Computer Networks

Chapter 5
Network Layer 1

Prof. Jerry Breecher
CSCI 280
Spring 2002

The Weeks Ahead

Mar 11	Chapter 5.1: Network Layer
Mar 13	Chapter 5.1
Mar 18	EXAM 2
Mar 20	Chapter 5.1:
Mar 21	LAB – You should have several tests running.
Mar 25	Chapter 5.2: More Network Layer
Mar 27	Chapter 5.2:
Apr 1	Chapter 5.2
Apr 3	Chapter 6.1: Transport Layer
Apr 8	Chapter 6.1:
Apr 10	EXAM 3
Apr 15	Chapter 6.1:
Apr 17	Chapter 6.1:
Apr 22	Chapter 6.1:
Apr 24	Chapter 6.1:
Apr 25	LAB – Drop Dead Date!!
May 3	Final Exam – 8:00 – 10:00

Chapter Overview

The Network Layer is concerned about getting packets from source to destination, no matter how many hops it may take. It's all about routing.

5.1 Network Layer Design Issues

What do we need to think about in this layer?

5.2 Routing Algorithms

Strategies for getting from source to destination.

5.3 Congestion Control Algorithms

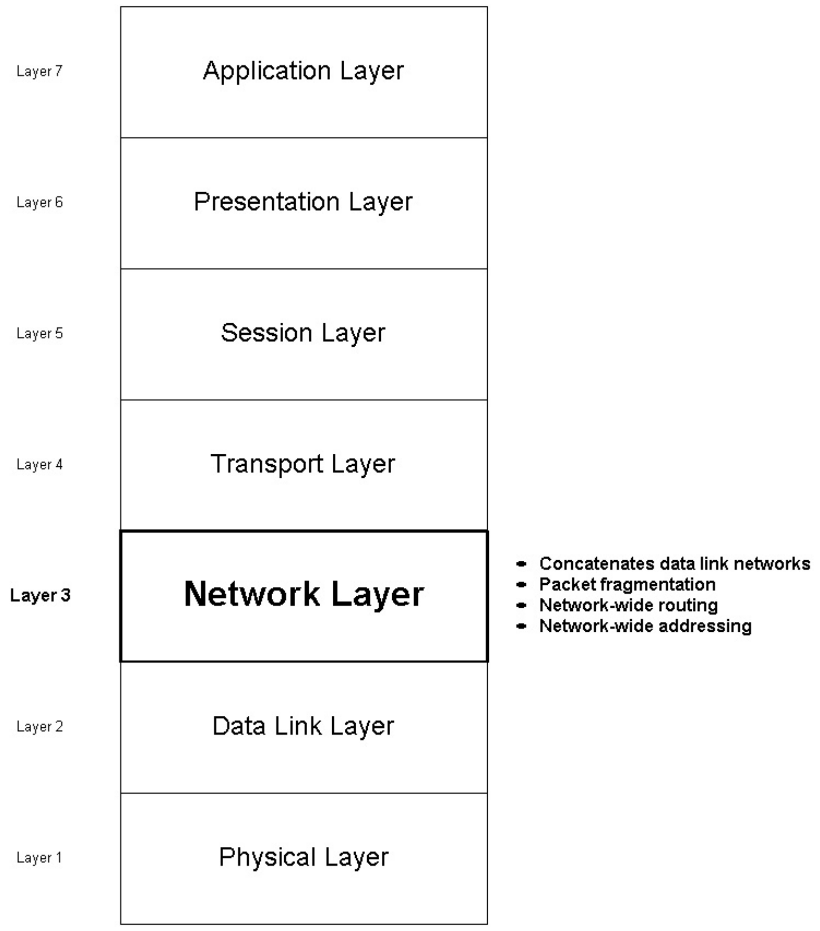
How do we keep from bottlenecking from too many packets?

5.4 Internetworking

Working with multiple networks and protocols in order to deliver packets.

5.5 The Network Layer in the Internet

Gluing together a collection of subnets.



Overview

5.1 Network Layer Design Issues
5.2 Routing Algorithms
5.3 Congestion Control Algorithms
5.4 Internetworking
5.5 The Network Layer in the Internet

Who does what with regard to:

1. Routing issues.
2. Congestion
3. Internetworking

We also throw in a few definitions.

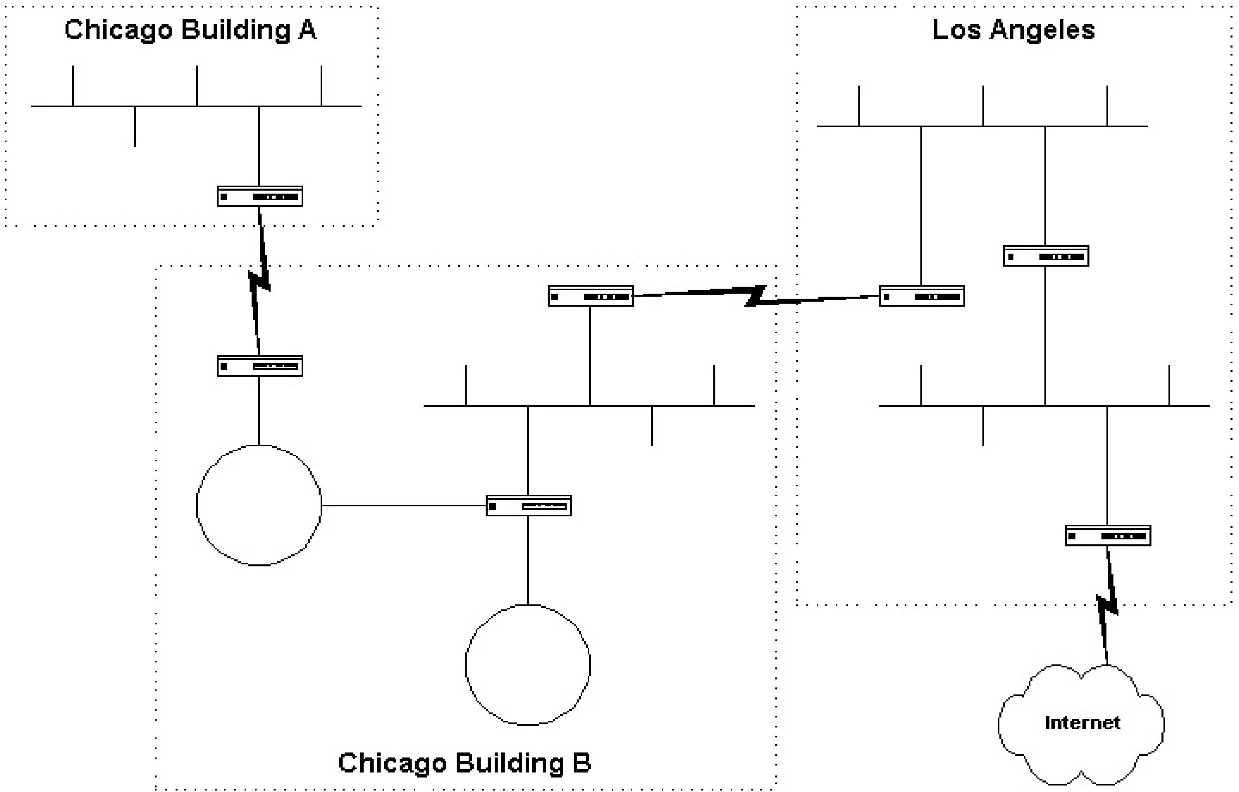
And examine differing views of the world.

Network Layer Design Issues

Overview

WHO-DOES-WHAT-ISSUES:

- The network layer is responsible for routing packets from the source to destination.
- The routing algorithm is the piece of software that decides where a packet goes next (e.g., which output line, or which node on a broadcast channel).
- For connectionless networks, the routing decision is made for each datagram. For connection-oriented networks, the decision is made once, at circuit setup time.



Network Layer Design Issues

Overview

ROUTING ISSUES:

The routing algorithm must deal with the following issues:

1. Correctness and simplicity: networks are never taken down; individual parts (e.g., links, routers) may fail, but the whole network should not.

2. Stability: if a link or router fails, how much time elapses before the remaining routers recognize the topology change? (Some never do..)

3. Fairness and optimality: an inherently intractable problem.
   a) Definition of optimality usually doesn't consider fairness.
   b) Do we want to maximize channel usage? Minimize average delay?

When we look at routing in detail, we'll consider both adaptive-those that take current traffic and topology into consideration-and nonadaptive algorithms.

Network Layer Design Issues

Overview

CONGESTION (NOT CONTENTION):

The network layer also must deal with congestion:

When more packets enter an area than can be processed, delays increase and performance decreases. If the situation continues, the subnet may have no alternative but to discard packets.

If the delay increases, the sender may (incorrectly) retransmit, making a bad situation even worse.

Overall, performance degrades because the network is using (wasting) resources processing packets that eventually get discarded.

INTERNETWORKING:

When we consider internetworking, connecting different network technologies together, there are the same problems, only worse:

1. Packets may travel through many different networks
2. Each network may have a different frame format
3. Some networks may be connectionless, other connection oriented.

Services Provided to the Transport Layer

The network layer should shield the transport layer from having to know details of the underlying subnet

The Transport layer should do exactly the same thing, whether sending across LAN or across the country on the Internet.

Should the host or the subnet be responsible for the delivery of all packets in order ?

Services Provided to the Transport Layer

Review of Definitions:

Connection-Oriented Service: The subnet, with the help of the network layer, should provide the following operations:

1. The sending side of the pair should open a connection with its peer.
2. This connection has properties negotiated by the pair.
3. Communication is bi-directional and packets are delivered in order.
4. Flow control is accomplished in the subnet.

Connectionless Service: The subnet has no state information. It does only send_packet and receive_packet. Error control and flow control are done by the host (network or higher layers.)

Virtual Circuit: Avoids choosing a new route for each packet. A virtual circuit is a state -- it remembers how to send a packet from source to destination. This state is held in the subnet, in the source DLL or in each of the DLL layers along the route.

Datagrams: Each packet sent is routed independently of its predecessors. Decisions are made "on the fly", so more computing required, however this method is more robust.

Network Layer Design Issues

TWO VIEWS OF THE WORLD

Two views on the question:

"Does error control belong in network or transport layer?"

DARPA Internet community viewpoint:

- The subnet is inherently unreliable no matter how it is designed.
- Thus, hosts are forced to do error control anyway.
- Given that they perform error control, why have the network layer duplicate the same function?
- The DARPA TCP/IP Internet is connectionless in its implementation but provides connections to its users.

Common carrier viewpoint:

- The connection oriented approach is the right way.
- Users don't want complex error control protocols in host computers.
- User's want reliable, trouble-free service. (X.25)

Network Layer Design Issues

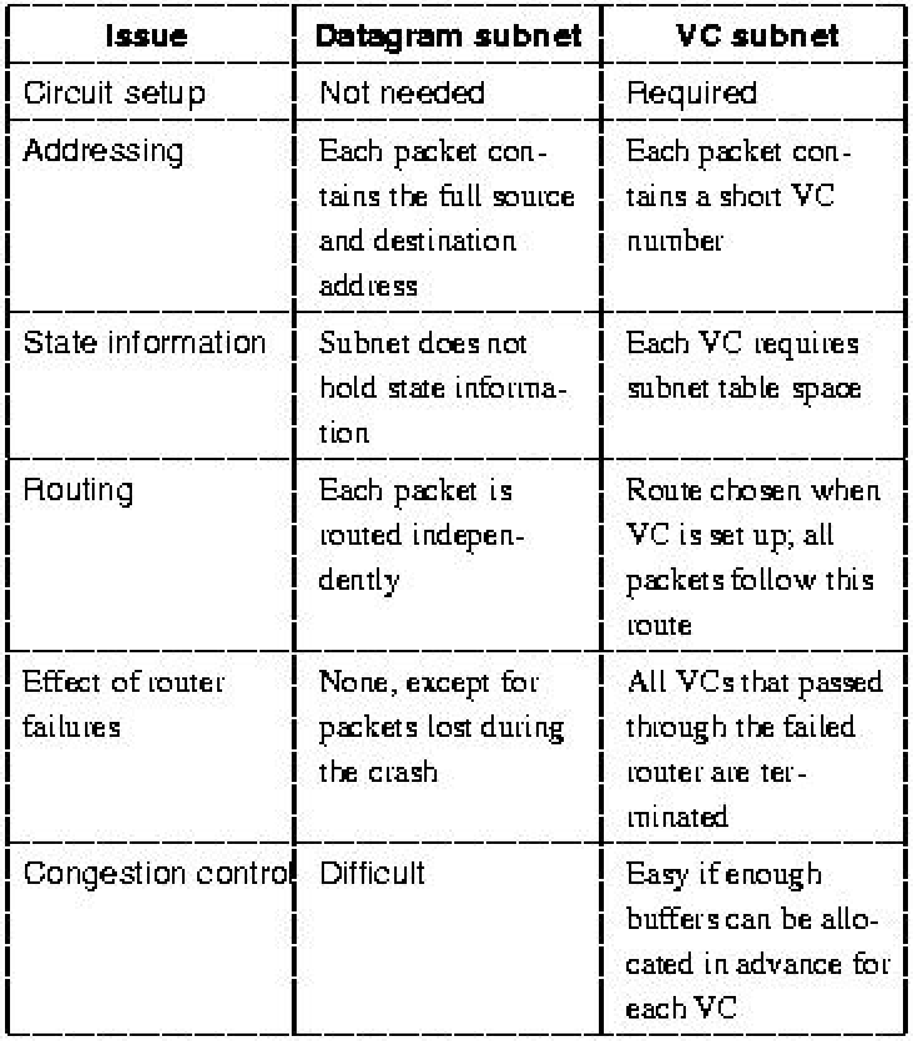
TWO VIEWS OF THE WORLD

COMPARISON OF VIRTUAL CIRCUITS AND DATAGRAMS:

Note that:

- Connection Oriented Service generally use Virtual Circuits.
- Connectionless Service uses Datagrams
- But can mix and match as desired.

See the Figure for additional properties.



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

Routing Algorithms

Overview

5.1 Network Layer Design Issues
5.2 Routing Algorithms
5.3 Congestion Control Algorithms
5.4 Internetworking
5.5 The Network Layer in the Internet

What matters when doing routing?

What are various techniques to do that routing?

Which is best?

Routing Algorithms

Overview

Routing is concerned with the question: Which line should router J use when forwarding a packet to router K ?

There are two types of algorithms:

Adaptive algorithms - use such dynamic information as current topology, load, delay, etc. to select routes.

Nonadaptive algorithms - routes never change once initial routes have been selected. Also called static routing.

Either of these algorithms can be applied to either datagrams or virtual circuits.

Obviously, adaptive algorithms are more interesting, as nonadaptive algorithms don't even make an attempt to handle failed links.

Routing Algorithms

Adaptive algorithms can be further divided in the following types:

1. Isolated: each router makes its routing decisions using only the local information it has on hand. Specifically, routers do not even exchange information with their neighbors.

2. Centralized: a centralized node makes all routing decisions. Specifically, the centralized node has access to global information.

3. Distributed: algorithms that use a combination of local and global information.

Goals for any of these algorithms include:

1. Correctness, simplicity and minimality.
2. Robustness. During years of continuous operation, being able to handle all kinds of hardware and software failures. Being able to handle changes in topology and traffic patterns.
3. Stability. Of the algorithms. Can you get a mathematical and realistic answer.
4. Fairness and optimality. Often contradictory.

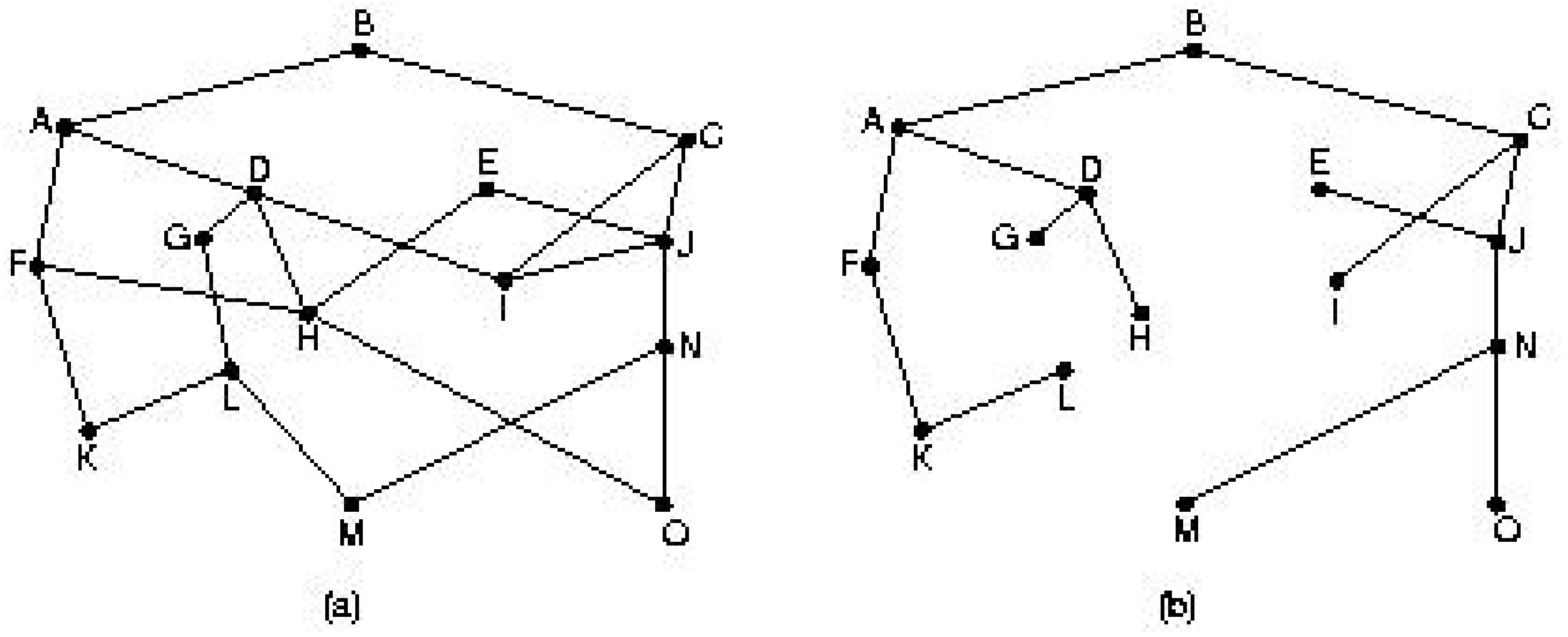
Routing Algorithms

Optimality Principle

THE OPTIMALITY PRINCIPLE:

This simply states that 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 this same path.

This means we can form a sink tree as in the Figure.



Routing Algorithms

Shortest Path

Often used because simple and easy to understand.

What if we `know' the complete topology of the network? Can look at computing the optimal path.

What if we have the following network and we want to route a packet from node A to node G. What is the shortest path (do not initially show distance).

Use Dijkstra's algorithm (or variation). Basic idea is:

- Choose the source, and put nodes connected to source in list to consider.
- From the list to consider choose the nearest node.

Non-Adaptive Algorithm

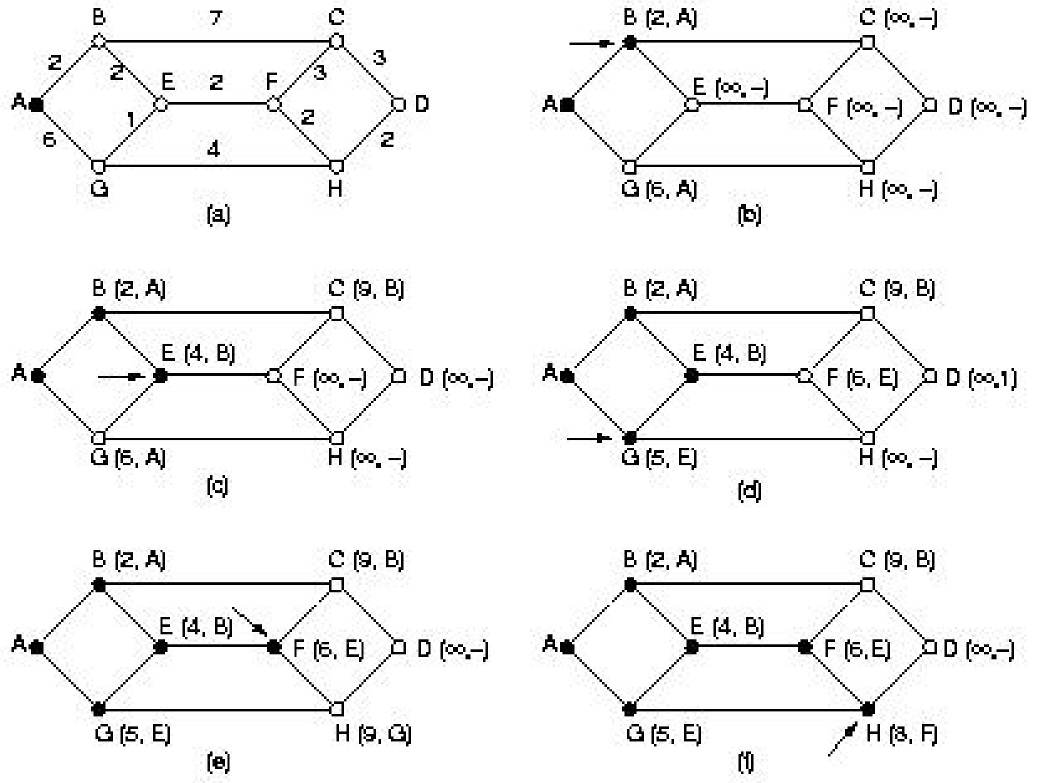
Routing Algorithms

Shortest Path

Let's do example in Figure 5.6.

Algorithm results:

nodes	list to consider
-----	----------------
(A, 0, -)	
(B, 2, A)	
(C, 9, B)	
(D, 10,H)	
(E, 4, B)	
(F, 6, E)	
(G, 5, E)	
(H, 8, F)	



Guaranteed to get the shortest path? How to prove? If an alternate shorter path to a node then we would have already tried the path.

Non-Adaptive Algorithm

Routing Algorithms

Flooding

Flooding is a form of isolated routing. Does not select a specific route. When a router receives a packet, it sends a copy of the packet out on each line (except the one on which it arrived):

To prevent packets from looping forever, each router decrements a hop count contained in the packet header. Whenever the hop count decrements to zero, the router discards the packet.

Non-Adaptive Algorithm

To reduce looping even further:

1. Add a sequence number to each packet's header.

2. Each router maintains a private sequence number. When it sends a new packet, it copies the sequence number into the packet, and increments its private sequence number.

3. For each source router S, a router:

   a) Keeps track of the highest sequence number seen from S.
   b) Whenever it receives a packet from S containing a sequence number lower than the one stored in its table, it discards the packet.
   c) Otherwise, it updates the entry for S and forwards the packet on

Routing Algorithms

Flooding

Flooding has several important uses:

1. In military applications, the network must remain robust in the face of (extreme) hostility

2. Sending routing updates, because updates can't rely on the correctness of a router's routing table.

3. Theoretical-chooses all possible paths, so it chooses the shortest one.

In selective flooding, a router sends packets out only on those lines in the general direction of the destination. That is, don't send packets out on lines that clearly lead in the wrong direction.

Non-Adaptive Algorithm

Routing Algorithms

Flow Based Routing

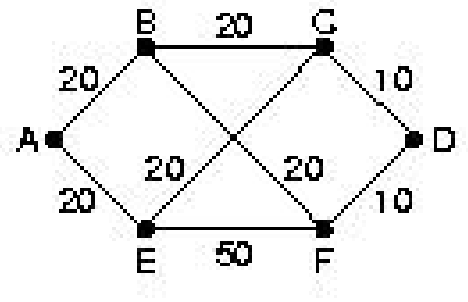
Takes into account both the topology AND the load (BUT still static.)

Assumes:

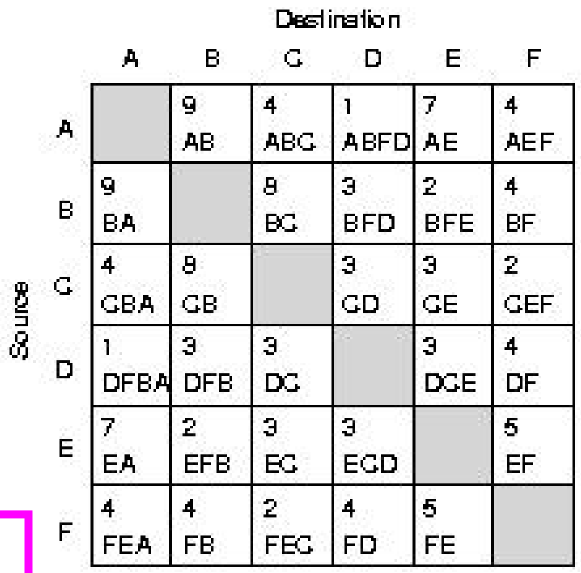
1. Traffic flows remain constant over time.
2. We can estimate the flow between all pairs of routers.
3. We know the topology of the network and the capacity of each link.

Given the line capacity and the flow, we can determine the delay. From that, we can calculate the delay for the whole subnet, and between any two nodes.

Use as an example figures Figure 5.8, 5.9.



(a)



Destination

Source	A	B	C	D	E	F
A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
B	9 BA		8 BC	3 BFD	2 BFE	4 BF
C	4 CBA	8 CB		3 CD	3 CE	2 CEF
D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
E	7 EA	2 EFB	3 EC	3 ECD		5 EF
F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

Non-Adaptive Algorithm

Routing Algorithms

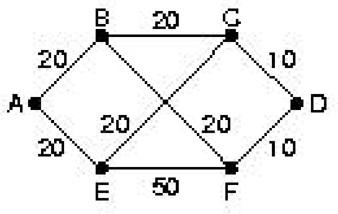
Flow Based Routing

Non-Adaptive Algorithm

Need to use the formula for delay time:

$$T = \frac{1}{\mu C - \lambda}$$

where $1/\mu$ = is the mean packet size in bits,
λ = mean number of arrivals in packets/sec.
and C = line capacity (bits/sec).



(a)

Destination

Source	A	B	C	D	E	F
A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
B	9 BA		8 BC	3 BFD	2 BFE	4 BF
C	4 CBA	8 CB		3 CD	3 CE	2 CEF
D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
E	7 EA	2 EFB	3 EC	3 ECD		5 EF
F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

Routing Algorithms

Adaptive Routing

In general, the methods are:

1. **Centralized** - uses a routing control center (RCC). Creates, modifies, and distributes routing tables to other routers. Gathers information from the routers.

   a) Good: adaptive routing, relieve burden on the routers of computing tables.

   b) Problems: Does not adapt quickly.

   c) Quicker the adaptation, the more overhead it causes.

   d) Synchronization of updates (some routers change, but not others, so could have situation where two routers send at each other).

   e) If RCC crashes the network becomes stale.

   f) Lines near the RCC are overloaded.

Adaptive Algorithm

Routing Algorithms

Adaptive Routing

2. Decentralized - Base decisions on local traffic and conditions.

Hot potato-choose output line with the shortest queue

Backward learning - each packet contains source address and number of hops so far. Use this information to learn shortest path to each source. Will learn shortest path to all routers. Only deal with good news, not bad. `Good' may no longer be good due to down router or congestion.

Must periodically forget and start over (with sub optimal performance after a purge).

Adaptive Algorithm

Routing Algorithms

Distance Vector Routing

Distributed routing algorithm. Routers work together.

1. Each router maintains a table (vector) giving the best known distance to a destination and the line to use for sending there. Tables are updated by exchanging information with neighbors.

2. Each router knows the distance (cost) of reaching its neighbors (e.g. send echo requests).

3. Routers periodically exchange routing tables with each of their neighbors.

4. Upon receipt of an update, for each destination in its table, a router:

   – Compares the metric in its local table with the metric in the neighbor's table plus the cost of reaching that neighbor.

   – if the path via the neighbor has a lower cost, the router updates its local table to forward packets to the neighbor.

Adaptive Algorithm

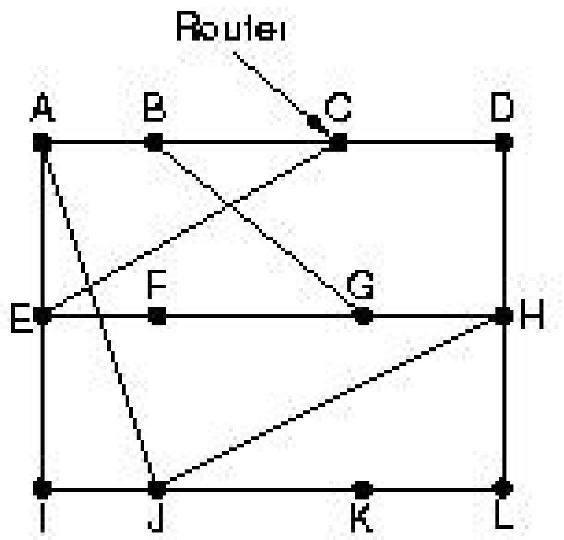
Routing Algorithms

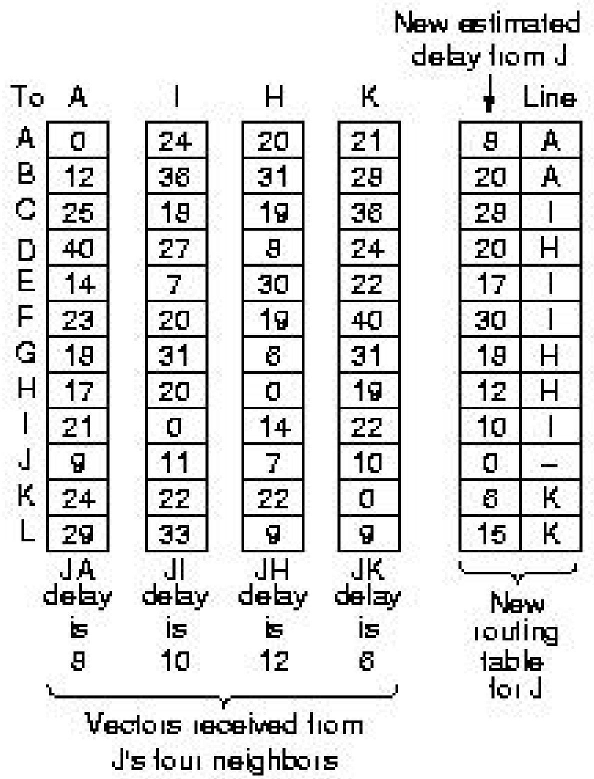
Distance Vector Routing

Use Figure 5.10 as an example.

This algorithm was used in the original ARPANET. Unfortunately, it suffers from the problem: good news travels quickly, bad news travels slowly (count-to-infinity problem).

The fundamental problem with the old Arpanet algorithm is that it continues to use `old' information that is invalid, even after newer information becomes available.





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		

Vectors received from J's four neighbors

New routing table for J

Adaptive Algorithm

Routing Algorithms

Link State Routing

The `old' Arpanet routing algorithm was replaced in 1979. Problems with old algorithm included:

1. High-priority routing update packets were large, adversely affecting traffic.
2. Network was too slow in adapting to congestion, too fast to react to minor changes.
3. Average queue length was used to estimate delay.
   - This works only if all lines have the same capacity and propagation delay.d
   - Doesn't take into account that packets have varying sizes.

Adaptive Algorithm

Routing Algorithms

Link State Routing

In the new algorithm:

1. Each router maintains a database describing the topology and link delays between each router. That is, each router keeps track of the full graph of links and nodes.

2. Each router periodically discovers it neighbors
   - Sends `hello' message on booting.
   - Measures the delays across its links (echo requests-should load be taken into account?),
   - Forwards that information to all other routers (link state packets).
3. Avoids the count to infinity problem since all routers get each other router's information.

4. Updates are propagated at high priority using flooding.
   - Updates contain sequence numbers, and a router forwards `new' copies of the packet.
   - Why use flooding? Because that way routing updates propagate even when routing tables aren't quite correct.
   - ACKs are sent to neighbors.

5. Each router uses an SPF algorithm to calculate shortest paths based on the current values in its database.

6. Because each router makes its calculation using the same information, better routing decisions are made.

Adaptive Algorithm

Routing Algorithms

Link State Routing

Limitations of new algorithm:

1. Doesn't take link capacity into consideration.
   - A 1200 baud link with 100 ms delay is favored over an Ethernet having an 200 ms delay.
2. Doesn't scale well, as each router receives updates from all other routers.
   - Today, we need to think of scaling to a system with a million nodes and many more links!
   - After all, 5 billion people will (eventually) be on the network!

Adaptive Algorithm

Routing Algorithms

Hierarchical Routing

One of the fundamental issues regarding routing is scaling.

a) As a network becomes larger, the amount of information that must be propagated increases, and the routing calculation becomes increasingly expensive.

b) Obviously, there are limits to how big a network can be.

Hierarchical routing is an approach that hides information from far-away nodes, reducing the amount of information a given router needs to perform routing:

Divide the network into regions, with a router only knowing the details of how to route to other routers in its region.

a) In particular, a router does not know about the internal topology of other regions.

b) Gateway is a router that knows about other regions.

A node in each region is designated as an entry point, and the entry point knows how to reach the entry points in all the other regions.

When traffic flows from A to B, it actually follows the path

A - AENTRY - BENTRY - B,

Adaptive Algorithm

where AENTRY and BENTRY are the entry points to the respective regions.

Routing Algorithms

Hierarchical Routing

Advantage: Scaling. Each router needs less information (table space) to perform routing.

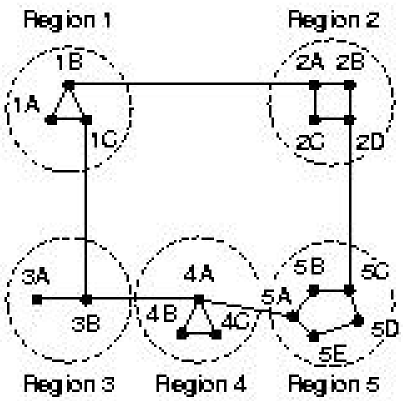
Disadvantage: Sub optimal routes. The average path length increases because there may be a shorter path that bypasses the entry points, but we don't use it.

See the Figure.

Hierarchical routing can be extended to multi-levels.

Example: telephone system:

- Area code identifies a region.
- Area code plus the first three digits identify the central office within a specific region.



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

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

Adaptive Algorithm

Routing Algorithms

Broadcast Routing

Adaptive Algorithm

Sending a packet to all destinations simultaneously is known as **broadcasting**.

There are several ways to implement broadcasting:

For Broadcast Networks:

1. The implementation is trivial: designate a special address as the `all hosts address'.

For Non-Broadcast Networks:

1. Send a unicast packet to each destination. However, this approach makes poor use of resources.

2. Flood packets to all nodes. Flooding generates many packets and consumes too much bandwidth.

3. Use multi-destination routing:
   a) Each packet contains a list (or bitmap) of all destinations, and when a router forwards a packet across two or more lines, it splits the packet and divides the destination addresses accordingly.
   b) This approach is similar to sending uni-cast packets, except that we don't send individual copies of each messages.
   c) However, the copy operations slow down the ability of a router to process many packets.

Routing Algorithms

Broadcast Routing

4. **Use a spanning tree**. If the network can be reduced to a tree
   - a) (There's only one path between any two pairs of routers), copy a packet to each line of spanning tree except the one on which it arrived.
   - b) Works only if each router understands the same spanning tree.

5. **Reverse Path Forwarding (RPF):**
   - a) Use a sink tree (assume sink/source trees are the same).
   - b) If a packet, originating from X, arrives on a line of the sink tree leading to X, the packet is traveling along the shortest path, so it "must" be the first copy we've seen.
   - c) Copy the packet to all outgoing lines of the sink tree.

If the packet arrives on another line, assume that the packet is a copy - it didn't arrive on the shortest path - and discard it.

RPF is easy to implement and makes efficient use of bandwidth.

Adaptive Algorithm

Routing Algorithms

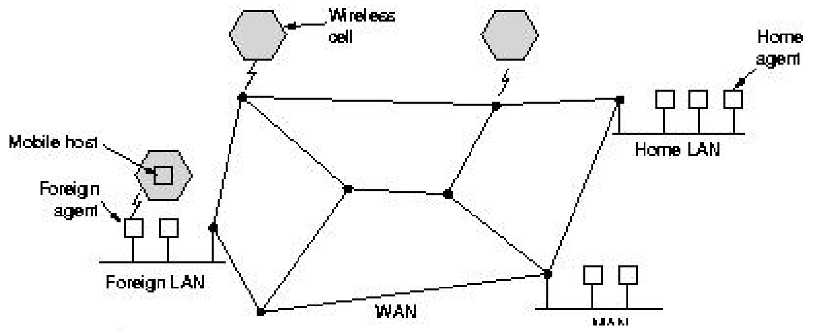
Other Routing

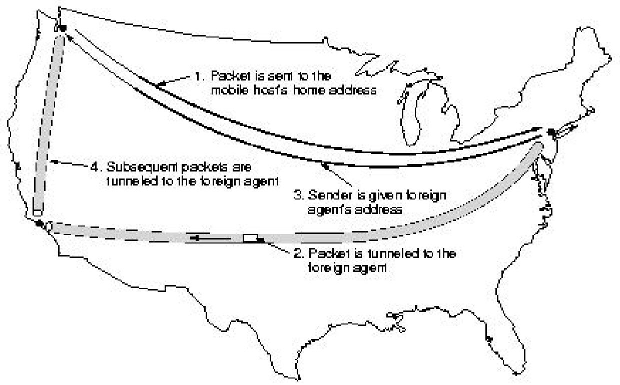
Mobile Hosts - machines that are not currently connected to their home location. Must get traffic to base host. Need intermediary agents. Look at Figures 5.18, 5.19. Base host uses encapsulation (tunneling) to send packet to mobile host.

Areas have: **Foreign Agents** - keep track of mobile users visiting an area.
Home Agent - keep track of users whose home is in the area but are away.

The algorithm for this is:

1. Foreign agents broadcast packet advertising their existence; or mobile host can ask where there's a Foreign Agent.
2. Mobile host registers with Foreign Agent.
3. Foreign agent contacts the Mobile Host's home agent, telling that Home Agent that messages for the Mobile Host should be sent to the Foreign Agent.
4. Home Agent checks security.
5. Foreign Agent now receives any traffic for the Mobile Host.




1. Packet is sent to the mobile host's home address
4. Subsequent packets are tunneled to the foreign agent
3. Sender is given foreign agent's address
2. Packet is tunneled to the foreign agent

Congestion Control

Overview

5.1 Network Layer Design Issues

5.2 Routing Algorithms

5.3 Congestion Control Algorithms

5.4 Internetworking

5.5 The Network Layer in the Internet

What's the problem?

What polices are needed?

Congestion Tactics: Control and Prevention

Congestion Control

CONGESTION MECHANISMS:

When one part of the subnet (e.g. one or more routers in an area) becomes overloaded, congestion results. Congestion can be dealt with by:

1. Congestion control - when the problem occurs, limit senders or reroute. This is a reactive mode. (No, this is not an antihistamine!!)

2. Congestion prevention - make resources available and enforce good behavior so that the congestion won't occur.

Note thse definitions:

Congestion control means making sure the subnet can handle the offered traffic.

Flow control means preventing one sender from overflowing one receiver. Congestion deals with wires and routers, while flow deals with hosts.

Congestion Control

Overview

CONGESTION CONTROL POLICIES:

The approach followed will include:

1. Monitor the system to detect when and where congestion occurs.

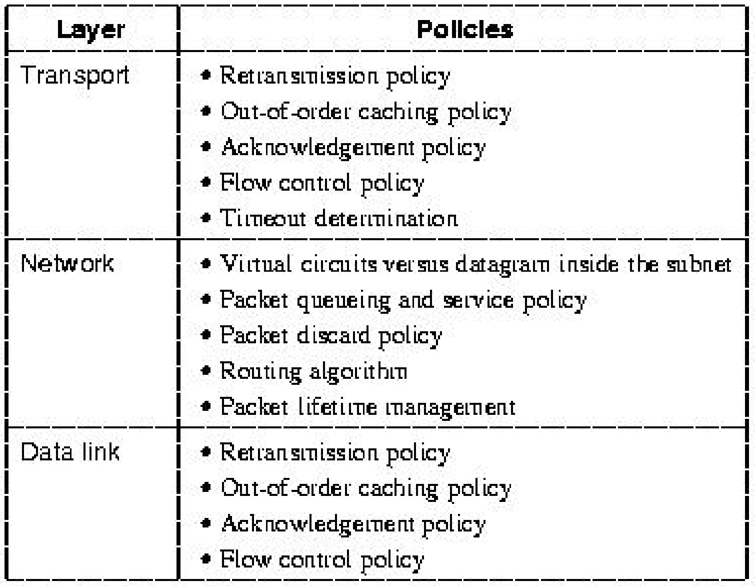
2. Pass this information to where something can be done about it.

3. Adjust the system. This can be done by:

   – The subnet must prevent additional packets from entering the congested region until those already present can be processed.

   – The congested routers can discard queued packets to make room for those that are arriving.

See Figure 5.23 for mechanisms that affect congestion.



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

Congestion Control

Congestion Prevention Policies

Pre-allocation schemes aim to prevent congestion from happening in the first place.

- For example, we can require that resources be pre-allocated before any packets can be sent, guaranteeing that resources will be available to process each packet.

In virtual circuit networks, for example, the sender opens a connection before sending data.

- The circuit setup operation selects a path through the subnet, and each router on the path dedicates buffer space and bandwidth to the new circuit.

What happens when a user attempts to open a virtual circuit and the subnet is congested?

- The subnet can refuse to open the connection, forcing the user to wait until sufficient resources become available.

Note: The ability of the subnet to reject requests to open connections is an important property of connection oriented networks.

Congestion Control

Traffic Shaping

Control the rate at which packets are sent (not just how many). Widely used in ATM networks.

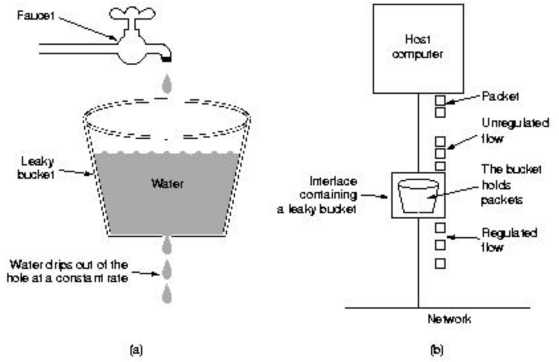
At set up, the sender and carrier negotiate a traffic pattern (shape).

Leaky Bucket Algorithm used to control rate in a datagram network. Fig. 5.24.
- A single-server queue with constant service time.
- If bucket (buffer) overflows then packets are discarded.

Enforces a constant output rate regardless of burstiness of input. Does nothing when input is idle.

The Token Bucket Algorithm causes a token to be generated periodically, which during idle periods can be saved up.

Related to traffic shaping is flow specification, where a particular quality of service is agreed upon between sender, receiver and carrier.



Congestion Control

Flow Control

Flow control is one way of preventing a fast sender from overwhelming a slow receiver. Flow control can be helpful at reducing congestion, but it can't really solve the congestion problem. For example, suppose we connect a fast sender and fast receiver (e.g., two Crays) using a 9.6 kbps line:

- If the two machines use a sliding window protocol, and the window is large, the link will become congested in a hurry.
- If the window size is small (e.g., 2 packets), the link won't become congested. Note how the window size limits the total number of packets that can be in transmission at one time.

Flow control can take place at many levels:

- User process to user process (end-to-end). Later, we'll see how TCP uses flow control at the end-to-end level.
- Host to host. For example, if multiple application connections share a single virtual circuit between two hosts.
- router to router. For example, in virtual circuits.

Congestion Control

Choke Packets

Routers can do the following:

1. Monitor the level of congestion around them
2. When congestion is present, they can send choke packets to the sender that say `slow down'.
3. How can a router measure congestion?
   – A router might estimate the level of congestion by measuring the percentage of buffers in use, line utilization, or average queue lengths.

Advantage:

1. Dynamic. Host sends as much data as it wants, the network informs it when it is sending too much.

Choke Packets

Disadvantages:

1. Difficult to tune.
   a) By how much should a host slow down?
   b) The answer depends on how much traffic the host is sending, how much of the congestion it is responsible for, and the total capacity of the congested region.
   c) Such information is not readily available in practice.

2. After receiving a choke packet….
   a) The sending host should ignore additional choke packets for a short
   b) Packets currently in transmission may generate additional choke packets.
   c) How long? Depends on such dynamic network conditions as delay.

Desirable:

More attention paid to reserving resources so that chances of congestion are reduced and the quality of service is more reliable.

Load Shedding

Opposite Approach:

- We could **preallocate** no resources in advance, and take our chances that resources will be available when we need them.
- When insufficient resources are present to process existing packets, discard queued packets to make room for newly arriving ones.

Who retransmits the discarded packets?

- In datagram (connectionless) networks, the sending host (transport layer) retransmits discarded packets (if appropriate).

- In virtual circuit networks, the previous-hop router retransmits the packet when it fails to receive an acknowledgment.

Congestion Control

Load Shedding

Failure to preallocate resources leads to two problems: potential **deadlock** and **unfairness**.

Deadlock. Suppose that all of a router's buffers hold packets.

- Because the router has no free buffers, it cannot accept additional frames.
- Unfortunately, it also ignores frames containing ACKs that would free up some of those buffers!
- If two adjacent routers, A and B, are sending packets to each other, since both are waiting for the other to accept a packet, neither can proceed.
- This condition is known as a deadlock.

Solution: Reserve at least one buffer for each input line and use it to hold incoming packets. Note that we can extract the ACK field and still discard the packet, if we don't have buffers to hold it.

Congestion Control

Load Shedding

There's an advantage to discarding packets when congested: Easy to implement. Disadvantages center around fairness:

1. Wastes resources. The network may have expended considerable resources processing a packet that is eventually discarded.

2. Non-deterministic. There is less guarantee than with virtual circuits that packets will ever reach their destination.

3. Requires that sending hosts pay attention to congestion. If the network can't prevent a host from sending data, a host can overload the network. In particular, a `broken' host may cause the network to become overly congested.

4. In the extreme case, congestion collapse occurs.
   - The network becomes so overloaded, that few packets reach their destination.
   - Meanwhile, the sending hosts continue to generate more data (both retransmissions and new packets).
   - This condition occurred several times in 1987, and the Internet/Arpanet became unusable for a period of hours to days.