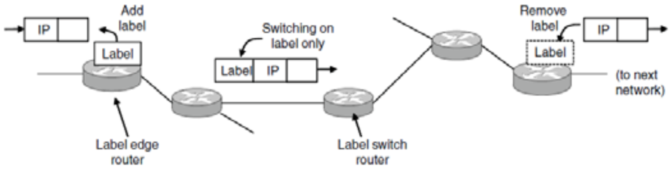
- MPLS – layer 2.5 (i.e. independent of both IP and datalink)
    - MPLS switch may forward both IP packets and non-IP packets ➔ "**<u>Multiprotocol</u>**"



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

MultiProtocol Label Switching (MPLS) – cont'd

- Label Edge Router (LER) – edge of an MPLS network; attaches an MPLS header to the IP packet
  - Inspects the dst IP address
  - Decides on the MPLS path the packet should follow
  - Assigns a label to be used for forwarding
- On the other edge of the network the LER remove the MPLS header and forwards the IP packet
- Label Switched Router (LSR)
  - Label is used to decide on the output line and also the new label to use (label switching)
  - Labels have only local significance (refer the slides on virtual-circuit routing)



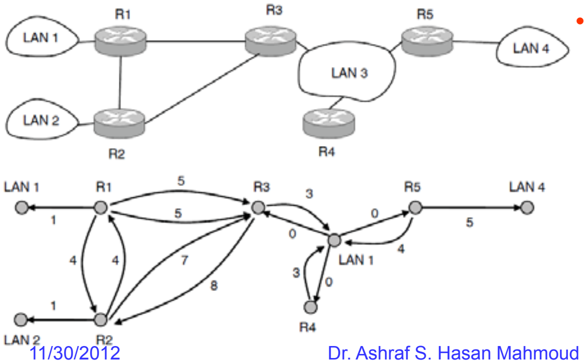
Forwarding an IP packet through an MPLS network

MultiProtocol Label Switching (MPLS) – cont'd

- Forwarding Equivalence Class (FEC) – routers group multiple flows that end at a particular router or LAN and use a single label for them
- With virtual-circuit routing it is not possible to group several distinct paths with different endpoints onto the same virtual-circuit identifier – there would be no way to distinguish then at the final destination
  - This is the not case for MPLS – dst IP address
- MPLS can be stacked (using the header bit S)
  - Different treatment in some particular zone depending on the label

Open Shortest Path First (OSPF)

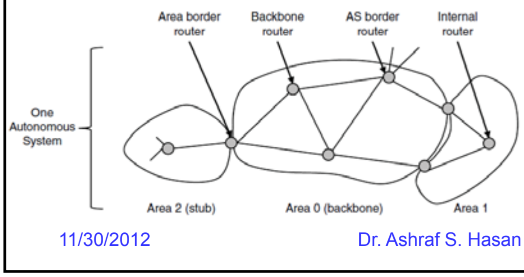
- OSPF standardized 1990 – link state routing protocol
  - Borrows a lot from IS-IS
- OSPF is widely used in enterprise networks while IS-IS is widely used in ISP networks
- Long list of requirements (open, support for variety of distance metrics, dynamic and adapt to topology changes, support for type of service, etc.)



- Abstract network into a directed graph
  - Point-to-point link are represented by a pair of arcs
  - A broadcast network is represented a node for the network itself, plus a node for each router – arcs from the network node to the routers have weight 0
  - Networks that have only hosts have only an arc reaching them

Open Shortest Path First (OSPF) – cont’d

- OSPF uses link state algorithm on the graph to find shortest path from src node to every other node
- OSPF keeps multiple shortest paths, if any – load balancing – Equal Cost Multipath (ECMP)
- A single AS is divided into numbered areas
  - Area – set of contiguous networks
  - Some routers may belong to no area
- Internal routers – routers lie wholly within an area
- Backbone area – area 0 → its routers are backbone routers
  - All areas are connected to area 0
- Area border router – area 0 routers connected to two or more areas
  - Summarize destinations in one area and inject this summary into other areas it is connected to

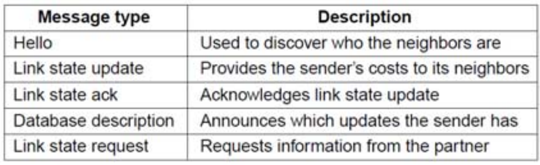


- Stub area – area with one area border router connecting to the backbone area
- AS boundary router – router that injects routes to external destinations on other Ases into the area
- One router may play multiple roles

Open Shortest Path First (OSPF) – Basic Operation

- Each router within an area has the same link state database and runs the same shortest path algorithm
- Area border routers needs the databases for all the area to which it is connected
  - It runs the algorithm for each area separately
- (Src, Dst) Routing:
  - Within same area – best intra-area route is chosen
  - Different area – inter-area route must go from the source to the backbone, across the backbone to the destination area, and then to the destination – **star configuration** of OSPF
- Five types of OSPF messages
- When a router boots, it sends HELLO messages on all of its point-to-point lines and multicasts them on LANs → Router learns who its neighbors are
- Information is exchanged between *adjacent* routers
- Designated router – backup designated router

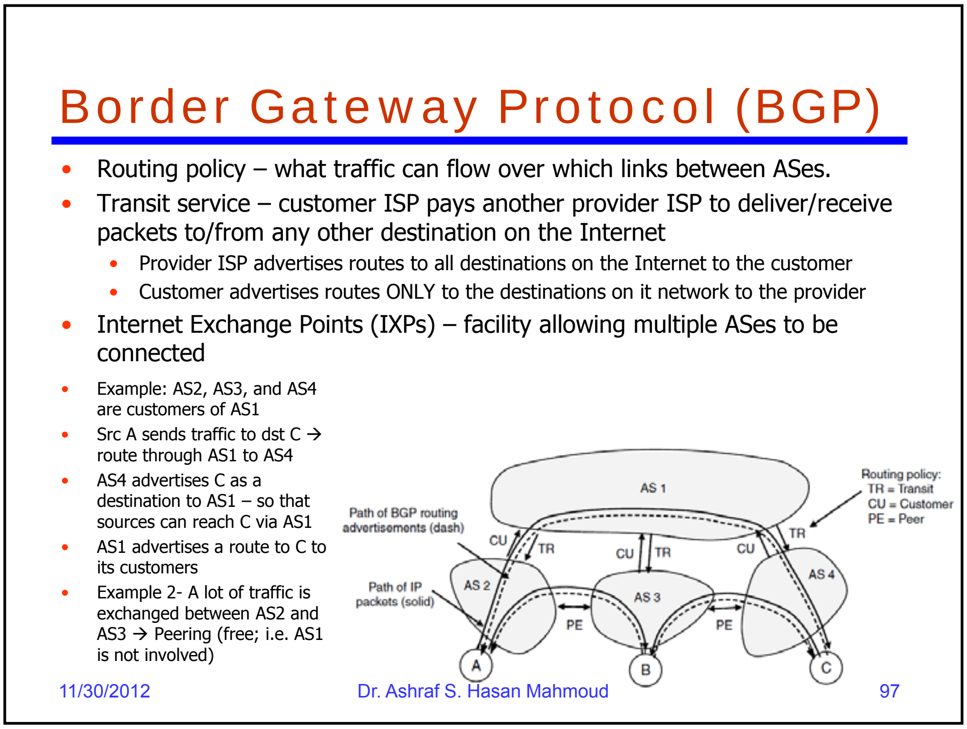
- Each routers periodically floods LINK STATE UPDATE message to each of its adjacent routers
  - Sequence #s + ACK
- DATABASE DESCRIPTION – gives the sequence #s of all the link state entries currently held by sender
- LINK STATE REQUEST – either party can request link state info



Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

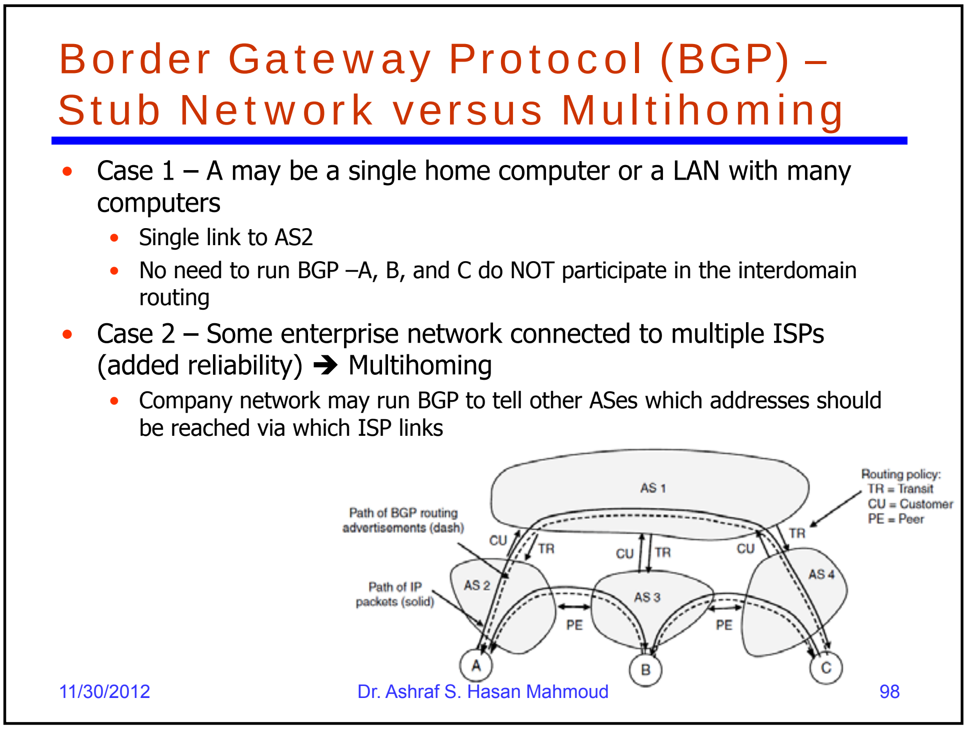
Open Shortest Path First (OSPF) – Basic Operation – cont'd

- Flooding is used – each router informs other routers in its area of it links to other routers and networks
- Routers construct the graph for its area(s) and shortest paths are computed
- Backbone routers do the above procedure in addition to accepting info from the area border routers in order to compute the best route from each backbone router to every other router
  - This info is communicated back to the area border routers which advertise within their area
- Internal routers can select best route to a destination outside their area, including best exit router to the backbone



Border Gateway Protocol (BGP)

- Routing policy – what traffic can flow over which links between ASes.
- Transit service – customer ISP pays another provider ISP to deliver/receive packets to/from any other destination on the Internet
  - Provider ISP advertises routes to all destinations on the Internet to the customer
  - Customer advertises routes ONLY to the destinations on it network to the provider
- Internet Exchange Points (IXPs) – facility allowing multiple ASes to be connected
- Example: AS2, AS3, and AS4 are customers of AS1
- Src A sends traffic to dst C → route through AS1 to AS4
- AS4 advertises C as a destination to AS1 – so that sources can reach C via AS1
- AS1 advertises a route to C to its customers
- Example 2- A lot of traffic is exchanged between AS2 and AS3 → Peering (free; i.e. AS1 is not involved)

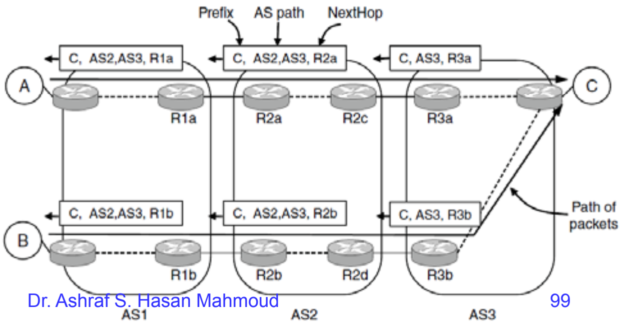


Border Gateway Protocol (BGP) – Stub Network versus Multihoming

- Case 1 – A may be a single home computer or a LAN with many computers
  - Single link to AS2
  - No need to run BGP –A, B, and C do NOT participate in the interdomain routing
- Case 2 – Some enterprise network connected to multiple ISPs (added reliability) ➔ Multihoming
  - Company network may run BGP to tell other ASes which addresses should be reached via which ISP links

Border Gateway Protocol (BGP) – Route Advertisements and Selection

- Similar to distance vector routing



Not for focus in 111 offering

Internet Multicasting

- TBC

Not for focus in 111 offering

Mobile IP

- TBC