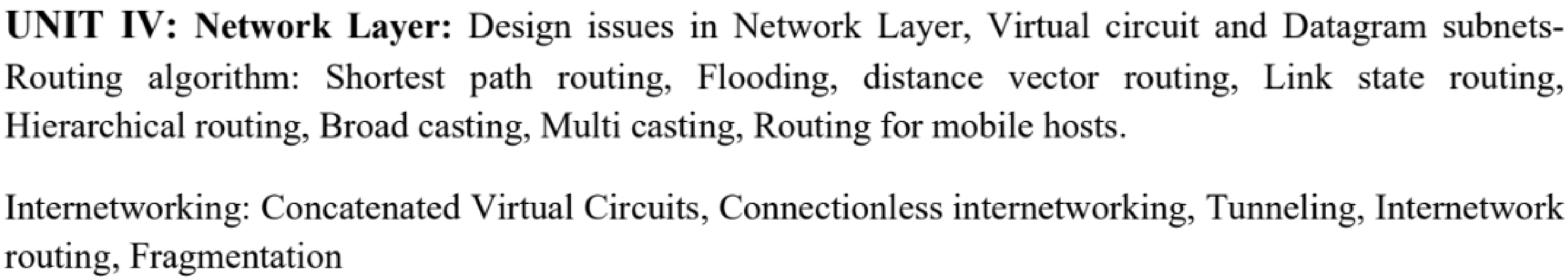
UNIT-4

The Network Layer



UNIT IV: Network Layer: Design issues in Network Layer, Virtual circuit and Datagram subnets-Routing algorithm: Shortest path routing, Flooding, distance vector routing, Link state routing, Hierarchical routing, Broad casting, Multi casting, Routing for mobile hosts.

Internetworking: Concatenated Virtual Circuits, Connectionless internetworking, Tunneling, Internetwork routing, Fragmentation

Contents….

- Design issues in Network layer
- Virtual circuit Vs Datagram subnets
- Routing Algorithms
- Internetworking

•The network layer is concerned about getting packets from source all the way to the destination.

•Thus it deals with end-to-end transmission.

•To achieve its goals, the network layer must know about the topology of the communication subnet and choose appropriate paths.

•It must also take care to choose routes to avoid overloading some of the communication lines and routers while leaving others idle.

•Finally, when source and destination are on different networks, new problems may arise. It is up to the network layer to deal with them.

Network Layer Design Issues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

Store-and-Forward Packet Switching

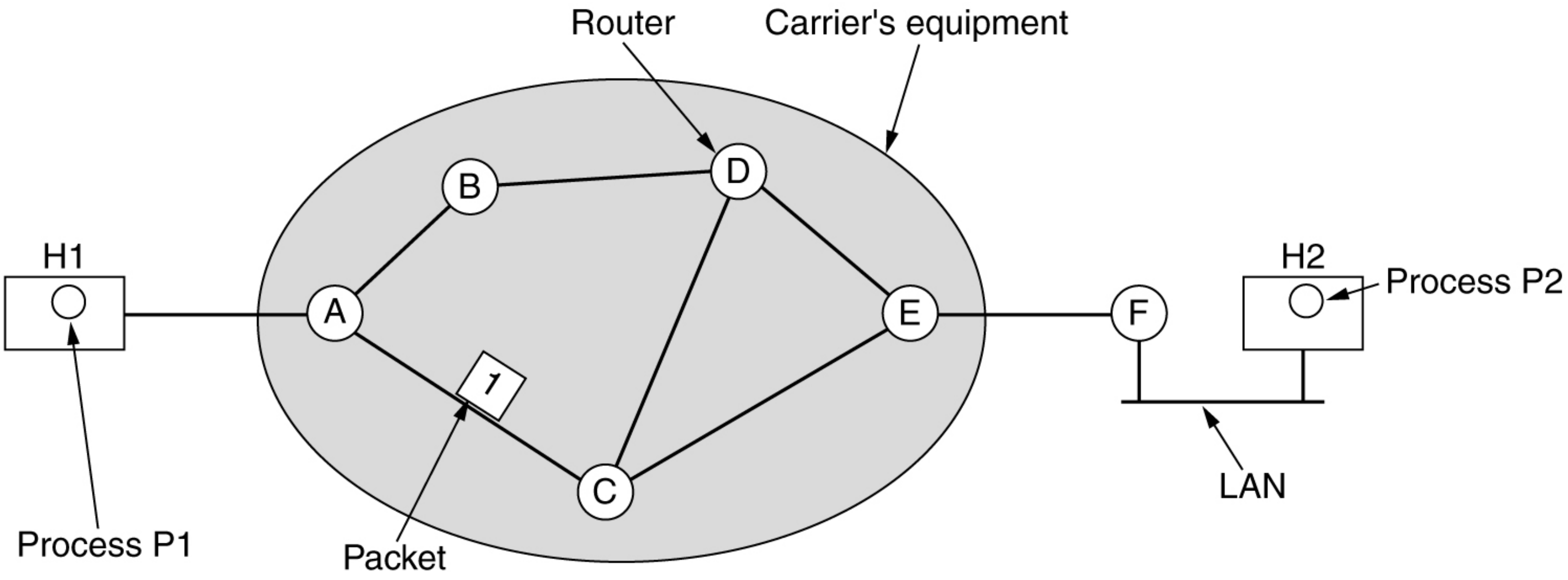


Fig 5.1The environment of the network layer protocols.

- Carrier equipment
- Store and forward packet switching

The equipment is used as follows:

- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.

- The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is called store-and-forward packet switching.

Services Provided to the Transport Layer

The Network layer services are designed with the following goals:

1. The services should be independent of the router technology.
2. The transport layer should be shielded from the number, type, and topology of the routers present.
3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Internet – Connection-less
ATM – Connection-oriented

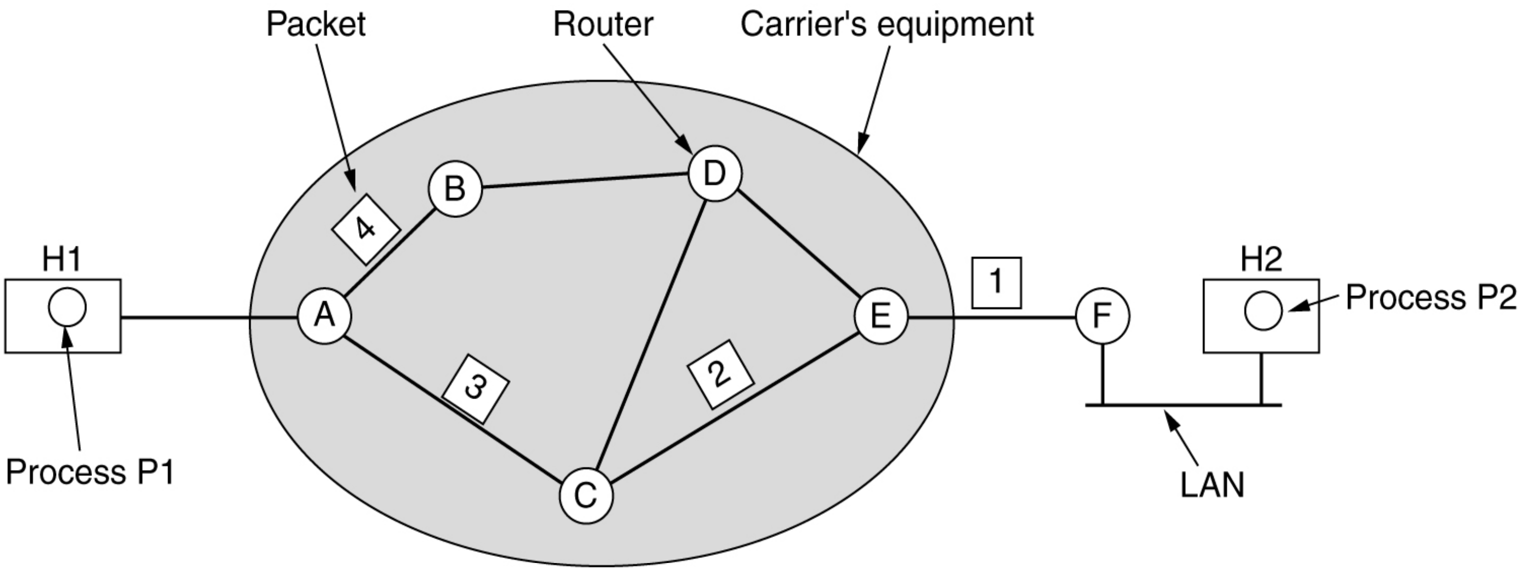
Implementation of Connectionless Service

.

1. If connection-less service is offered, packets are injected into the subnet individually and routed independently of each other.

2. No advance setup is needed.

3. In this context, the packets are called datagrams and the subnet is called datagram subnet.



A's table

initially		later		C's table		E's table	
A	–	A	–	A	A	A	C
B	B	B	B	B	A	B	D
C	C	C	C	C	–	C	C
D	B	D	B	D	D	D	D
E	C	E	B	E	E	E	–
F	C	F	B	F	E	F	F

Dest. Line

Fig 5.2. Routing within a datagram subnet

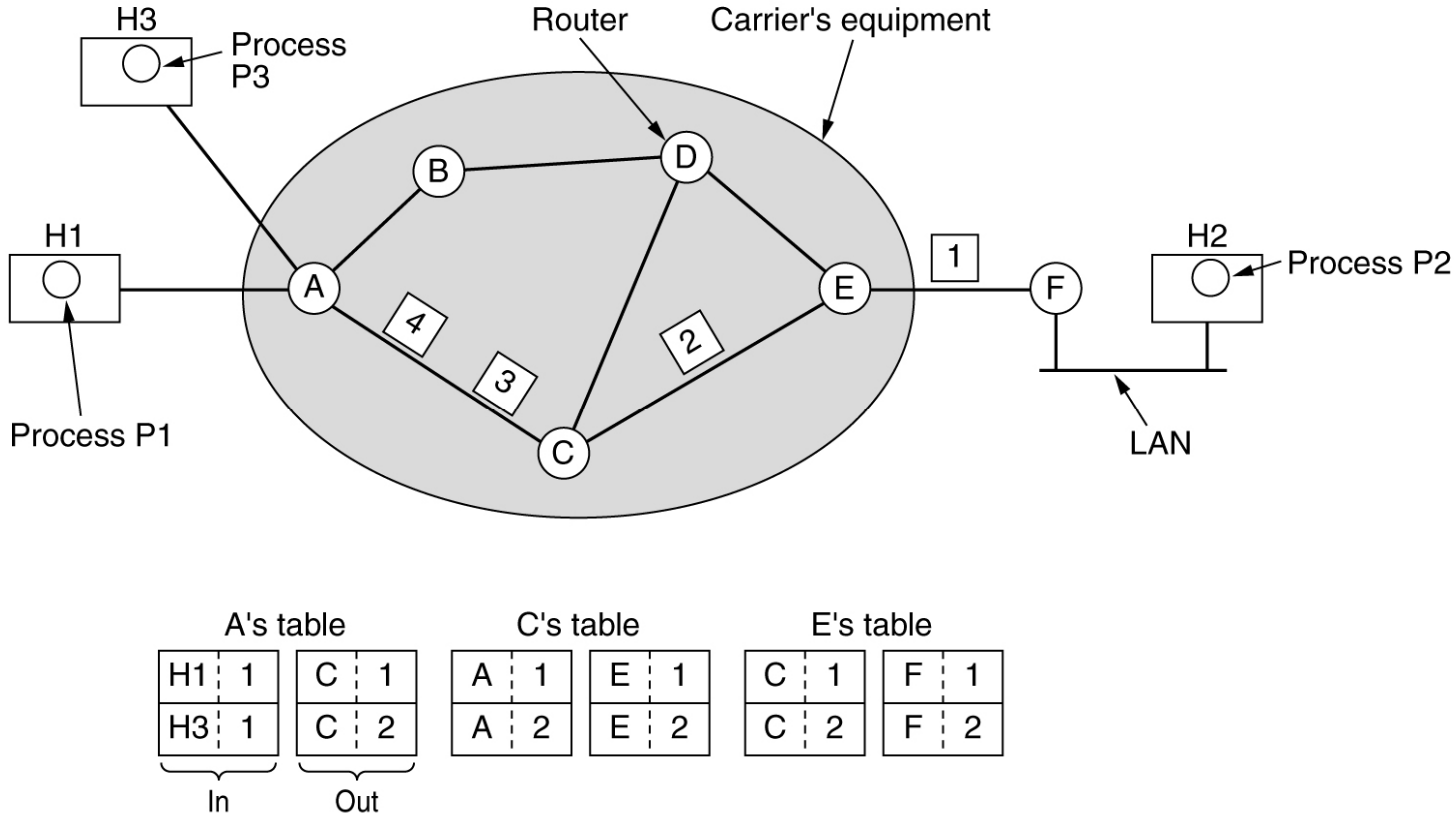
- Suppose that the process P1 has a long message for P2. It hands the message to transport layer with instructions to deliver it to process P2 on host H2.
- Let us assume that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1,2,3 and 4 and sends each of them in turn to router A using point-to-point protocol, for example, PPP. At this point the carrier takes over.
- Every router has an internal table telling it where to send packets for each possible destination.
- Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly-connected lines can be used.

- ‘A’ has only two out going lines- to B and C- so every incoming packet must be sent to one of these routers, even if the ultimate destination is some other router.
- A’s initial table is shown under the label “initially”.
- As they arrived at A, packets 1,2, and 3 were stored briefly(to verify checksum). Then each was forwarded to C according to A’s table.
- Packet 1 was then forwarded to E and then to F. When it got to F, it was encapsulated in a data link layer frame and sent to H2 over the LAN. Packets 2 and 3 follow the same route.
- When packet 4 got to A it was sent to router B, even though it is also destined for F.
- For some reason, A decided to send packet 4 via a different route. Perhaps it learned of a traffic jam somewhere along ACE path and updated its routing table as shown under the label “later”.
- The algorithm that manages the tables and makes the routing decisions is called the Routing algorithm

Implementation of Connection-Oriented Service

- For connection-oriented service, we need a virtual-circuit(VC) subnet
- The idea behind VC is to avoid having to choose a new route for every packet sent.
- Instead , when a connection is established, a route from source machine to destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- When the connection is released, the VC is also terminated.
- Here, each packet carries an identifier telling which virtual circuit it belongs to.
- For example consider the fig that is present in the next slide.

Implementation of Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Fig 5.3. Routing within a virtual-circuit subnet.

- Here, host H1 has established connection 1 with host H2.
- The first line of A’s table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1.
- Similarly, the first entry at C routes the packet to E, also with connection identifier 1.
- Now let us consider what happens if H# also wants to establish a connection to H2. It chooses connection identifier 1(because it is initiating the connection and this is its only connection) and tells subnet to establish the virtual circuit. This leads to second row in the tables.
- We have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this.

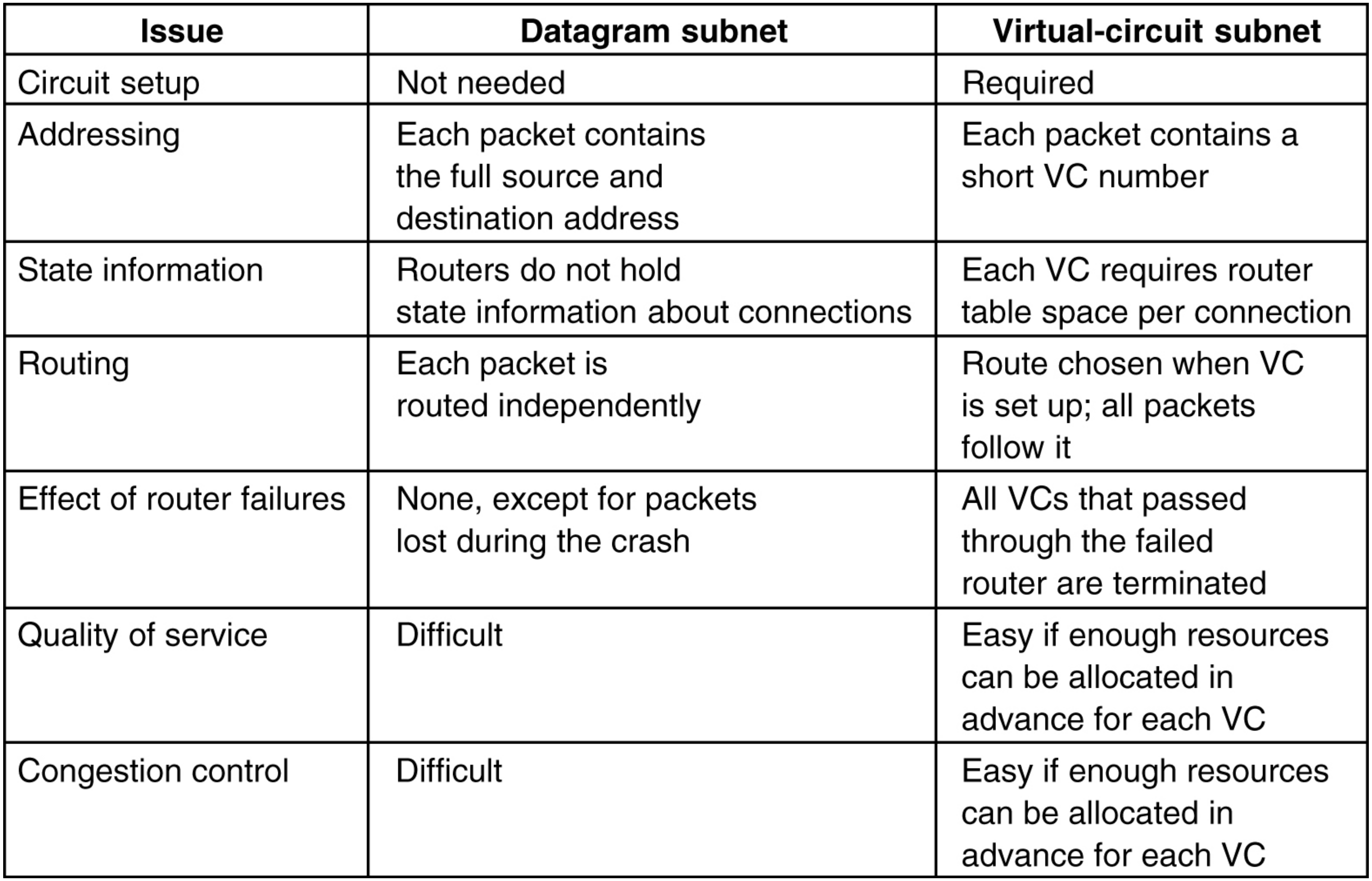
- For this reason, A assigns a different connection identifier to the out going traffic for the second connection.
- Avoiding conflicts of this kind is why routers need the ability to replace connection identifier in outgoing packets. This is called Label switching.

Comparison of Virtual-Circuit and Datagram Subnets

Inside the subnet, several trade-offs exist between virtual circuit and data-grams.

- Router memory space and bandwidth
- Setup time versus address parsing time
- Amount of table space required in router memory
- Routing
- Quality of service
- Effect of router failure
- Congestion Control

Comparison of Virtual-Circuit and Datagram Subnets



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

Routing Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts

Desirable Properties (Elaborate)

Routing versus Forwarding
Properties of routing algorithm…

1. Correctness
2. Simplicity
3. Robustness
4. Stability
5. Fairness
6. Optimality.

Correctness: The routing should be done properly and correctly so that the packets may reach their proper destination.

Simplicity: The routing should be done in a simple manner so that the overhead is as low as possible. With increasing complexity of the routing algorithms the overhead also increases.

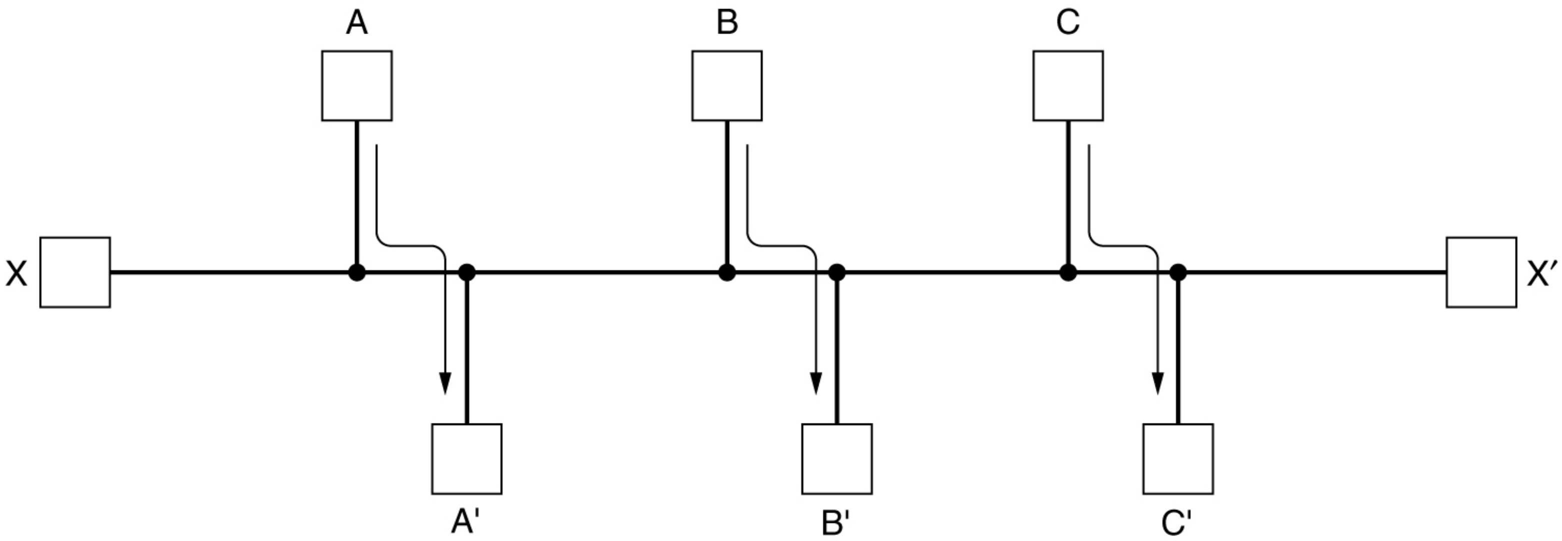
Robustness: Once a major network becomes operative, it may be expected to run continuously for years without any failures. The algorithms designed for routing should be robust enough to handle hardware and software failures and should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted and the network rebooted every time some router goes down.

Stability: The routing algorithms should be stable under all possible circumstances.

Fairness: Every node connected to the network should get a fair chance of transmitting their packets. This is generally done on a first come first serve basis.

Optimality: The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays. Here there is a trade-off and one has to choose depending on his suitability.

Routing Algorithms (2)



A – A’, B – B’, C – C’, can fill the channel, then X-X’ doesn’t get a chance

Conflict between fairness and optimality.

Minimizing the mean packet delay is an obvious candidate to send traffic through the network effectively

Types of Routing Algorithms

Routing Algorithms can be grouped into two major classes:

1. Non-Adaptive(Static)
2. Adaptive(Dynamic)

Non-Adaptive: They don’t base their routing decisions on measurements or estimates of the current traffic and topology. Instead the choice of the route is computed in advance and downloaded to the routers when the network is booted.

Adaptive: They change their routing decisions to reflect changes in the topology and usually traffic as well

The Optimality Principle

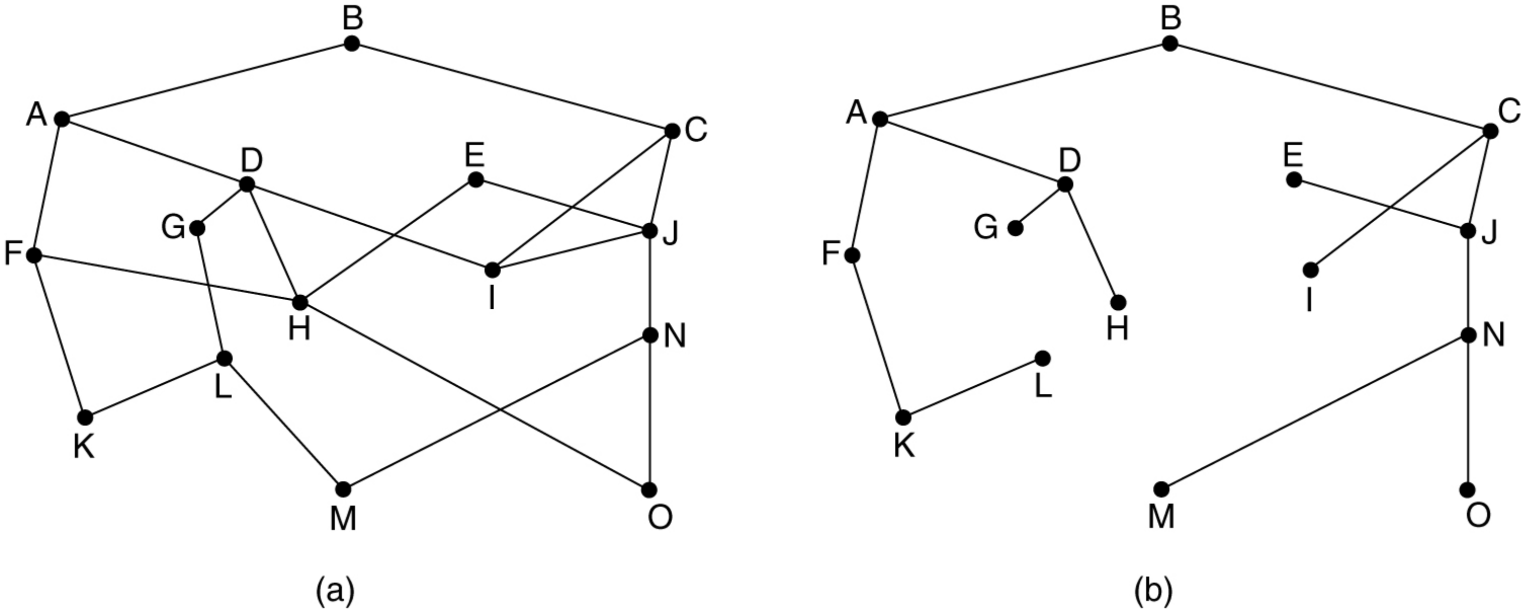
Optimality principle and sink trees

Without regard to topology we can say:If a router J is on the optimal path from router I to router K, then the optimal path from J to K also follows the same route.

Proof: if there was a better way from J to K, then you could use that with the path from I to J for a better path from I to K, so your starting point (the path from I to K was optimal) is contradicted.

If you apply the optimality principle then you can form a tree by taking the optimal path from every other router to a single router, B. The tree is rooted at B. Since it is a tree you don't have loops, so you know that each frame will be delivered in a finite number of hops. Of course finding the set of optimal trees is a lot harder in practice than in theory, but it still provides a goal for all real routing algorithms.

The Optimality Principle



(a) A subnet. (b) A sink tree for router B.

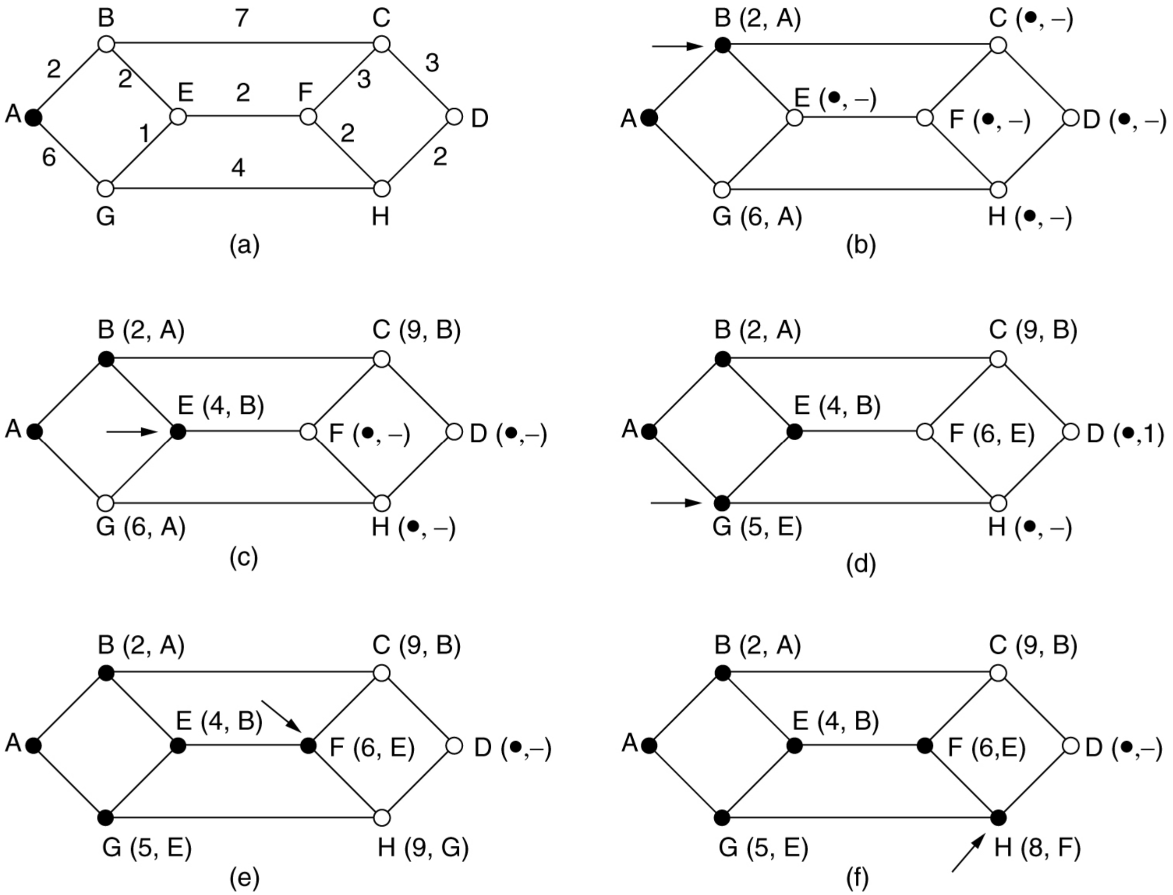
Shortest Path Routing

- Here the idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line.
- In this algorithm, one way of measuring path length is no. of hops, using this paths ABC & ABE are equally long.
- Another metric is geographic distance in kms. ABC is longer than ABE.
- Other different metrics are also possible. For example, each arc could be labeled with the mean queuing & transmission delay for some standard test packets.
- Labels on arc can be computed as a function of distance, Bandwidth, average traffic, communication cost, mean queue length, measured delays and others.

Shortest Path Routing

- By Dijkstra, each node is labeled with its distance from source node along the best known paths.
- Initially all nodes are labeled with infinity.
- A label may be either tentative or permanent.
- Initially all are tentative.
- When it is discovered that label represents a shortest possible path from source to that node then it is made permanent & never changed.

Shortest Path Routing



The first 5 steps used in computing the shortest path from A to D.
The arrows indicate the working node.

Dijkstra’s Algorithm to Compute The Shortest Path Through a Graph

Step 1: Plot the subnet, assign weights to each of the edges between nodes.

Step 2: Using the metrics as distance in km calculate the shortest path from the given source to destination.

Step 3: Initially mark all the paths from source as infinity.

Step4: Starting with the source node check the adjacent nodes for shortest path, and mark them as tentative nodes.

Step 5: From this tentative nodes, select one which is having short distance from source, and mark as permanent.

Step 6: Distance from source to that tentative node should be recorded.

Step 7: Now this node is considered as source node and repeat the steps from 3 to 6.

Step 8: Repeat the steps along the path with the distances being added throughout the path to reach the destination.

Flooding

- ➢ Flooding is a static algorithm , in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- ➢ It generates vast number of duplicate packets, an infinite number unless some measures are taken.
- One such measure is to have a hop counter contained in the header of each packet.
- Second is to keep track of which packets have been flooded, to avoid sending them out a second time.
- Selective Flooding: Instead of sending every incoming packet on every line, the packet is sent only on those lines that go in the right direction.
- Applications:

  In military application, Distributed database, in wireless networks,
- ➢ It can also be used as a metric against which other routing algorithms can be compared.

Distance Vector Routing

- Modern computer networks use dynamic routing algorithms.
- DVR is a dynamic routing algorithm
- This algorithm operates by having each router maintain a table(i.e, a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors.
- This algorithm is some times called by other names, mostly the distributed **Bellman-Ford** routing algorithm and **Ford-Fulkerson** algorithms.
- In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination.

Distance Vector Routing

- The metrics can be number of hops, time delay in msecs, total no.of packets queued along the path, or something similar.
- If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.
- Assume that the metric is delay and that each router knows the delay to each of its neighbors.
- Once every T msec each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor.
- Imagine that one of these tables has just come in from neighbor X, with X_i being X’s estimate of how long it takes to get to router i.

Distance Vector Routing Algorithm：

Step 1: Plot the subnet showing the delay between the nodes.

Step 2: Construct the routing table for each node consisting of delay to reach other nodes in the subnet and the line to be used.

Step 3: Each routers will calculate the delay to reach its adjacent nodes. And this will be recorded in its routing table.

Step 4: Routing table will be exchanged among the routers for every T seconds.

Step 5: Each router will wait until it receives an updated table from its neighbors. For example, router A receives routing table from its neighbor X, with Xi as the delay to reach I from X.

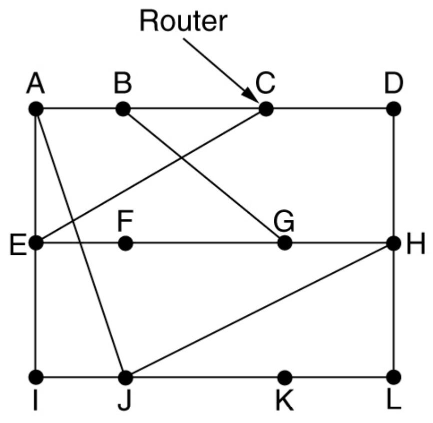
Step 6: With this information, A calculate the new routing table with a delay of Xi+m to reach router I. Where m is a required time for A to reach X.

Step 7: Router A will now can reach router I via X.

Step 8: This steps will be repeated for every source to every other destination.

Step 9: Display routing table of each router.

Distance Vector Routing



(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	

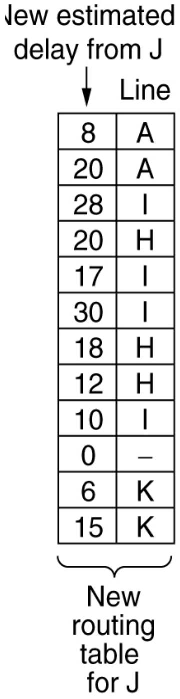
Vectors received from J's four neighbors

(b)

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

Distance Vector Routing

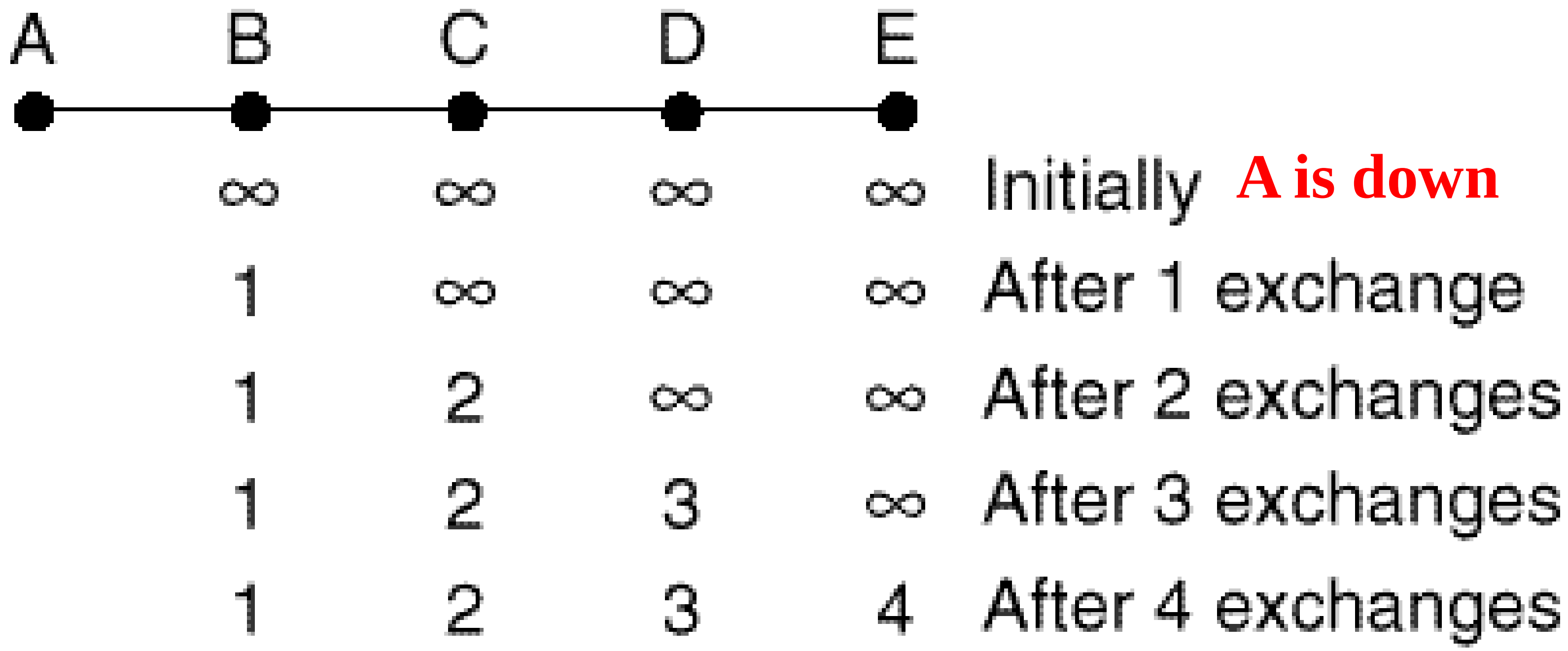
- If the router knows that the delay to X is *m* msec, it also knows that it can reach router i via X in X_i +m msec.
- By performing this calculation for each neighbor, a router can find out which estimates seems the best and use that estimate and corresponding line in its new routing table.
- Consider how J computes its new route to router G.
- JA= 8 msec, AG=18 msec, therefore J to G is 8+18=26 msec via A.
- Similarly J to G via I, H and K as 41(31+10), 18(6+12), and 37(31+6) msec, respectively.
- The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18 msec and that the route to use is via H.
- The same calculations are done for all other destinations and a new routing table is constructed



New estimated delay from J	Line
8	A
20	A
28	I
20	H
17	I
30	I
18	H
12	H
10	I
0	–
6	K
15	K

New routing table for J

Distance Vector Routing: **The count-to-infinity problem**



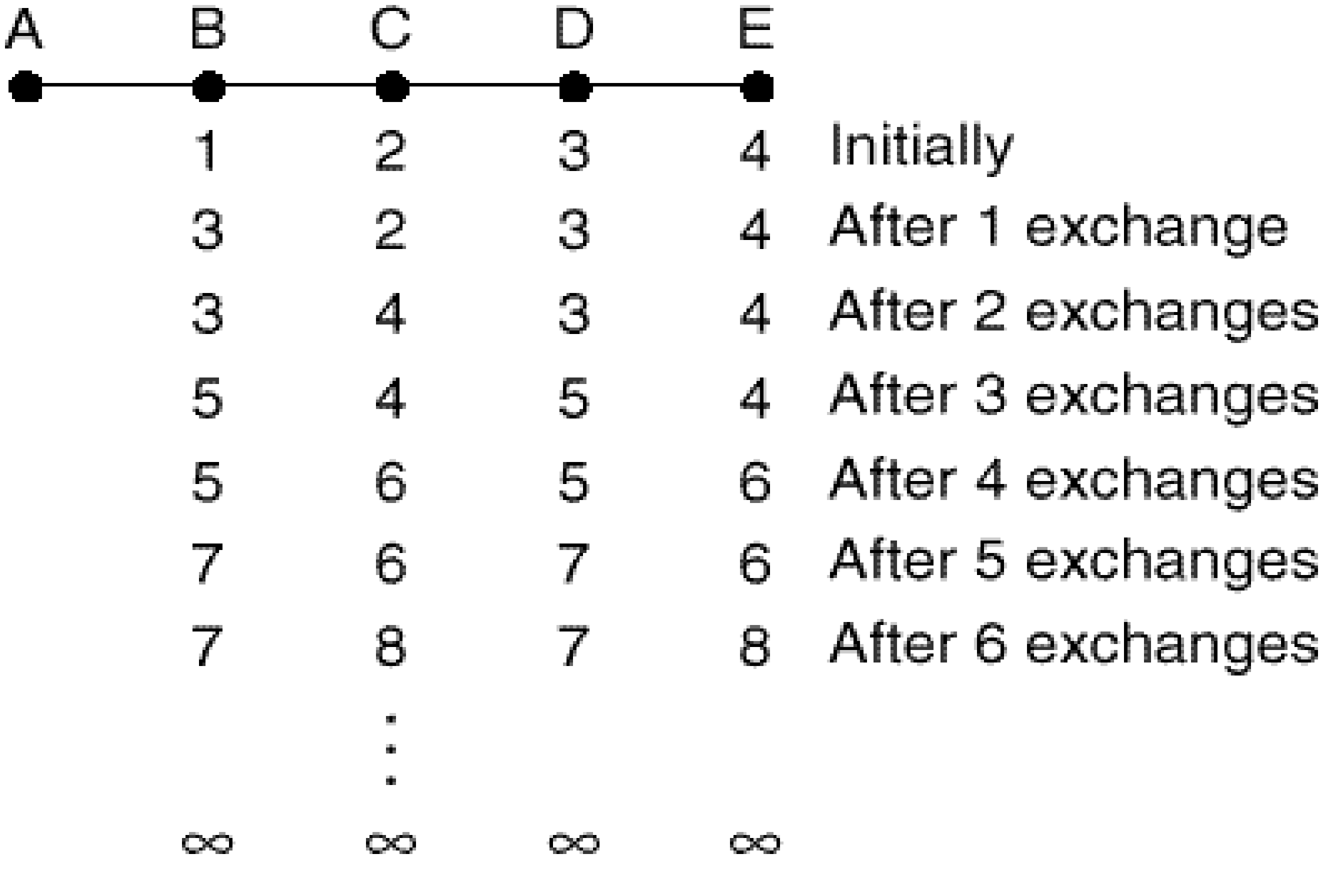
A	B	C	D	E	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

A is down

- Then A comes up. The good news spreads quickly.

The count-to-infinity problem

- Then A comes down. The bad news travels slowly.



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
		⋮			
	∞	∞	∞	∞	

The count-to-infinity problem

- It should be clear why bad news travels slowly: no router ever has a value more than one higher than the minimum of all its neighbors
- Gradually, all the routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity.
- For this reason, it is wise to set infinity to the longest path plus 1 (if using hop count as metric).
- If the metric is time delay, there is no well-defined upper bound, so a high value is needed to prevent a path with a long delay from being treated as down

Partial Solutions:

- Make infinity small

  -use for example16 to represent infinity

- Split Horizon

  -Don’t send routes learnt from a neighbor back to it.

- Split Horizon with poison reverse

  -Send routes learnt from a neighbor back to it but with infinite cost

Link State Routing

- Distance vector routing was used in the ARPANET until 1979, then it was replaced by link state routing.

Two primary reasons caused its demise

- First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes
- Second, the algorithm often took too long to converge(the count-to-infinity problem).
- For these reasons, it was replaced by an entirely new algorithm now called link state routing.
- It is also available in two variants

  1.IS-IS(Intermediate System-Intermediate System)

  2.OSPF(Open Shortest Path First)

Link State Routing

Each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

Link State Routing

- Distance vector routing differs significantly from the link state routing.
- With link state algorithms, routers share only the identity of their neighbors, but they flood this information through the entire network. Distance vector algorithms adopt an opposite approach. Routers periodically share knowledge of the entire network, but only with their neighbors
- Link state routing requires more memory and computation

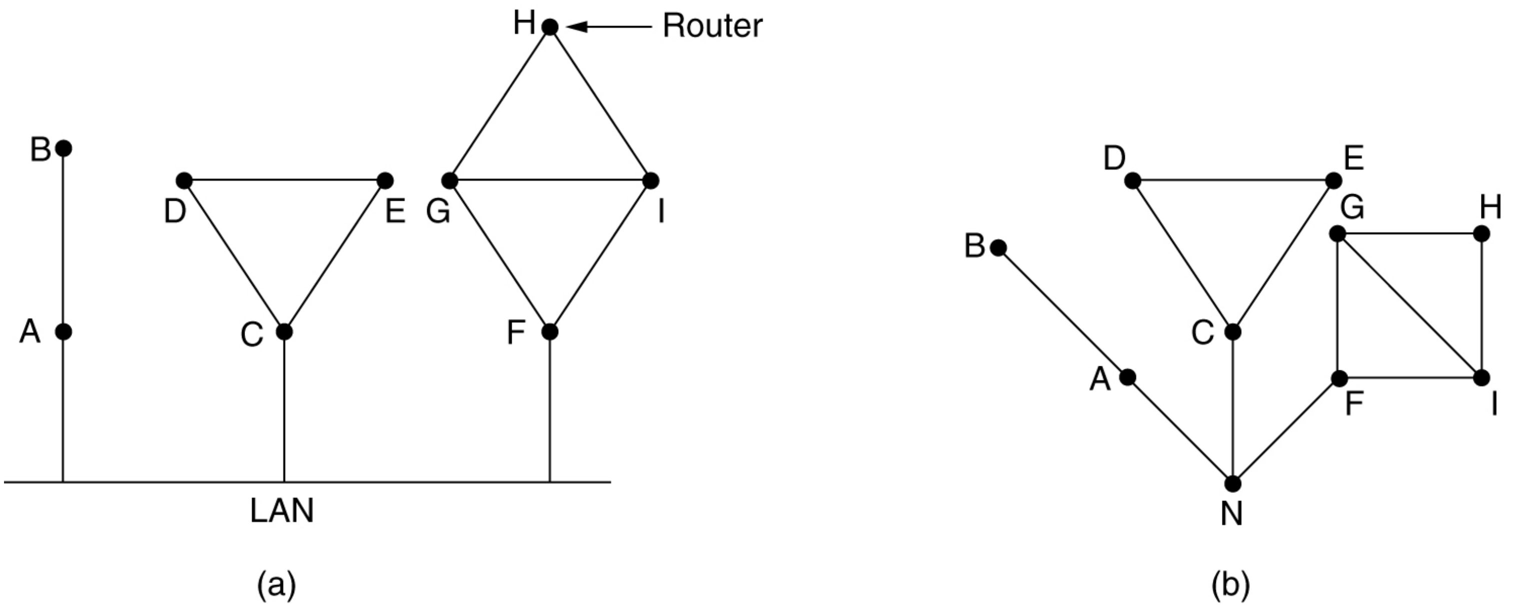
Link State Routing

Learning about the Neighbors

- When a router is booted, its first task is to learn who its neighbor are. It accomplishes this goal be sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is
- When two or more routers are connected by a LAN, the situation is slighted more complicated. One way to model the LAN is to consider it as a node itself

Learning about the Neighbors

……………..By sending HELLO packets



(a) Nine routers and a LAN. (b) A graph model of (a).

Link State Routing

Measuring Line Cost

- The link state routing algorithm requires each router to know, or at least have a reasonable estimate, of the delay to each of its neighbors.
- The most direct way to determine this delay is to send a special ECHO packet over the line that the other side is required to send back immediately.
- By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay

Link State Routing

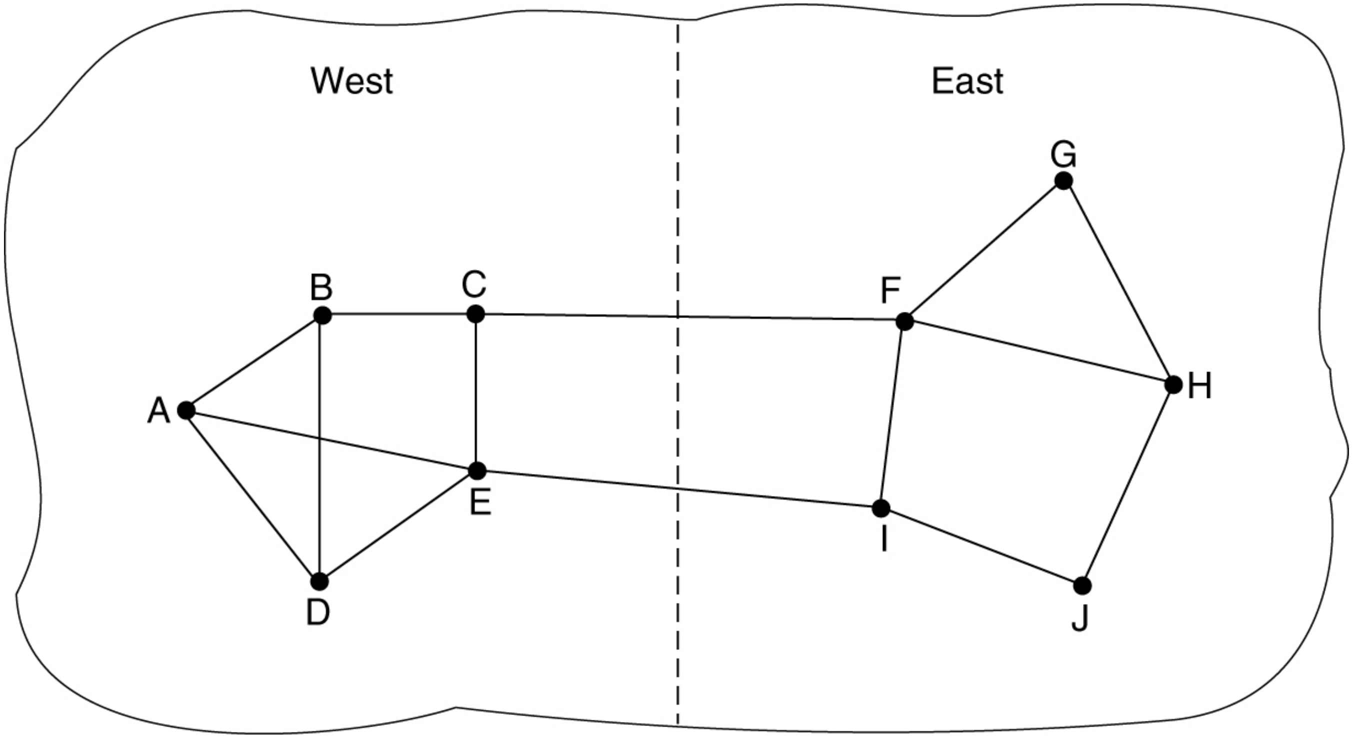
Measuring Line Cost

- An interesting issue is whether or not to take the load into account when measuring the delay.

- To factor the load in, the round-trip timer must be started when the ECHO packet is queued.

- To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.

Measuring Line Cost



A subnet in which the East and West parts are connected by two lines.

- ECHO packet
- RTT/2
- An important issue is whether to take load into account when measuring the delay.
- To factor in ,RTT must be started when ECHO packet is queued.
- To ignore the load, the timer should be started when the ECHO packet reaches the front of queue.
- Two arguments:

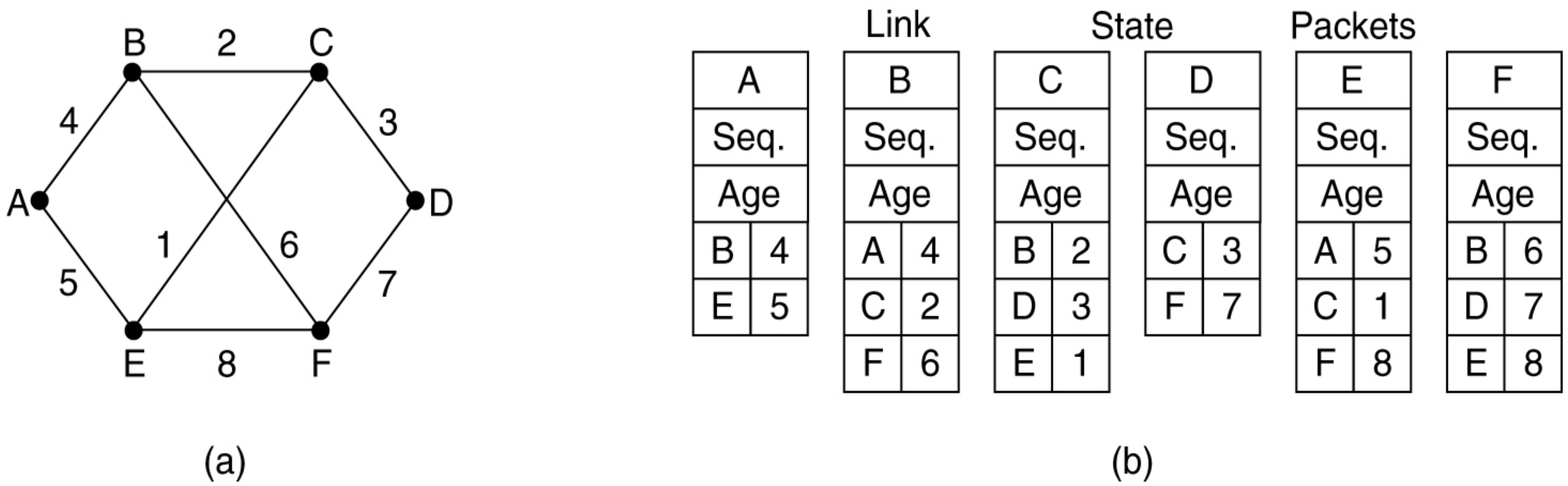
  Including traffic-induced delays in the measurement.

  Including load in delay calculation.

The best solution is to distribute the load over multiple lines, with some known fraction going over each line.

Building Link State Packets

- Building the link state packets is easy. The hard part is determining when to build them. 1. Periodically 2. When some significant event occurs, such as a line or neighbor going down or coming back up ,changes in its properties.



Link State Packets

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

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

Link State Routing

Distributing the Link State Packets

- The trickiest part of the algorithm is distributing the link state packets reliably. As the packets are distributed and installed, the routers getting the first ones will change their routes.
- Consequently, the different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machines, and other problems.
- The fundamental idea is to use flooding to distribute the link state packets

Link State Routing

Distributing the Link State Packets

- To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see.

- When a new link state packet comes in, it is checked against the list of packets already seen.

  1. If new: forward on all lines except the one it arrived on
  2. If duplicate or old packet: discard

Link State Routing

Distributing the Link State Packets & its problems

- If the sequence numbers wrap around, confusion will reign. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around.
- If a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet will be rejected as a duplicate.
- If a sequence number is ever corrupted and 65540 is received instead of 4 (a 1-bit error), packets 5 through 65540 will be rejected as obsolete, since the current sequence number is thought to be 65,540.

Link State Routing

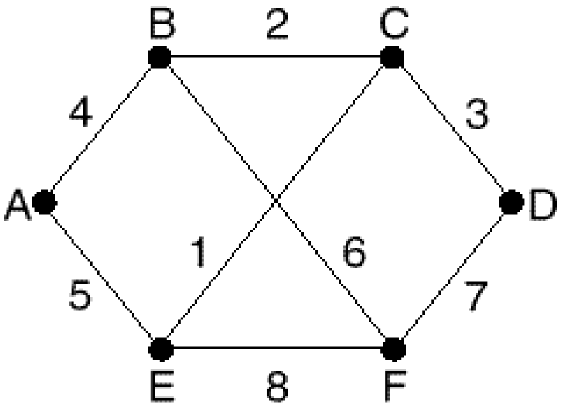
Distributing the Link State Packets

- The solution to all these problems is to include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.
- The age field is also decremented by each router during the initial flooding process, to make sure no packet can get lost and live for an indefinite period of time.

Refinement:

- When a link State packet comes in to a router for flooding, it is not queued for transmission immediately. Instead it is first put in holding area. If another packet from same source comes in before the first packet is transmitted, their sequence numbers are compared.

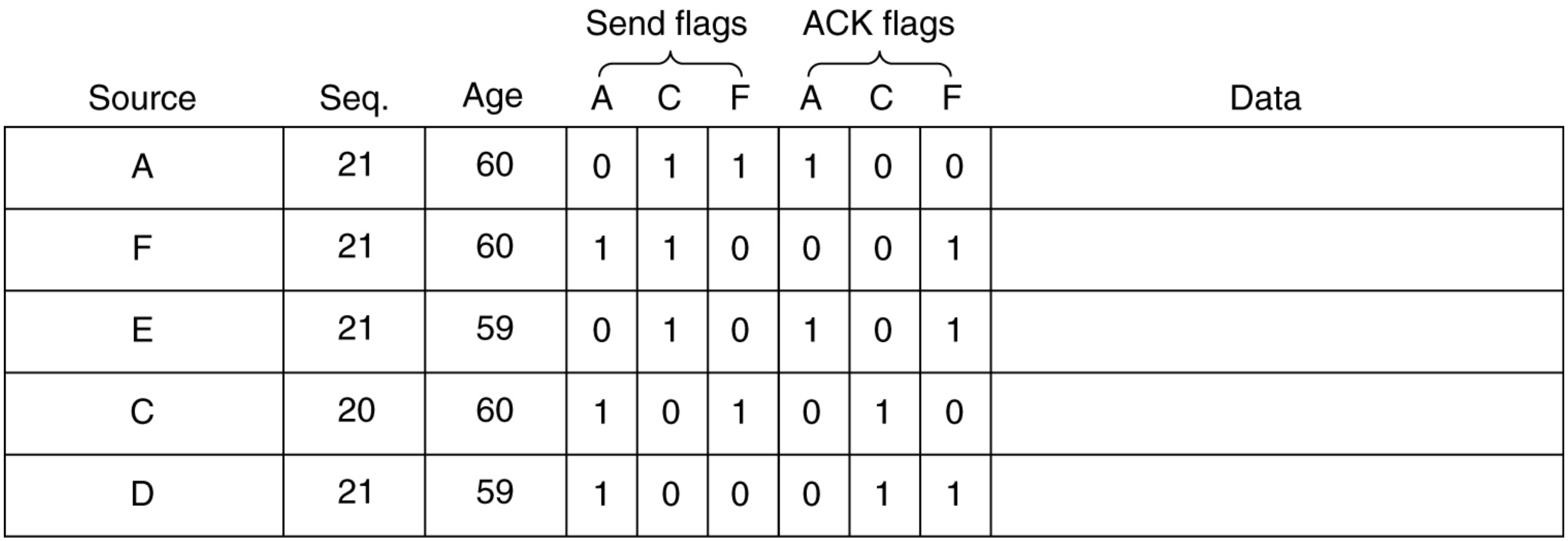
Link State Routing



Distributing the Link State Packets

- To guard against errors on the router-router lines, all link state packets are acknowledged.

Packet buffer for router B



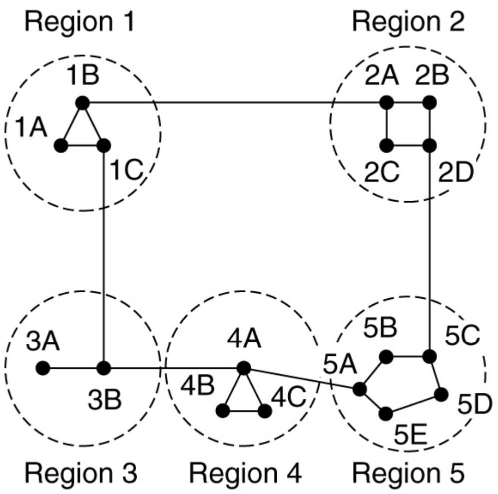
			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Link State Routing

Computing the New Routes

- Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Now Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations.
- The OSPF (Open Shortest Path First) protocol uses link state routing algorithm
- Another link state is IS-IS, was used in connectionless network layer protocol

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Hierarchical routing.

Hierarchical Routing

- Unfortunately, the gain in routing table space are not free. There is a penalty to be paid, and this penalty is in the form of increased path length.

- For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region.
- Consider a subnet with 720 routers. If there is no hierarchy, each router needs 720 routing table entries.

Hierarchical Routing

- If the subnet is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries.
- If a three level hierarchy is chosen, with eight clusters, each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries.
- Kamoun and kleinrock discovered that the optimal number of levels for an N router subnet is ln N, requiring a total of e ln N entries per router.

Broadcast Routing

- One broadcasting method that requires no special features from the subnet is for the source to simply send a distinct packet to each destination.

- Waste bandwidth and require the source to have a complete list of all destinations.

- Flooding is another obvious candidate. But it generates too many packets and consumes too much bandwidth

Broadcast Routing: Multidestination routing

- Each packet contains either a list of destinations or a bit map indicating the desired destinations. When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed.

- The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line. In effect, the destination set is partitioned among the output lines

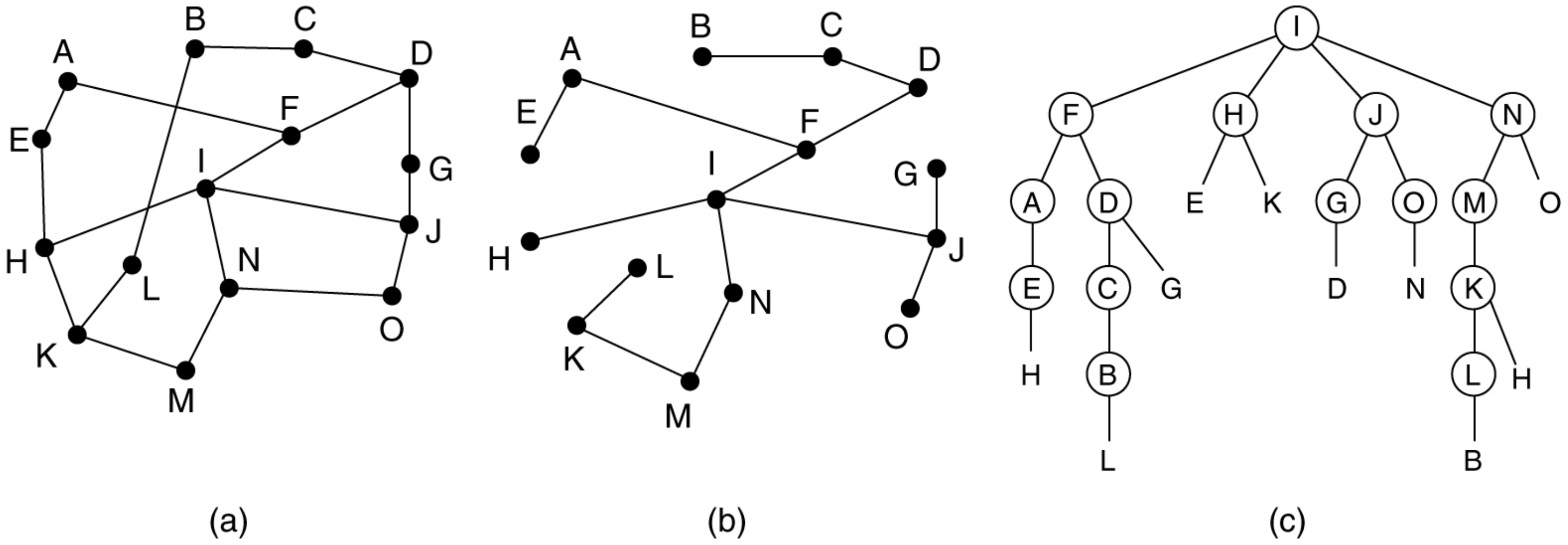
Broadcast Routing

- A fourth broadcast algorithm makes explicit use of the sink tree for the router initiating the broadcast, or any other convenient spanning tree for that matter
- This method makes excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job. The only problem is that each router must have knowledge of some spanning tree for it to be applicable

If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router. This being the case, the router forwards copies of it onto all lines except the one it arrived on. If, however, the broadcast packet arrived on a line other than the preferred one for reaching the source the packet is discarded as a likely duplicate

Broadcast Routing: Reverse Path Forwarding

- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast. If so, forward it.



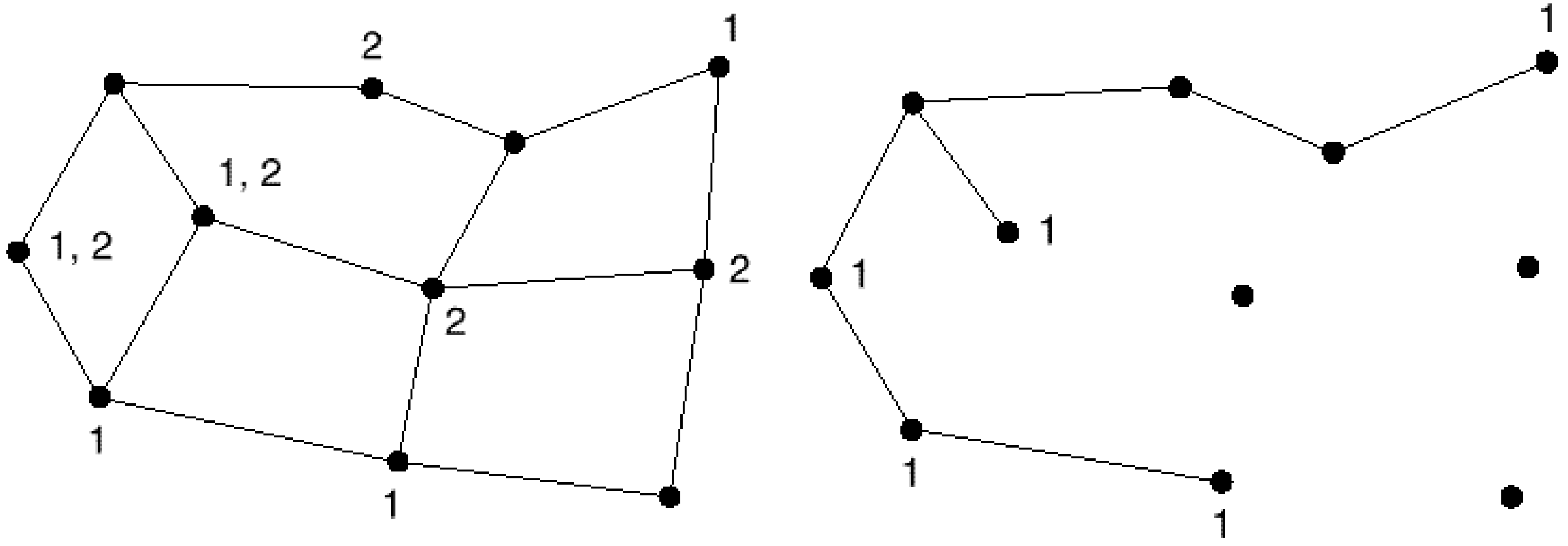
Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

Multicast Routing

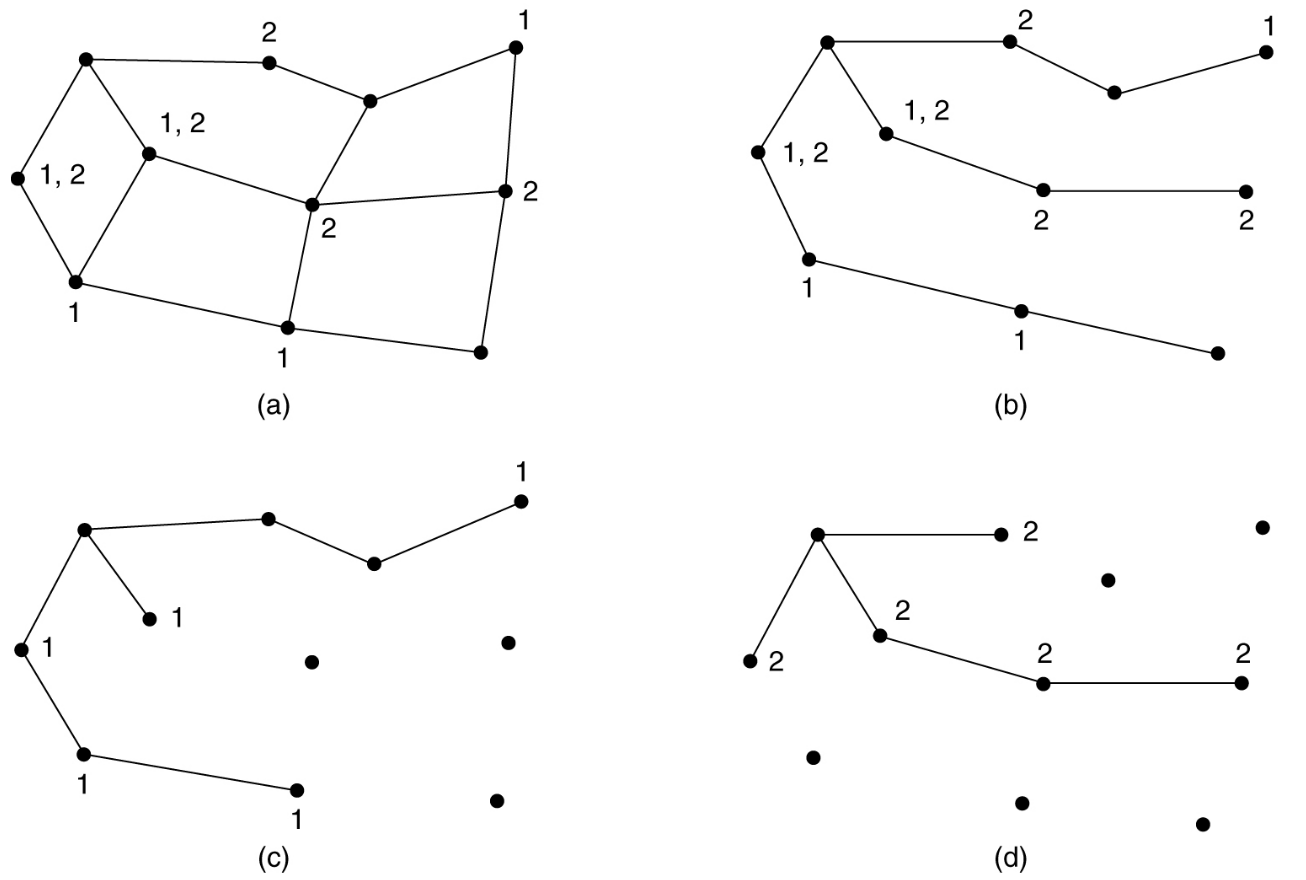
- To do multicasting, group management is required. Some way is needed to create and destroy groups, and for processes to join and leaves groups. It is important that **routers know which of their hosts belong to which groups**.
- Either hosts must inform their routers about changes in group membership, or routers must query their hosts periodically. Either way, routers learn about which of their hosts are in which groups. **Routers tell their neighbors**, so the information propagates through the subnet

Multicast Routing

To do multicast routing, each router computes a spanning tree covering all other routers in the subnet



Multicast Routing



(a) A network. (b) A spanning tree for the leftmost router.
(c) A multicast tree for group 1. (d) A multicast tree for group 2.

Multicast Routing

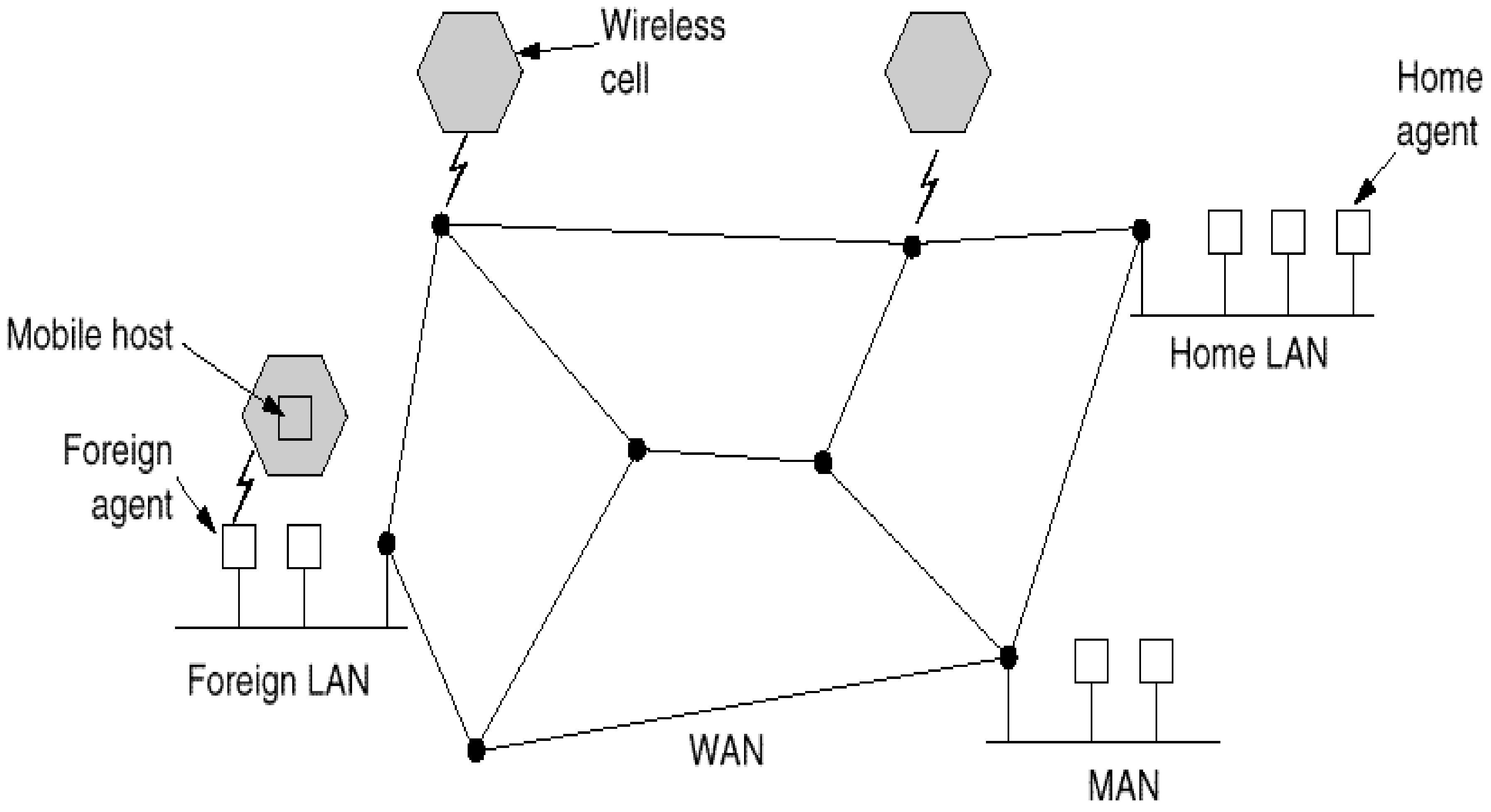
- Various ways of pruning the spanning tree are possible. The simplest one can be used if **link state routing** is used, and each router is aware of the complete subnet topology, including which hosts belong to which groups.

- Then the spanning tree can be pruned by **starting at the end of each path and working toward the root**, removing all routers that do not belong to the group in question

Multicast Routing

- With distance vector routing, a different strategy is used. The basic algorithm is Distance vector algorithm whenever a router with no hosts interested in a particular group and no connections to other routers receives a multicast message for that group, it responses with a PRUNE message, telling the sender not to send it any more multicasts for that group
- source-specific multicast trees: scales poorly to large networks n groups, m members: a total of nm trees
- core-based tree approach: each group has only one multicast tree n groups: n trees

Routing for Mobile Hosts



Routing for Mobile Hosts

When a new user enters an area, either by connecting to it, or just wandering into the cell, his computer must register itself with the **foreign agent** there. The registration procedure typically works like this:

1. Periodically, each foreign agent broadcasts a packet announcing its existence and address. A newly arrived mobile host may wait for one of these messages, but if none arrives quickly enough, the mobile host can broadcast a packet saying: “Are there any foreign agents around?”

Routing for Mobile Hosts

2. The mobile host registers with the foreign agent, giving its home address, current data link layer address, and some security information.

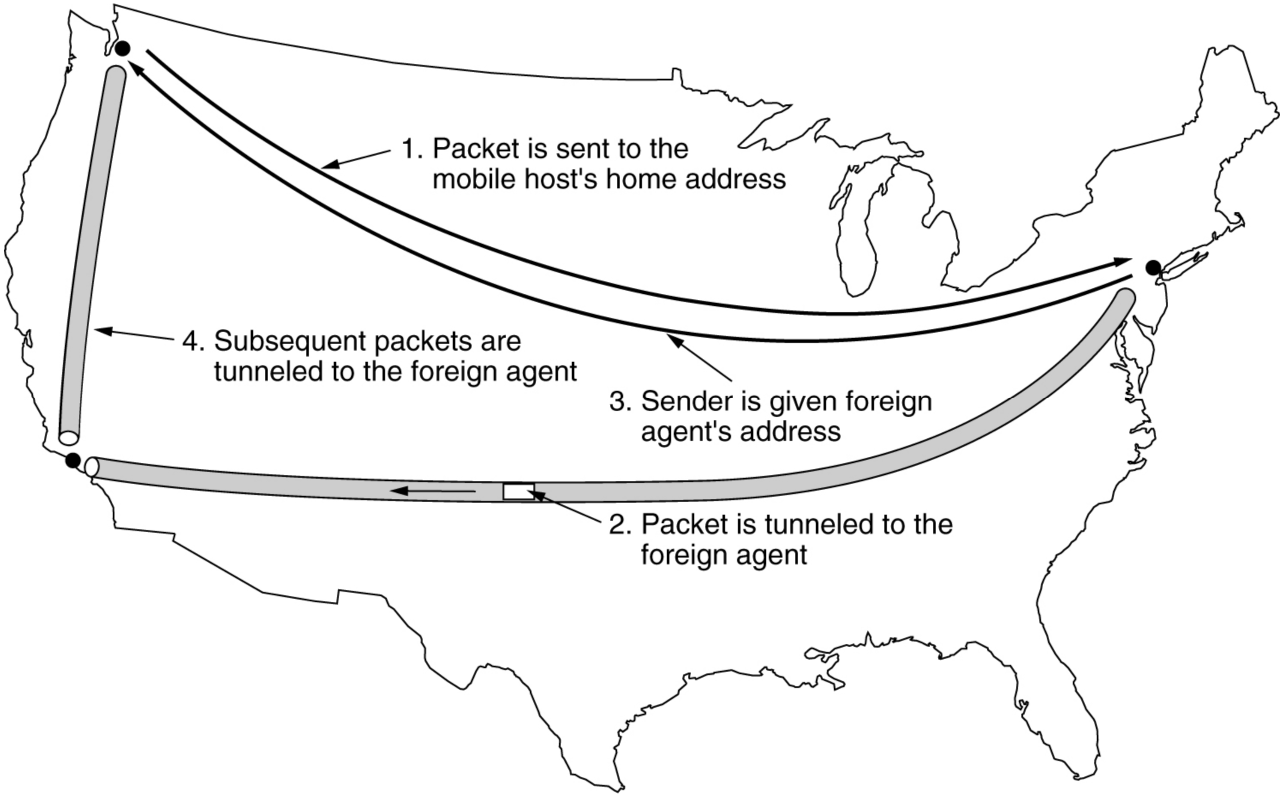
3. The foreign agent contacts the mobile host’s home agent and says: “One of your hosts is over here.” The message from the foreign agent to the home agent contains the foreign agent’s network address. It also includes the security information, to convince the home agent that the mobile host is really there.

Routing for Mobile Hosts

4. The home agent examines the security information, which contains a time stamp, to prove that it was generated within the past few seconds. If it is happy, it tells the foreign agent to proceed.
5. When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.

Ideally, when a user leaves an area, that, too, should be announced to allow deregistration, but many users abruptly turn off their computers when done.

Routing for Mobile Hosts (2)

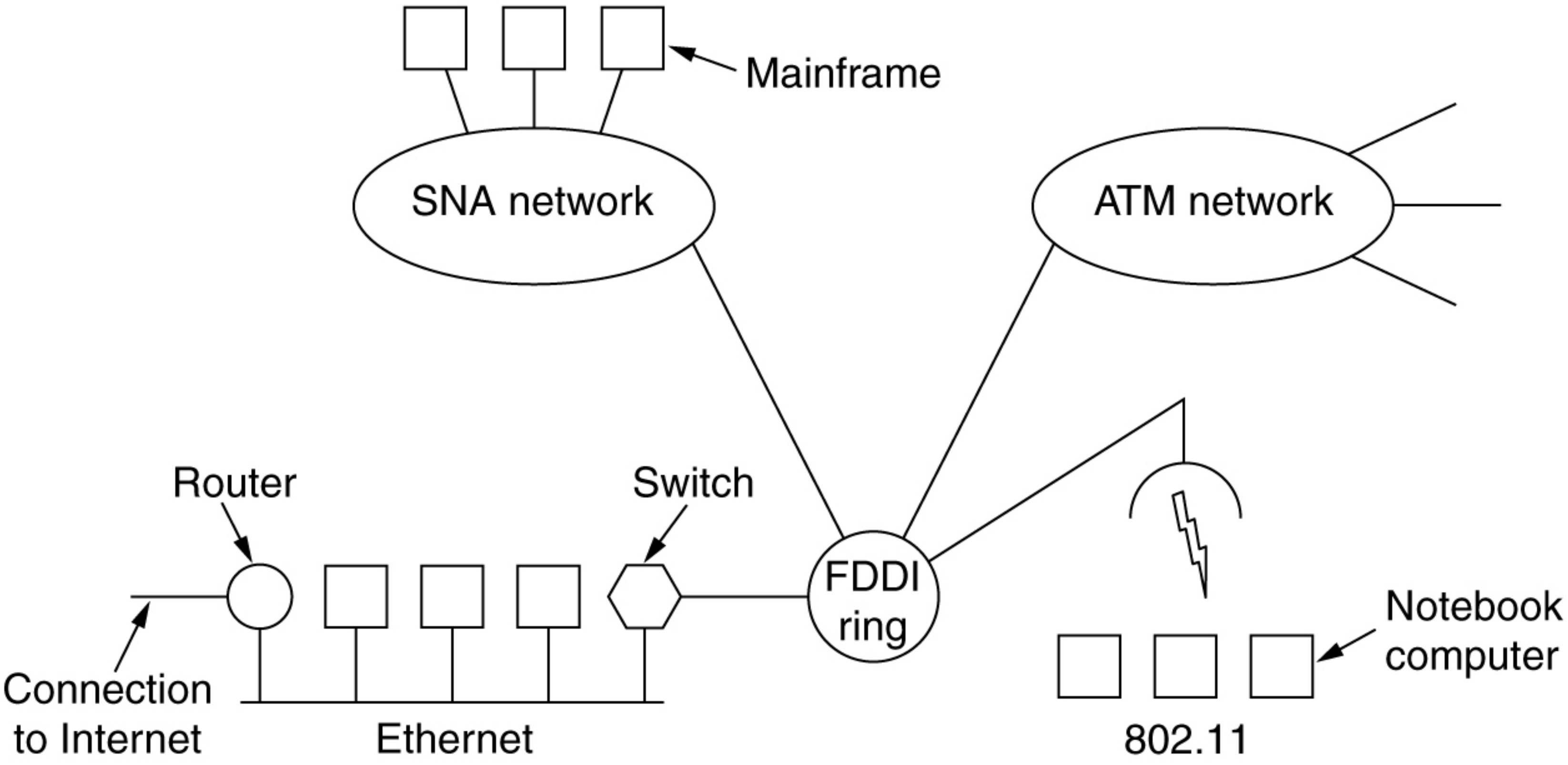

1. Packet is sent to the
mobile host's home address
4. Subsequent packets are
tunneled to the foreign agent
3. Sender is given foreign
agent's address
2. Packet is tunneled to the
foreign agent

Packet routing for mobile users.

Internetworking

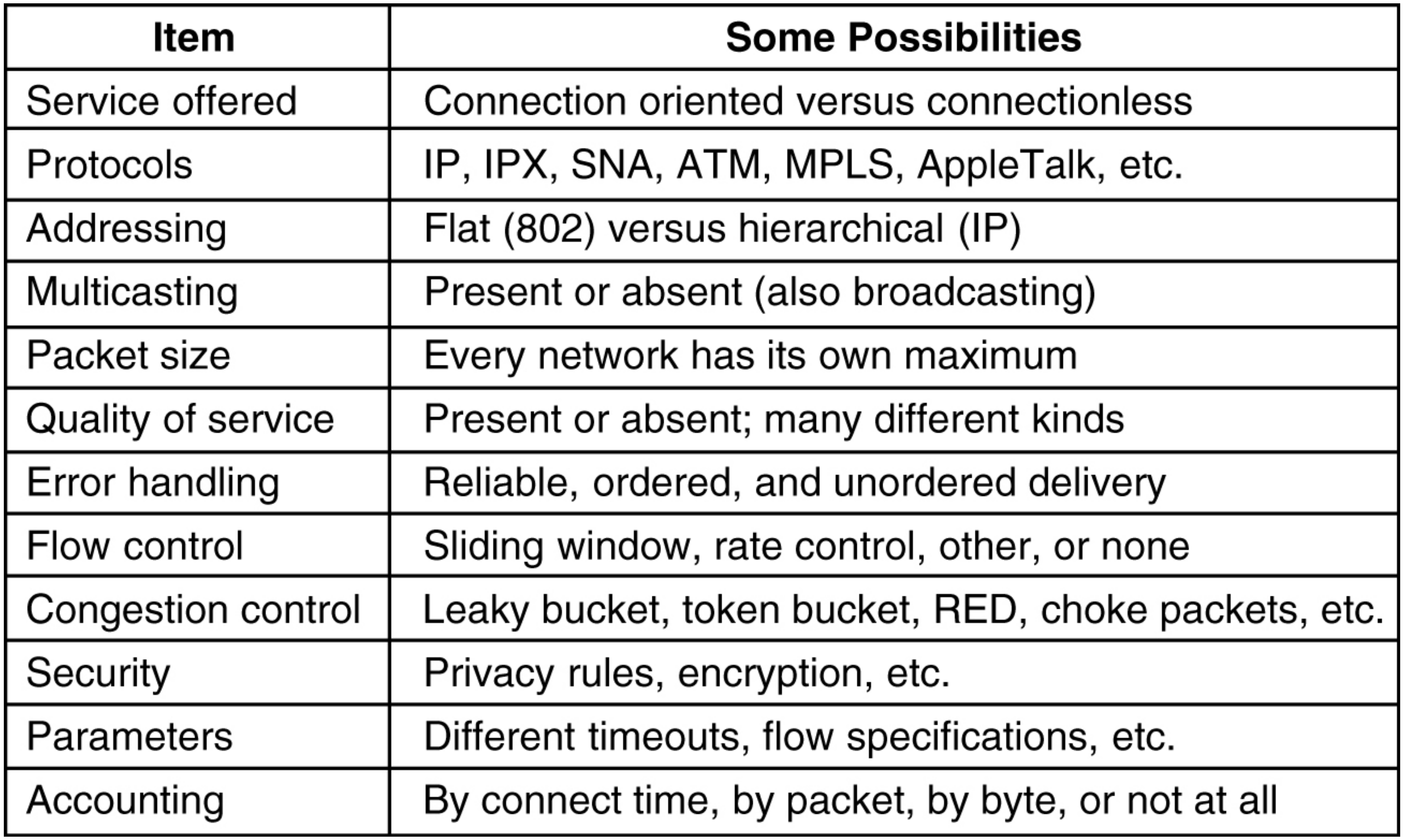
- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

Connecting Networks



A collection of interconnected networks.

How Networks Differ



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

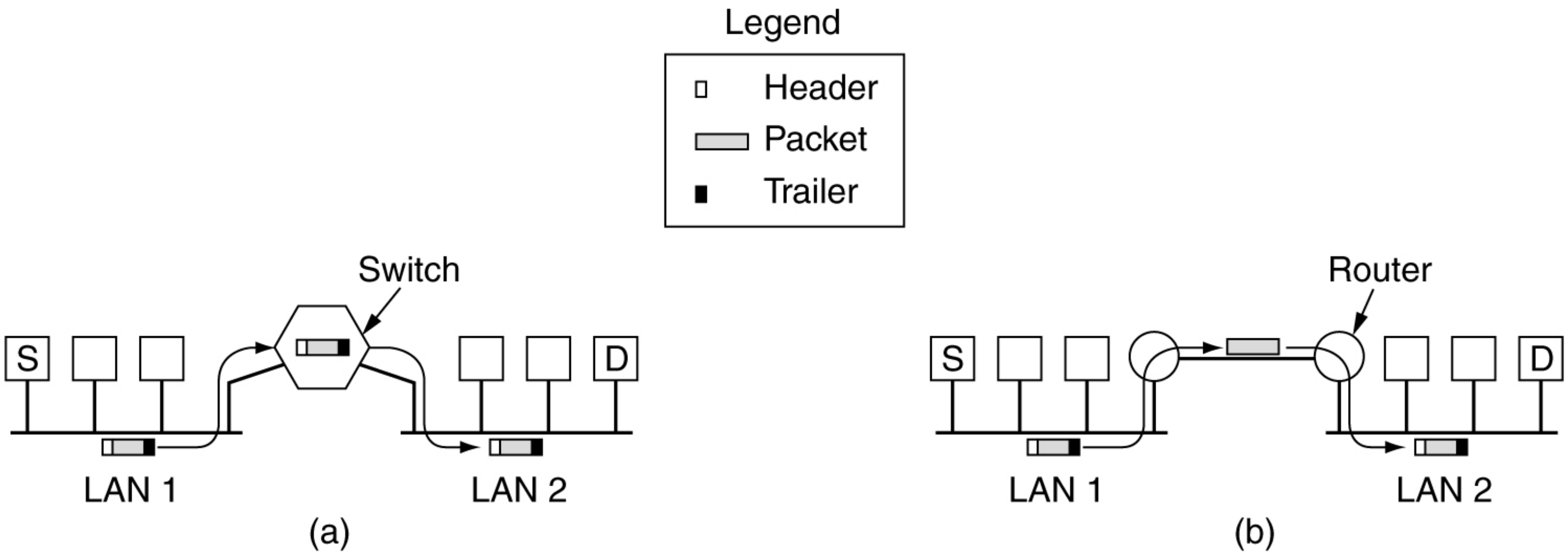
Some of the many ways networks can differ.

How Networks Can Be Connected

Networks can be interconnected by different devices.

- In PL, Repeaters or hubs can be used
- In DLL, Bridges and Switches can be used.(examines MAC address, do minor translations).
- In NL, Routers can be used(if differ they are able to translate between packet formats, which is rare). A router that can handle multiple protocols is called a multiprotocol router.
- In TL & AL gateways do translation.

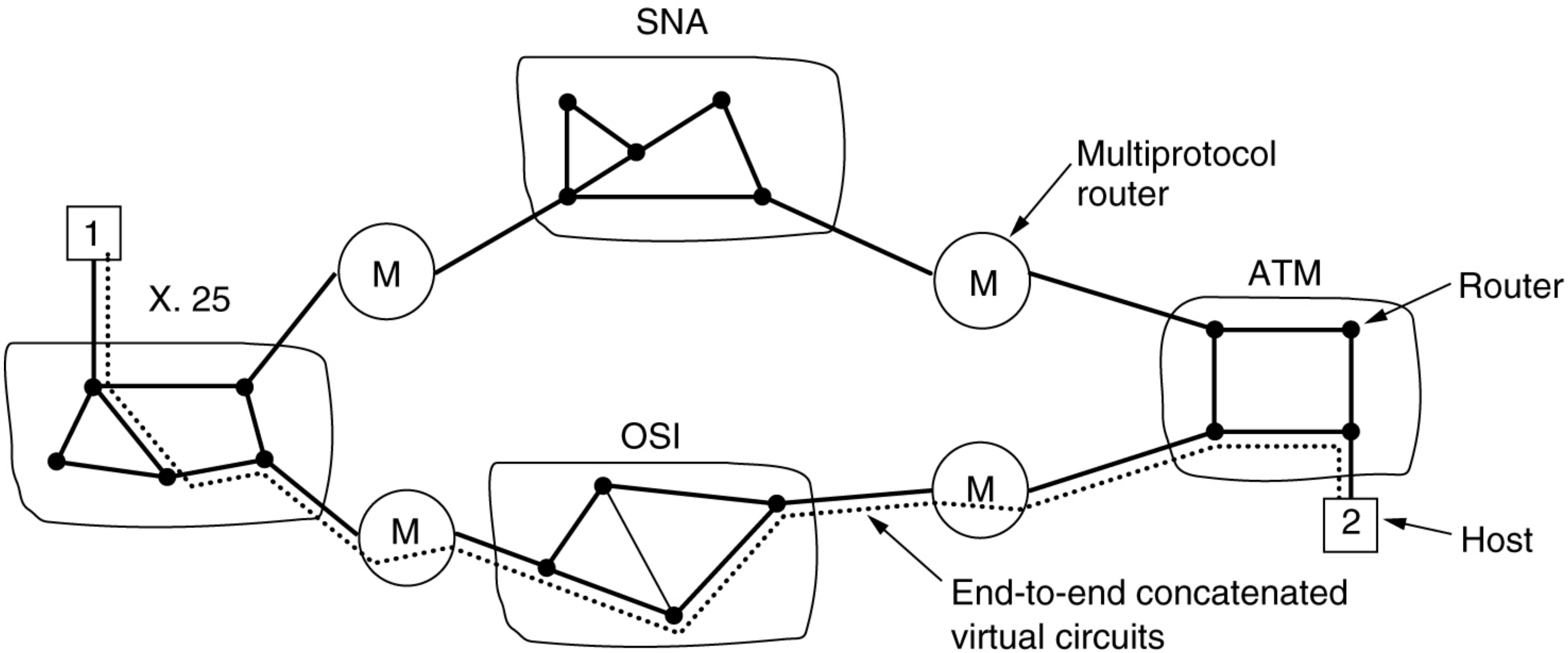
How Networks Can Be Connected



(a) Two Ethernets connected by a switch.

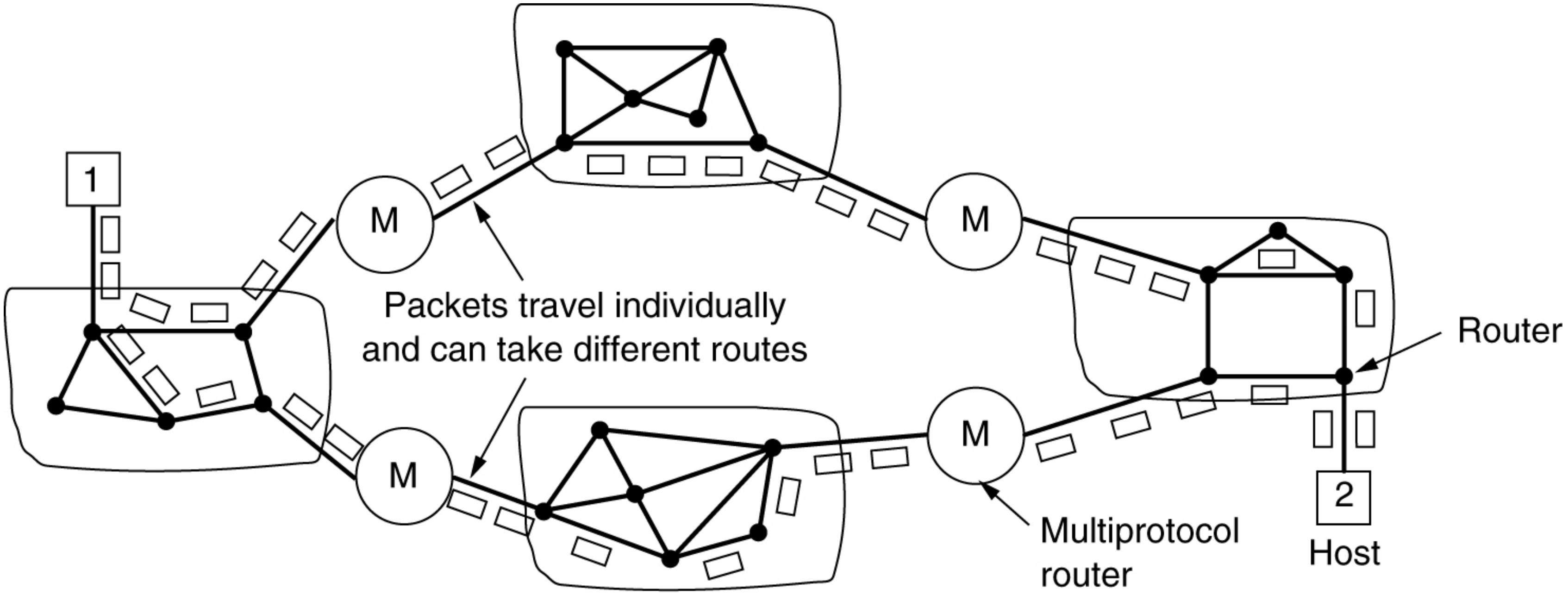
(b) Two Ethernets connected by routers.

Concatenated Virtual Circuits



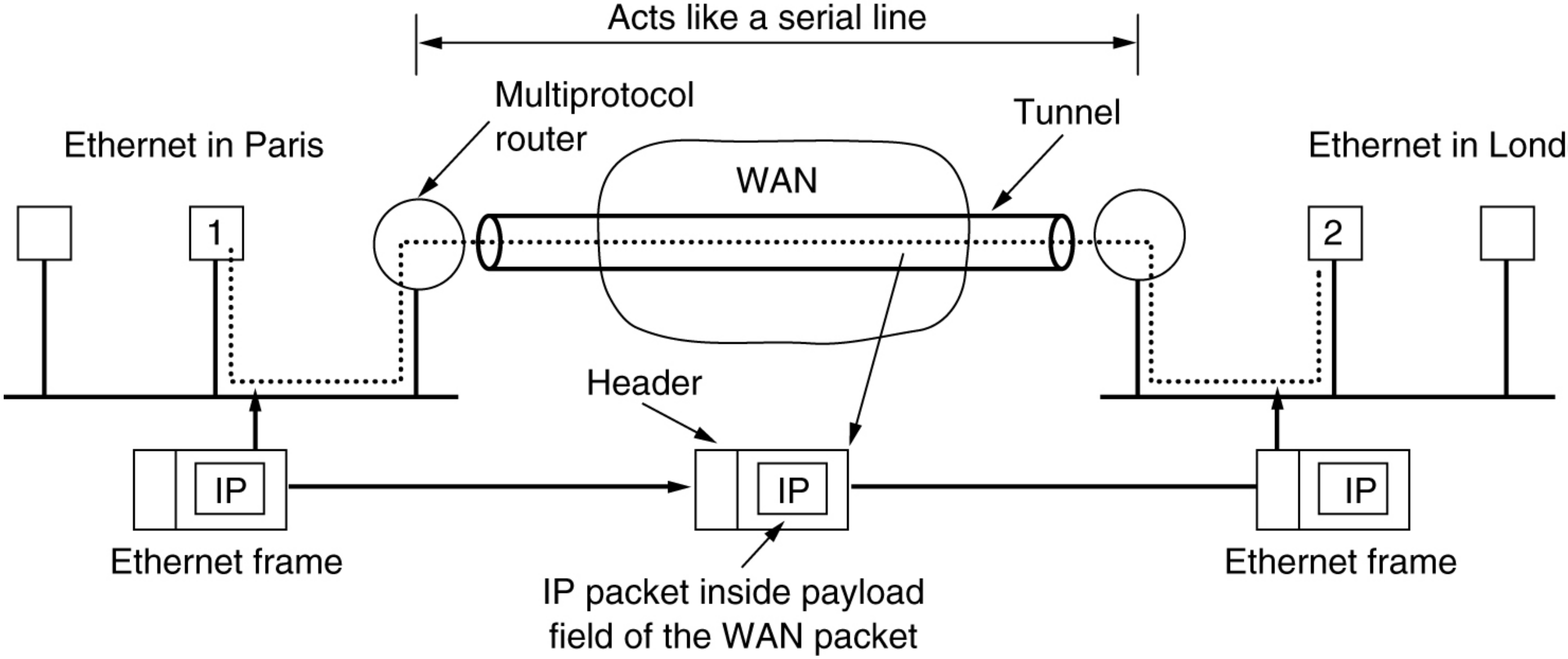
Internetworking using concatenated virtual circuits.

Connectionless Internetworking



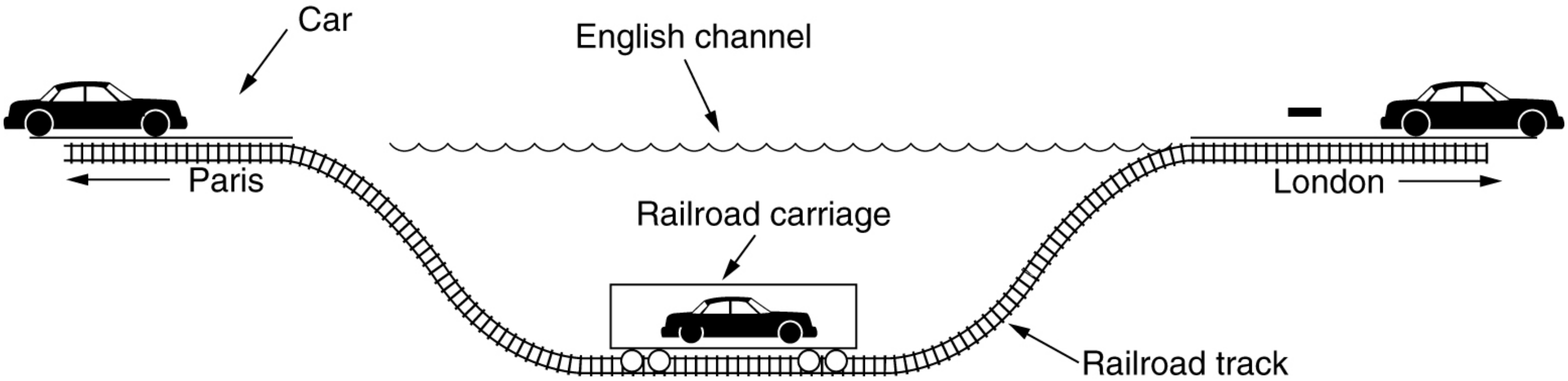
A connectionless internet.

Tunneling



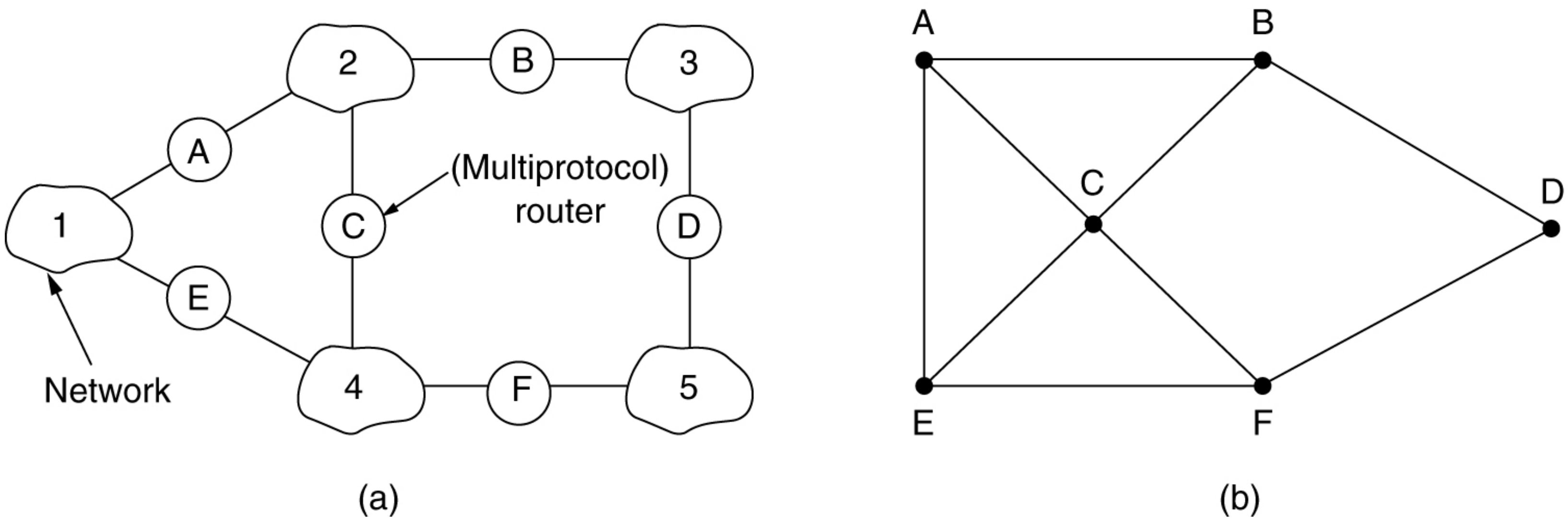
Tunneling a packet from Paris to London.

Tunneling (2)



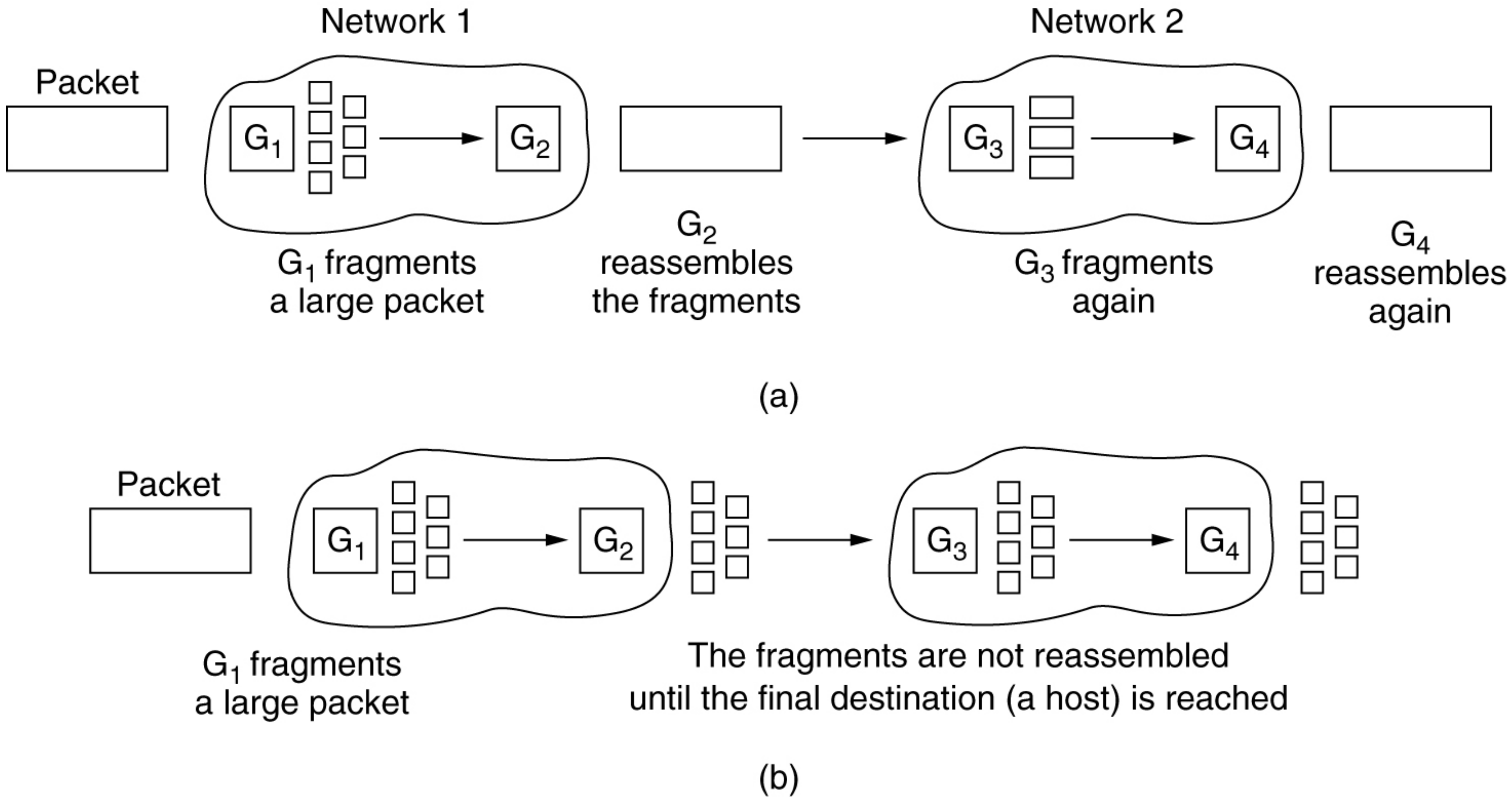
Tunneling a car from France to England.

Internetwork Routing



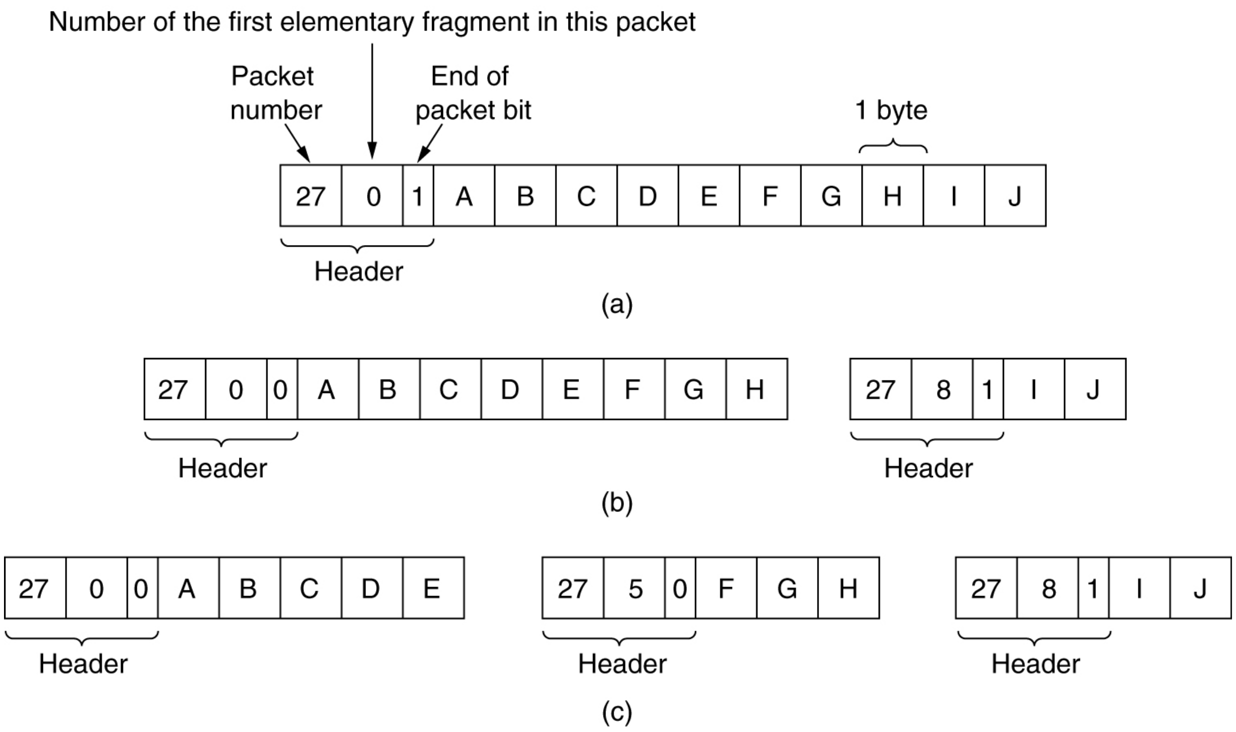
(a) An internetwork. (b) A graph of the internetwork.

Fragmentation


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)

(a) Transparent fragmentation. (b) Nontransparent fragmentation.

Fragmentation (2)


Number of the first elementary fragment in this packet
Packet number
End of packet bit
1 byte
27 | 0 | 1 | A | B | C | D | E | F | G | H | I | J
Header
(a)
27 | 0 | 0 | A | B | C | D | E | F | G | H
Header
27 | 8 | 1 | I | J
Header
(b)
27 | 0 | 0 | A | B | C | D | E
Header
27 | 5 | 0 | F | G | H
Header
27 | 8 | 1 | I | J
Header
(c)

Fragmentation when the elementary data size is 1 byte.

(a) Original packet, containing 10 data bytes.

(b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.

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