Solutions for Homework #6

7.2. Is it possible for a network to offer best-effort connection-oriented service? What features would such a service have, and how does it compare to best-effort connectionless service?

Solution:

Best-effort connection-oriented service would involve the transfer of packets along a pre-established path in a manner that does not provide mechanisms for dealing with the loss, corruption or misdelivery of packets. Best-effort connection-oriented service would require some means for establishing a path prior to the transfer of packets. Best-effort connectionless service would involve the transfer of packets in a datagram fashion, where routing decisions are made independently for each packet.

The path setup requirement makes connection-oriented service more complex than connectionless service. On the other hand, once a path is established, less processing is required to decide how a packet is to be forwarded. Connectionless service is more robust than connection-oriented service since connectionless service readily reroutes packets around a failure while VC service requires that new paths be established.

7.4. Where is complexity concentrated in a connection-oriented network? Where is it concentrated in a connectionless network?

Solution:

The complexity in connection-oriented networks revolves around the need to establish and maintain connections. Each node must implement the signaling required by the connection establishment process; each node must also maintain the state of the node in terms of connections already established and transmission resources available to accommodate new connections. End systems must be capable of exchanging signaling information with the network nodes to initiate and tear down connections. A connection oriented network must also include routing to select the paths for new connections.

Connectionless networks only require that nodes forward packets according to its routing tables. End systems only need to place network address information in the packet headers. The complexity of connectionless networks revolves around routing tables. Routing tables may be static and set up by a network administrator, or they may be dynamic and involve processing and exchange of link state information among nodes.

7.7. Apply the end-to-end argument to the question of how to control the delay jitter that is incurred in traversing a multi-hop network.

Solution:

Delay jitter is the variability in the packet delay experienced by a sequence of packets traversing the network. Many audio and video applications have a maximum tolerable bound on the delay jitter.

To control the delay jitter inside the network, every router must forward the packets within a certain time window by means of delaying the "early" packets or immediately forwarding the "late" packets. This requires significant amount of state information and processing at each router, and hence greatly complicates the network.

In the end-to-end approach, the applications at the end-systems have the knowledge of the tolerable range of jitter. A smoothing buffer can be used at the end system to collect the packets, and pass the packets to the application layer at a constant rate according to the application's specification. Consequently, the jitter is controlled while the network service remains simple.

20. A message of size m bits is to be transmitted over an L-hop path in a store-and-forward packet network as a series of N consecutive packets, each containing k data bits and h header bits. Assume that $m >> k + h$. The transmission rate of each link is R bits/second. Propagation and queueing delays are negligible.

Solutions follow questions:

We know from Figure 7.10 that if the message were sent as a whole, then an entire message propagation delay must be incurred at each hop. If instead we break the message into packets, then from Figure 7.12 a form of pipelining takes place where only a packet time is incurred in each hop. The point of this problem is to determine the optimum packet size that minimizes the overall delay.

We know that:

- message size = m bits = N packets
- one packet = k data bits + h header bits
- $m >> k + h$
- there are L hops of store-and-forward
- transmission rate = R bits/sec
- t_{prop} and t_{Qing} are negligible
- transmission time of one packet, T, equals the amount of bits in a packet / transmission rate, or $T = (k + h) / R$.

(a) What is the total number of bits that must be transmitted?

$N (k + h)$ bits are transmitted.

(b) What is the total delay experienced by the message (that is, the time between the first transmitted bit at the source and the last received bit at the destination)?

The total delay, D, can be expressed by

$$D = Lt_{prop} + LT + (N - 1)T.$$

Since the propagation delay is negligible, the first term in the above equation is zero. Substituting for T we get,

$$D = L\frac{k+h}{R} + (N-1)\frac{k+h}{R}$$

$$= (N-1+L)\frac{k+h}{R}$$

(c) What value of k minimizes the total delay?

We determined D in the previous equation. We also know that $N = \left\lceil \frac{m}{k} \right\rceil$, where m is the ceiling.

Therefore,

$$D \approx \left(\frac{m}{k} - 1 + L\right)\frac{k+h}{R} = \left(\frac{m}{k}\right)\frac{k+h}{R} + (L-1)\frac{k+h}{R}$$

Note that there are two components to the delay: 1. the first depends on the number of packets that need to be transmitted, *m/k*, as *k* increases this term decreases; 2. the second term increases as the size of the packets, *k* + *h*, is increased. The optimum value of *k* achieves a balance between these two tendencies.

$$\frac{dD}{dk} = \left[\frac{d}{dk}\left(\frac{m}{k} - 1 + L\right)\right]\frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{d}{dk}\left(\frac{k+h}{R}\right)$$

$$= \frac{-m}{k^2} \times \frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{1}{R}$$

If we set *dD/dk* to zero, we obtain:

$$\frac{-m}{k^2} \times \frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{1}{R} = 0$$

$$\frac{m}{k^2}\left(\frac{k}{R} + \frac{h}{R}\right) = \frac{m}{kR} - \frac{1}{R} + \frac{L}{R}$$

$$\frac{m}{kR} + \frac{mh}{k^2 R} = \frac{m}{kR} + \frac{L-1}{R}$$

$$\frac{mh}{k^2} = L - 1$$

$$k^2 = \frac{mh}{L-1}$$

$$k = \sqrt{\frac{mh}{L-1}}$$

It is interesting to note that the optimum payload size is the geometric mean of the message size and the header, normalized to the number of hops (-1).

7.21. Suppose that a datagram packet-switching network has a routing algorithm that generates routing tables so that there are two disjoint paths between every source and destination that is attached to the network. Identify the benefits of this arrangement. What problems are introduced with this approach?

Solution:

There are several benefits to having two disjoint paths between every source and destination. The first benefit is that when a path goes down, there is another one ready to take over without the need of finding a new one. This reduces the time required to recover from faults in the network. A second benefit is that traffic from a source to a destination can be load-balanced to yield better delay performance and more efficient use of network resources. On the other hand, the computing and maintaining two paths for every destination increases the complexity of the routing algorithms. It is also necessary that the network topology be designed so that two disjoint paths are available between any two nodes.

7.26. Suppose a routing algorithm identifies paths that are "best" in the sense that: (1) minimum number of hops, (2) minimum delay, or (3) maximum available bandwidth. Identify conditions under which the paths produced by the different criteria are the same? are different?

Solution:

The first criterion ignores the state of each link, but works well in situations were the states of all links are the same. Counting number of hops is also simple and efficient in terms of the number of bits required to represent the link. Minimum hop routing is also efficient in the use of transmission resources, since each packet consumes bandwidth using the minimum number of links.

The minimum delay criterion will lead to paths along the route that has minimum delay. If the delay is independent of the traffic levels, e.g. propagation delay, then the criterion is useful. However, if the delay is strongly dependent on the traffic levels, then rerouting based on the current delays in the links will change the traffic on each link and hence the delays! In this case, not only does the current link delay need to be considered, but also the derivative of the delay with respect to traffic level.

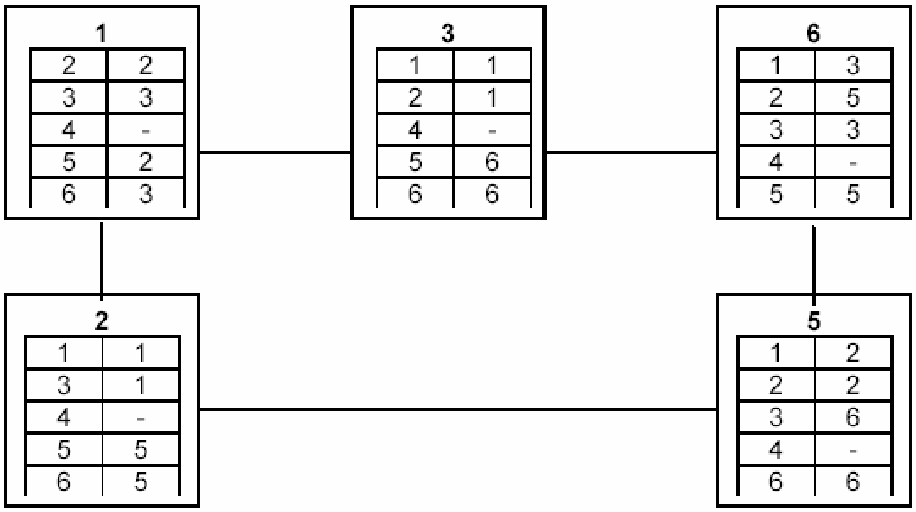
The maximum available bandwidth criterion tries to route traffic along pipes with the highest "cross-section" to the destination. This approach tends to spread traffic across the various links in the network. This approach is inefficient relative to minimum hop routing in that it may use longer paths.

At very low traffic loads, the delay across the network is the sum of the transmission times and the propagation delays. If all links are about the same length and bit rate, then minimum hop routing and minimum delay routing will give the same performance. If links vary widely in length, then minimum hop routing may not give the same performance as minimum delay routing.

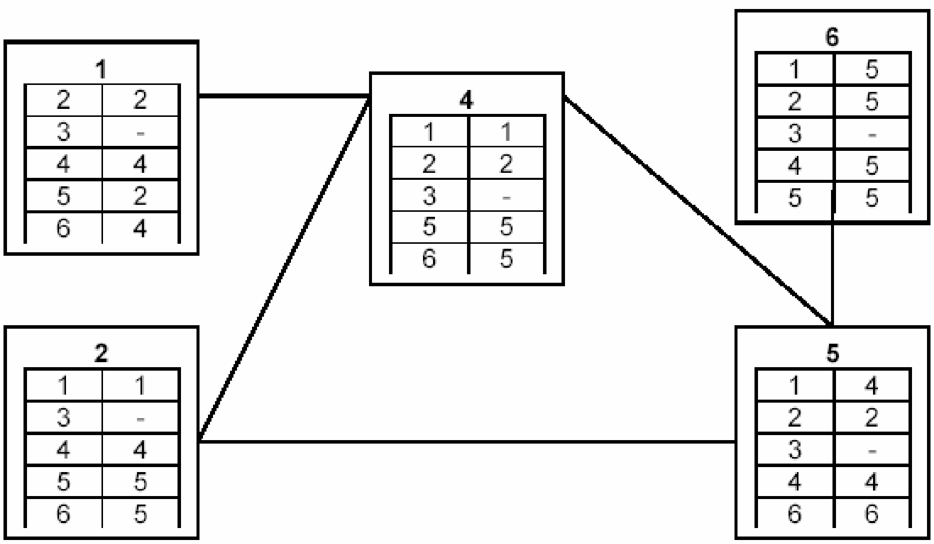
Minimum hop routing will yield the same paths as maximum available bandwidth routing if links are loaded to about the same levels so that the available bandwidth in links is about the same. When link utilization varies widely, maximum available bandwidth routing will start using longer paths.

7.29. Consider the datagram packet network in Figure 7.25. Reconstruct the routing tables (using minimum hop routing) that result after node 4 fails. Repeat if node 3 fails instead.

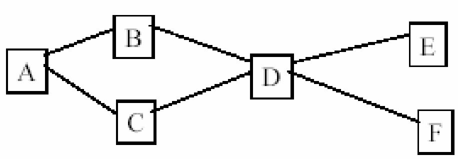
Solution:



If node 3 fails:



7.30. Consider the following six-node network. Assume all links have the same bit rate *R*.



Solutions follow questions:

(a) Suppose the network uses datagram routing. Find the routing table for each node, using minimum hop routing.

Node A:

destination	next node
B	B
C	C
D	B
E	B
F	B

Node B:

destination	next node
A	A
C	D
D	D
E	D
F	D

Node C:

destination	next node
A	A
B	A
D	D
E	D
F	D

Node D:

destination	next node
A	B
B	B
C	C
E	E
F	F

Node E:

destination	next node
A	D
B	D
C	D
D	D
F	D

Node F:

destination	next node
A	D
B	D
C	D
D	D
E	D

(b) Explain why the routing tables in part (a) lead to inefficient use of network bandwidth.

Minimum-hop based routing causes all traffic flow through the minimum hop path, while the bandwidth of other paths of equal and longer distance are unused. It is inefficient in its usage of network bandwidth and likely to cause congestion in the minimum-hop paths.

For example, in the routing table of node A, the traffic flows from A to E and A to F both pass through node B, and leave the bandwidth of link AC and CD unused.

(c) Can VC routing give better efficiency in the use of network bandwidth? Explain why or why not.

Yes. During the VC setup, the links along the path can be examined for the amount of available bandwidth, if it below a threshold, the connection will be routed over alternate path. VC routing allows more even distribution of traffic flows and improves the network bandwidth efficiency.

(d) Suggest an approach in which the routing tables in datagram network are modified to give better efficiency. Give the modified routing tables.

Assign a cost to each link that is proportional to the traffic loading (number of connections) of the link. A higher cost is assigned to more congested links. Thus the resulting routing table avoids congested links and distributes traffic more evenly.

The changes in the routing table are in bold and italicized.

Node A:

destination	next node
B	B
C	C
D	B
E	***C***
F	***C***

Node B:

destination	next node
A	A
C	***A***
D	D
E	D
F	D

Node C:

destination	next node
A	A
B	A
D	D
E	D
F	D

Node D:

Destination	next node
A	B
B	B
C	C
E	E
F	F

Node E:

destination	next node
A	D
B	D
C	D
D	D
F	D

Node F:

destination	next node
A	D
B	D
C	D
D	D
E	D

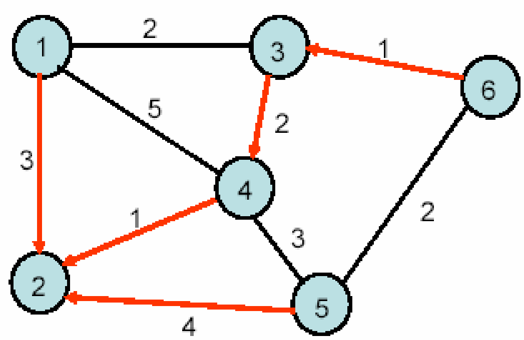
7.32. Consider the network in Figure 7.30.

Solutions follow questions:

(a) Use the Bellman-Ford algorithm to find the set of shortest paths from all nodes to destination node 2.

Iteration	Node 1	Node 3	Node 4	Node 5	Node 6
Initial	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)
1	(2 , 3) (3 , ∞) (4 , ∞)	(1 , ∞) (4 , ∞) (6 , ∞)	(1 , ∞) (2 , 1) (3 , ∞) (5 , ∞)	(2 , 4) (4 , ∞) (6 , ∞)	(3 , ∞) (5 , ∞)
2	(2 , 3) (3 , ∞) (4 , 6)	(1 , 5) (4 , 3) (6 , ∞)	(1 , 8) (2 , 1) (3 , ∞) (5 , 7)	(2 , 4) (4 , 4) (6 , ∞)	(3 , ∞) (5 , 6)
3	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 7)	(1 , 8) (2 , 1) (3 , 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 8)	(3 , 4) (5 , 6)
4	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 5)	(1 , 8) (2 , 1) (3 , 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 6)	(3 , 4) (5 , 6)

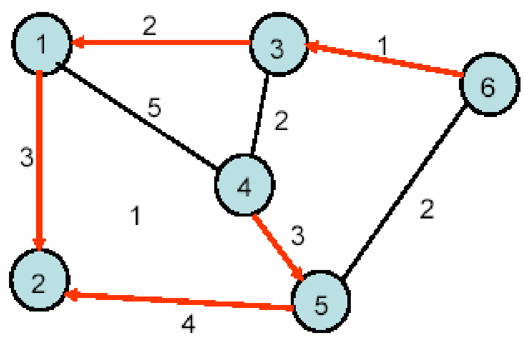
The set of paths to destination 2 are shown below:



Now continue the algorithm after the link between node 2 and 4 goes down.

Iteration	Node 1	Node 3	Node 4	Node 5	Node 6
Before break	(2 , 3)	(4 , 3)	(2 , 1)	(2 , 4)	(3 , 4)
1	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 5)	(1 , 8) (3, 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 6)	(3 , 4) (5 , 6)
2	(2 , 3) (3 , 5) (4 , 10)	(1 , 5) (4 , 7) (6 , 5)	(1 , 8) (3 , 5) (5 , 7)	(2 , 4) (4 , 8) (6 , 6)	(3 , 4) (5 , 6)
3	(2 , 3) (3 , 7) (4 , 10)	(1 , 5) (4 , 7) (6 , 5)	(1 , 8) (3 , 7) (5 , 7)	(2 , 4) (4 , 8) (6 , 6)	(3 , 6) (5 , 6)
4	(2 , 3) (3 , 7) (4 , 12)	(1 , 5) (4 , 9) (6 , 7)	(1 , 8) (3 , 7) (5 , 7)	(2 , 4) (4 , 10) (6 , 8)	(3 , 6) (5 , 6)

The new set of paths are shown below:

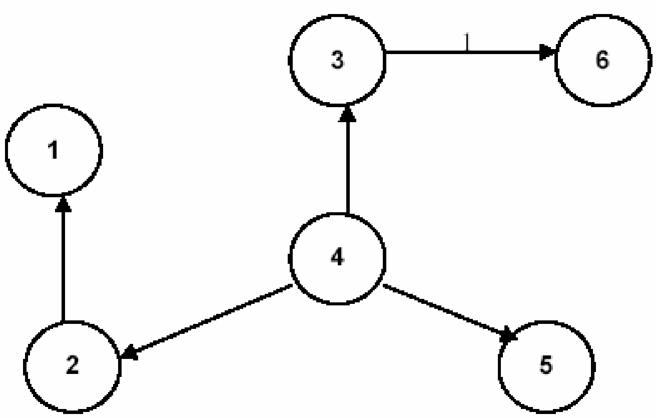


7.33. Consider the network in Figure 7.30.

Solutions follow questions:

(a) Use the Dijkstra algorithm to find the set of shortest paths from node 4 to other nodes.

Iteration	N	D_1	D_2	D_3	D_5	D_6
Initial	{ 4 }	5	1	2	3	∞
1	{ 2, 4 }	4	**1**	2	3	∞
2	{ 2, 3, 4 }	4	1	**2**	3	3
3	{ 2, 3, 4, 5 }	4	1	2	**3**	3
4	{ 2, 3, 4, 5, 6 }	4	1	2	3	**3**
5	{1, 2, 3, 4, 5, 6}	**4**	1	2	3	3

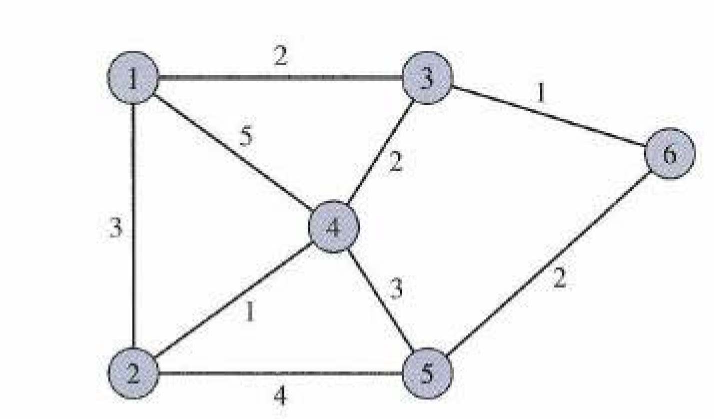


(b) Find the set of associated routing table entries.

Destination	Next Node	Cost
1	2	4
2	2	1
3	3	2
5	5	3
6	3	3

Additional problem

A1: *Using Bellman-Ford algorithm and Dijkstra algorithm, respectively, find the shortest path tree to node 5 in Figure 7.30.*



Solution:

a)Bellman Ford’ s:

Iteration	Node 1	Node 2	Node 3	Node 4	Node 6
Initial	(-1, ∞)	(-1, ∞)	(-1, ∞)	(-1, ∞)	(-1, ∞)
1	(-1, ∞)	(5,4)	(-1, ∞)	(5,3)	(5,2
2	(2,7)	(5,4)	(6,3)	(5,3)	(5, 2)
3	(3, 5)	(5, 4)	(6, 3)	(5, 3)	(5, 2)
4	(3, 5)	(5, 4)	(6, 3)	(5, 3)	(5, 2)

b)Dijkstra’ s :

Iteration	N	D 1	D 2	D33	D 4	D 6
Initial	{5}	∞	4	∞	3	2
1	{5,6}	∞	4	∞	3	***2***
2	{5,6,4}	7	4	3	***3***	2
3	{5,6,4,3}	5	***4***	3	3	2
4	{5,6,4,3,2}	5	4	***3***	3	2
5	{5,6,4,3,2,1}	***5***	5	3	3	2