

Solutions to Assignment 4

1. **[15 marks]** Define the following parameters for a switching network:

N = number of hops between two given end systems

L = message length in bits

B = data rate, in bits per second (bps), on all links

P = fixed packet size, in bits

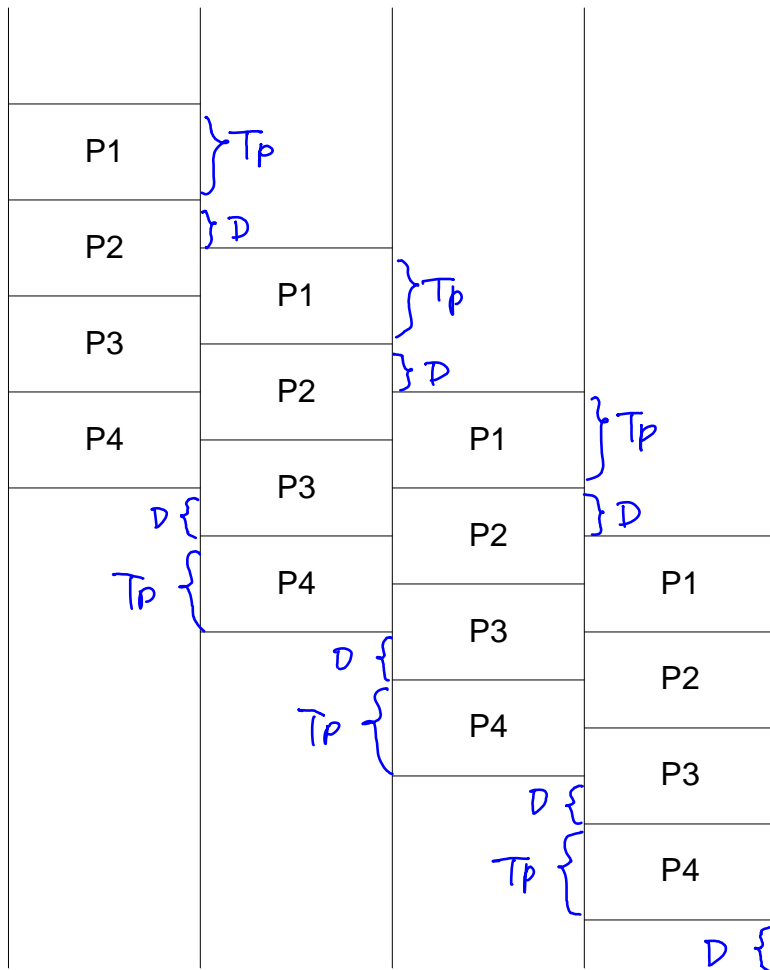
H = overhead (header), bits per packet

S = call setup time (circuits switching or virtual circuit) in seconds

R = call release time (circuits switching or virtual circuit) in seconds

D = propagation delay per hop in seconds

For $N=4$, $L=3200$, $B=9600$, $P=1024$, $H=16$, $S=0.2$, $D=0.001$, compute the end-to-end delay for circuit switching, virtual circuit packet switching, and datagram packet switching. Assume that there are no acknowledgments. Ignore processing delay at nodes.



Packet size (P) = 1024 bits

Header size (H) = 16 bits, which leaves $1024 - 16 = 1008$ bits

Message size (L) = 3200 bits

Thus number of packets required to transmit message = $3200/1008 = 3.17$ packets. The packet size is fixed. So 4 packets of 1024 bits are needed.

T_p = time to transmit a full packet (1024 bits)

= 1024 bits / (9600 bits/sec) = 0.107 secs
 R is not considered here.

Circuit Switching:

- $T_{total} = S + H/B + N \cdot D = 0.2 + 3200/9600 + 4 \cdot 0.001 = 0.2 + 0.333 + 0.004 = 0.537 \text{ sec.}$

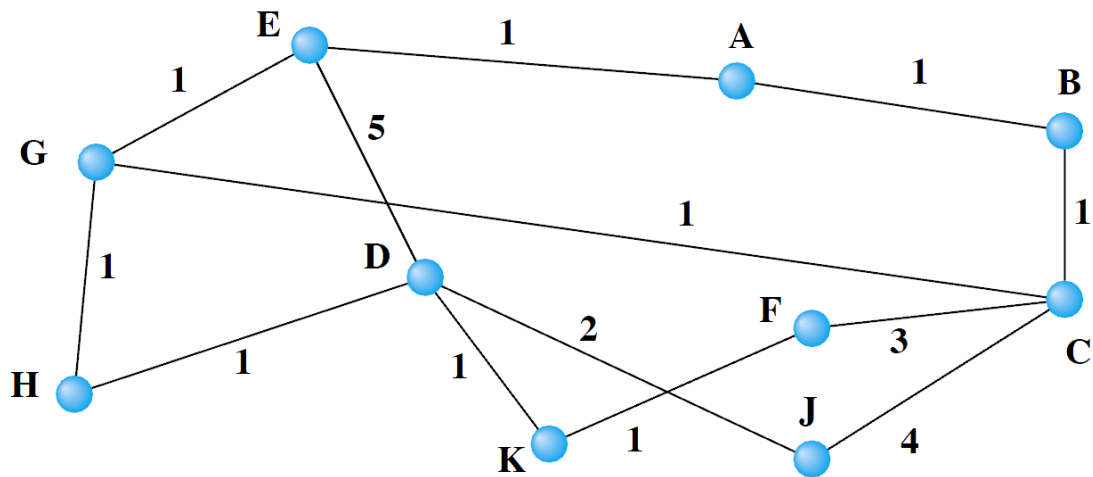
Datagram Packet Switching

- $T_{total} = 4 \cdot T_p + 3 \cdot (D + T_p) + D = 4 \cdot 0.107 + 3 \cdot (0.001 + 0.107) + 0.001 = 0.428 + 0.324 + 0.001 = 0.753 \text{ sec.}$

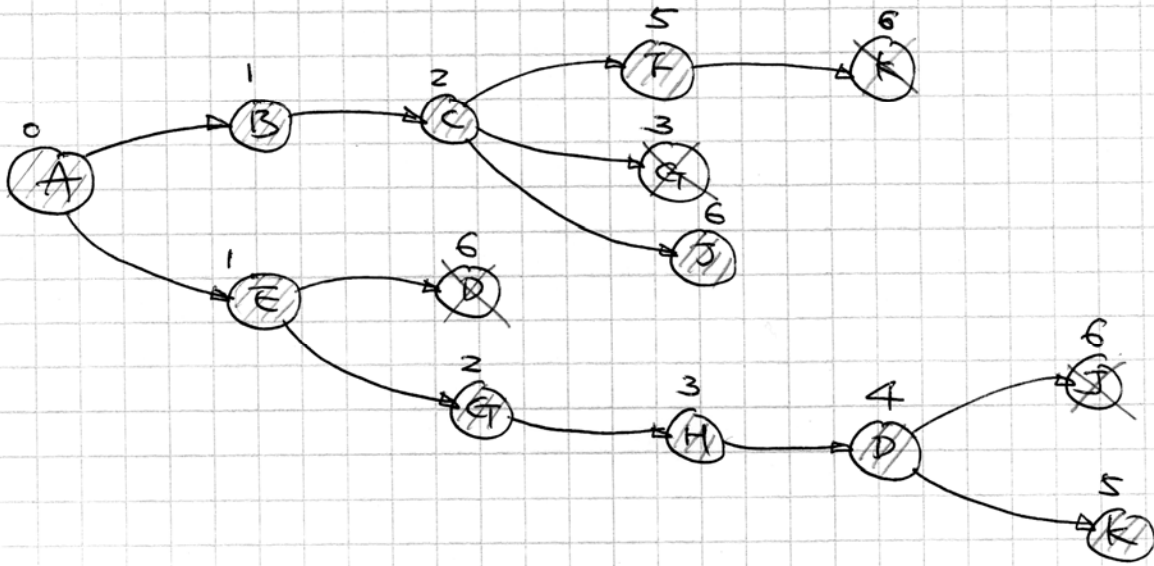
Virtual Circuit Packet Switching:

- Same as Datagram Packet Switching plus time for setup.
- $T_{total} = T_s + 0.753 = 0.2 + 0.753 = 0.953 \text{ sec.}$

2. [20 marks] Apply Dijkstra's algorithm to the following network to generate a shortest-path-tree for node A and node B, and build a routing table for node A and node B.



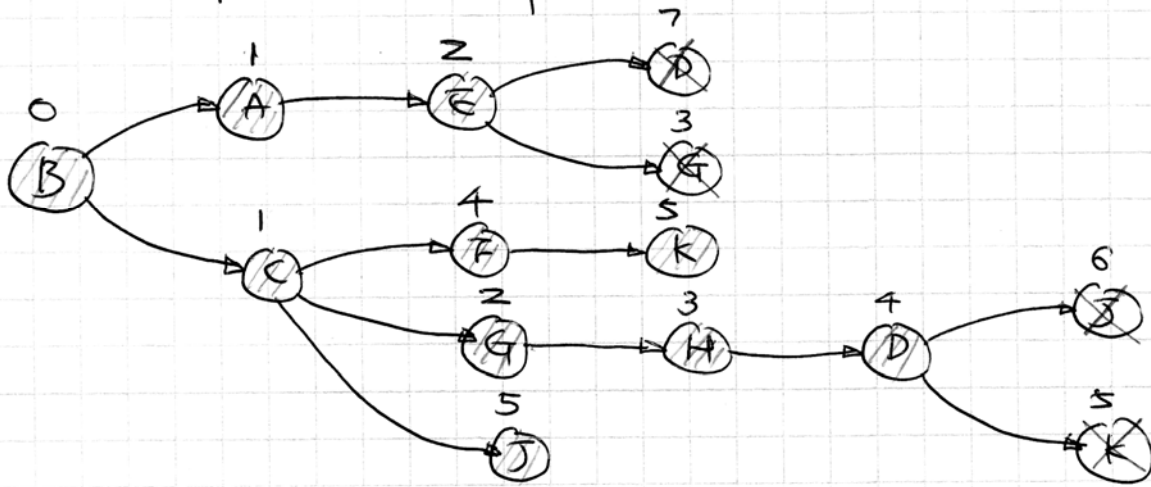
shortest path tree for node A:



Routing table for node A:

Destination	Next hop	Cost
A	-	0
B	-	1
C	B	2
D	E	4
E	-	1
F	B	5
G	E	2
H	E	3
J	B	6
K	E	5

Shortest path tree for node B:



Routing table for node B:

Destination	Next hop	Cost
A	-	1
B	-	0
C	-	1
D	C	4
E	A	2
F	C	4
G	C	2
H	C	3
J	C	5
K	C	5

3. **[20 marks]** In the following figure, there are eight (8) sub-networks belonging to a company. The company has Class C addresses, 192.121.152.0. Help the company partition the network into the needed number of sub-networks, by identifying the subnet mask, assigning addresses to each of the sub-networks, assigning needed IP addresses to routers and computers. Suppose that there are two computers on each sub-network.

Subnet mask: 255.255.255.224

Subnet 0: 192.121.152.0

Subnet 1: 192.121.152.32

Subnet 2: 192.121.152.64

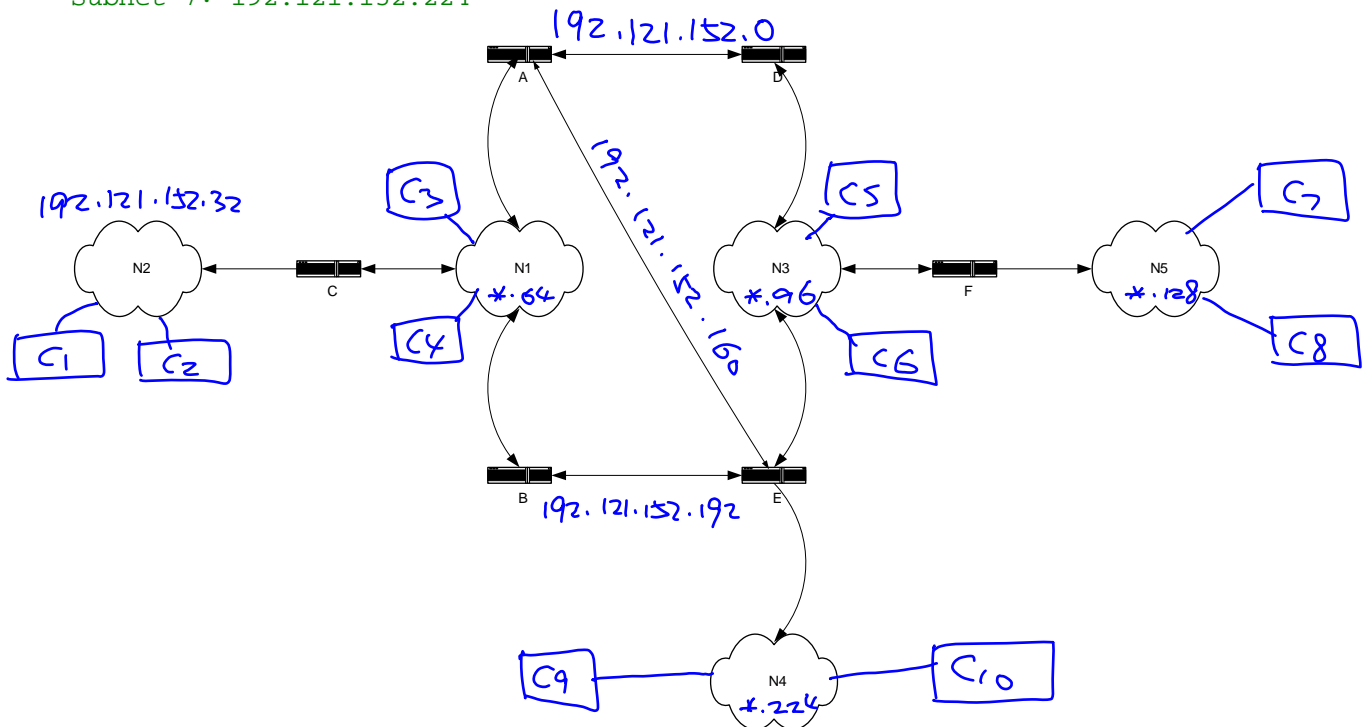
Subnet 3: 192.121.152.96

Subnet 4: 192.121.152.128

Subnet 5: 192.121.152.160

Subnet 6: 192.121.152.192

Subnet 7: 192.121.152.224



A: 192.121.152.1, 192.121.152.161, 192.121.152.65

B: 192.121.152.66, 192.121.152.193

C: 192.121.152.33, 192.121.152.67

D: 192.121.152.2, 192.121.152.97

E: 192.121.152.162, 192.121.152.98, 192.121.152.194, 192.121.152.225

F: 192.121.152.99, 192.121.152.129

C1: 192.121.152.34

C2: 192.121.152.35

C3: 192.121.152.68

C4: 192.121.152.69

C5: 192.121.152.100

C6: 192.121.152.101

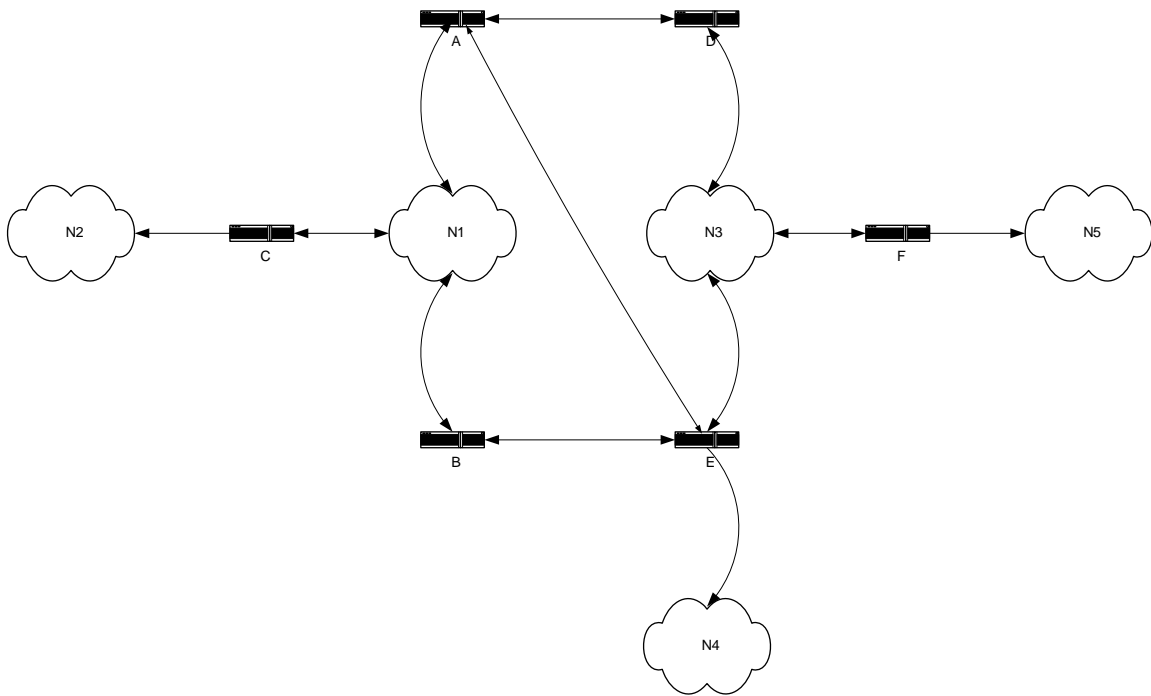
C7: 192.121.152.130

C8: 192.121.152.131

C9: 192.121.152.226

C10: 192.121.152.227

4. **[20 marks]** Compute RIP routing table for all the routers in the following figure. Then describe how an IP datagram is routed from a computer on N2 to a computer on N5. In the figure, rectangles are used to represent routers and clouds are used to represent networks.



Routing table for A

Destination	Next hop	Cost
N1	-	1
N2	C	2
N3	D	2
N4	E	2
N5	D	3

Routing table for B

Destination	Next hop	Cost
N1	-	1
N2	C	2
N3	E	2
N4	E	2
N5	E	3

Routing table for C

Destination	Next hop	Cost
N1	-	1
N2	-	1
N3	A	3
N4	B	3
N5	A	4

Routing table for D

Destination	Next hop	Cost
N1	A	2
N2	A	3

N3	-	1
N4	E	2
N5	F	2

Routing table for E

Destination	Next hop	Cost
N1	B	2
N2	B	3
N3	-	1
N4	-	1
N5	F	2

Routing table for F

Destination	Next hop	Cost
N1	A	2
N2	A	3
N3	-	1
N4	-	1
N5	F	2

5. **[15 marks]** A 4460-byte TCP segment is carried by an IP datagram with no option. The IP datagram with Identification of 0000H is to be transmitted and needs to be fragmented because it will pass through an Ethernet (N1) with a maximum payload of 1500 bytes. Show the Identification, Total Length, More Flag, and Fragment Offset values in each of the resulting fragments. The above fragmentations will have to go through another network (N2) with a maximum payload of 1000 bytes. Again, show the Identification, Total Length, More Flag, and Fragment Offset values in each of the resulting fragments.

One of the possible solutions:

	ID	Total length	More	Offset
Original	0000	4480 (20+4460)	0	0
Fragment1	0000	764 (20+744)	1	0
Fragment2	0000	764 (20+744)	1	93
Fragment3	0000	764 (20+744)	1	186
Fragment4	0000	764 (20+744)	1	279
Fragment5	0000	764 (20+744)	1	372
Fragment6	0000	760 (20+740)	0	465

6. **[10 marks]** Refer to Frame 3. By fragmenting the carried IP datagram to two fragments with about equal length, form two corresponding Ethernet-II frames.

Frame 3:

```

0000  00 1f 3a 68 6b 0a 78 8d f7 b0 6d 72 08 00 45 00  ..:hk.x...mr..E.
0010  00 34 1e 31 00 00 38 06 76 f2 4a 7d e2 6d c0 a8  .4.l..8.v.J}.m..
0020  00 0e 00 50 12 d4 bd 31 75 da 02 90 60 07 80 10  ...P...lu...`...
0030  02 95 1b 92 00 00 01 01 05 0a 02 90 60 06 02 90  .....`...
0040  60 07  `

```

From Frame 3, it is clear that the total length of the IP datagram is 52 bytes (carrying TCP segment of 32 bytes). The original IP datagram can be divided into two fragments with respective total lengths of 36 and 36. In the two resulting IP datagram fragments, all the IP header fields are kept the

same except for the Total Length, More flag, and the Checksum. The frame padding is not considered.

Frame 3-1:

```
0000 00 1f 3a 68 6b 0a 78 8d f7 b0 6d 72 08 00 45 00
0010 00 24 1e 31 20 00 38 06 57 02 4a 7d e2 6d c0 a8
0020 00 0e 00 50 12 d4 bd 31 75 da 02 90 60 07 80 10
0030 02 95 1b 92 00 00 01 01 05 0a 02 90 60 06 02 90
0040 60 07
```

Frame 3-2:

```
0000 00 1f 3a 68 6b 0a 78 8d f7 b0 6d 72 08 00 45 00
0010 00 24 1e 31 00 02 38 06 77 00 4a 7d e2 6d c0 a8
0020 00 0e 00 50 12 d4 bd 31 75 da 02 90 60 07 80 10
0030 02 95 1b 92 00 00 01 01 05 0a 02 90 60 06 02 90
0040 60 07
```