Week 1:

1. Sample Question: Message Switching vs Packet Switching (Please discuss)

Q:

This was a past exam question. Students are strongly encouraged to discuss the solution in the comments below. It may help to visualise the delays using the timing diagrams used in the lectures.

Consider an *N* -hop path (i.e. *N-1* intermediate routers) between a source and destination as depicted in the figure below. The source wants to transmit a file of size *kP* bits to the destination. There are two options: (i) Transmit the entire file as one large chunk (i.e. packet) of data. This is what we refer to as *message switching* or (ii) Break up the file into *k* packets, each of size *P* bits and transmit these packets back-to-back. As you may recall, this is *packet switching* .

All links (i.e. hops) have the same transmission delay and propagation delay. Assume that the propagation delay of a link is *d* sec. Assume that the transmission delay for transmitting P bits on a link is *T* sec. Thus, transmitting the entire file (as is the case in message switching) on a link takes *kT* sec.

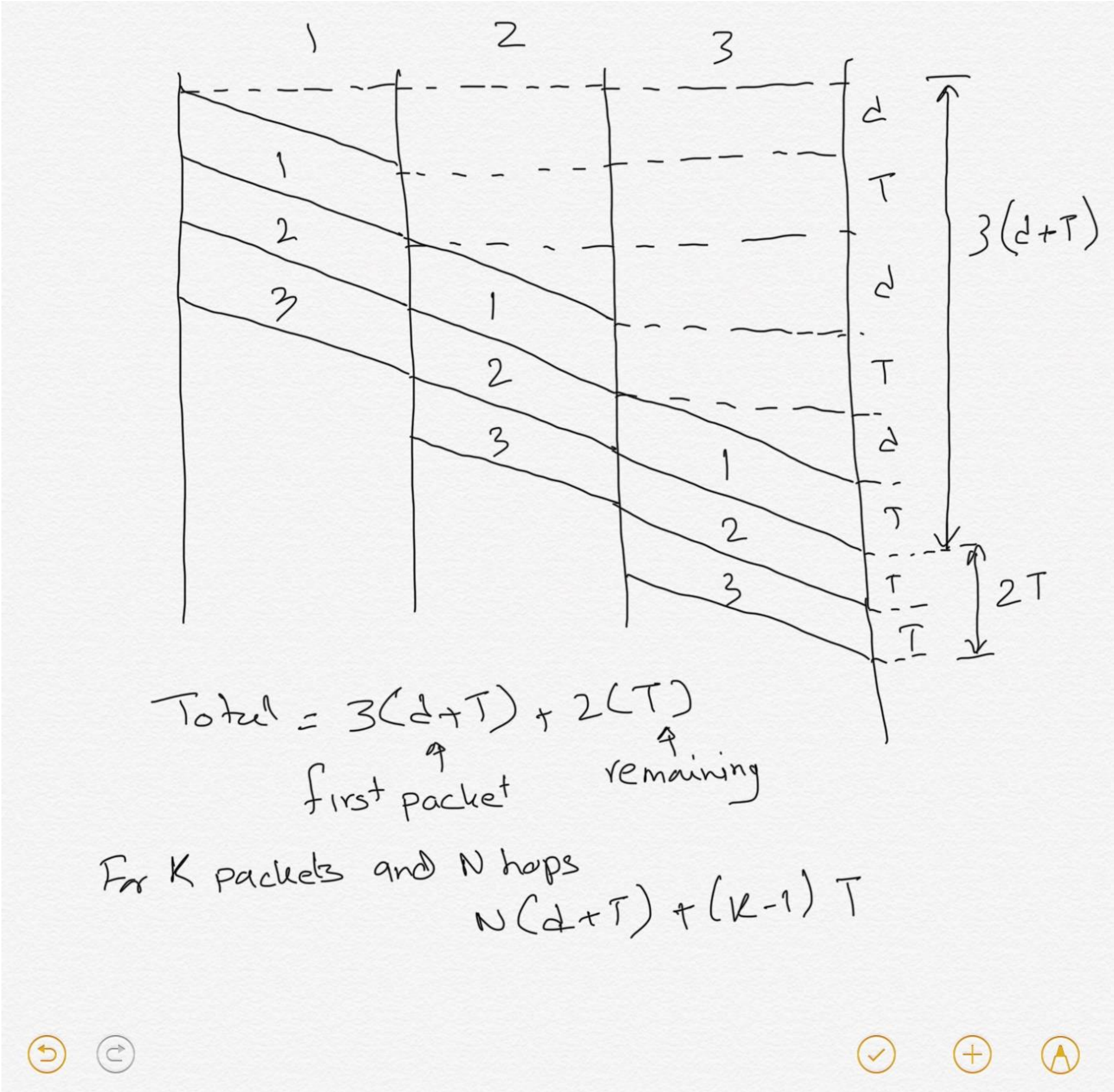
Assume that there is no other traffic on the network. Ignore the time taken by each router to process each packet (or message). Assume that packet headers are negligible.

Compare the end-to-end delay incurred in transmitting the file for the two options outlined above, i.e. message switching vs packet switching. Which incurs lower delay and under what conditions?

Solution:

For Message switching: N (kT+d)

For Packet Switching: Consider this timing diagram with 3 Hops and 3 back to back packets. d is the propagation delay and T is the transmission delay at each hop. 3(d+T) is the time taken by the first packet to reach the destination. When this first packet arrives at the destination, all other packets are already in the pipeline. Next two packets would incur only the transmission delay. At the end just generalise this and come up with N(d+T) + (k-1) T.



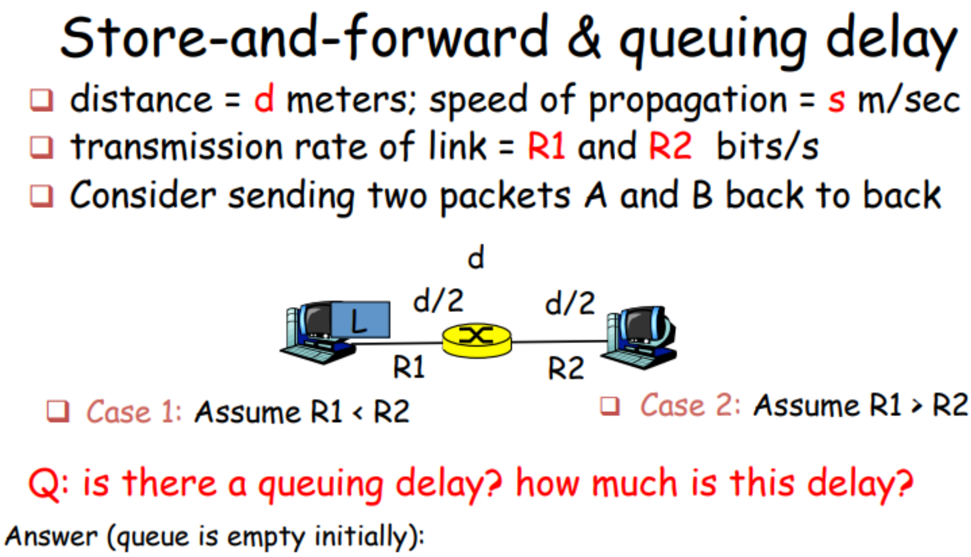
Comparing both delays you can found out that delay for message switching would be more than packet switching as long as you have more than 1 chunks (k >1) and more than 1 hop (N >1) and T ! =0.

Why do we ignore propagation delay for the other two packets?

back-to-back. When the first packet arrives at the destination at time N(d+t) in the above example, the second has already reached the last hop router (N-1) and ready for its last transmission. You thus continue receiving more packets after every T.

2. Sample Question: To Queue or Not to Queue is the Question (Please discuss)

Q:



Note that the two packets are transmitted back to back. The first (L/R1) is for the transmission of the first packet. The second (L/R1) is for the transmission of the second packet.

Solution:

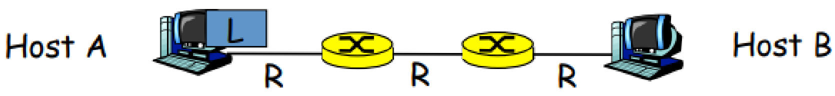
Case 1: There is no delay.

Case 2: The queuing delay would be L/R2 - L/R1

3. Sample Question: Throughput (Please discuss)

Q:

Quiz: Throughput



- Suppose: Host A has a large file of size F bits to send to Host B
- File is split into N packets, each of length L bits (i.e., N=F/L)
- Ignore propagation delay for now

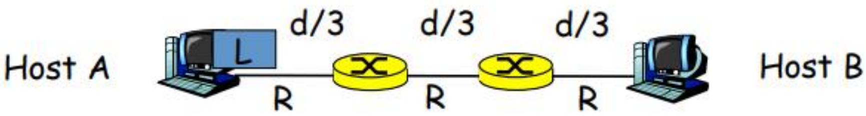
❑ Question 1: how long does it take for the file to arrive at B?

❑ Question 2: what is the average throughput achieved when sending the file?

Note: throughput = number of total bits sent / total time taken

1-1

Quiz: Throughput



- Suppose: Host A has a large file of size F bits to send to Host B
- File is split into N packets, each of length L bits (i.e., N=F/L)
- Do NOT ignore propagation delay (assume prop. speed = s m/s)

❑ Question 1: how long does it take for the file to arrive at B?

❑ Question 2: what is the average throughput achieved when sending the file?

A

Note that, the propagation delay is only for the first packet and hence not related to the total number of packets.

There are 3 hops, $3(d/3s) = d/s$

The first packet will take $3L/R$ as there are three hops. Once this packet arrives at Host B, rest of the packets (N-1) would take L/R each as they have been transmitted back to back.

Solution:

Scenario 1 Q1:

Total delay = Dproc + Dqueue + Dtrans + Dprop

Ignoring processing and propagation delay. Also queueing delay is zero.

Total delay = $3L/R + (N - 1)L/R = (2+N) L/R = (2 + F/L) (L/R) = (2L + F)/R$

Scenario 1 Q2: Throughput

$F / [(2L +F) / R] = FR / (F+2L) = R / (1+2L/F)$

Scenario 2 Q1:

Total delay = $(2L +F) / R + d/s$

Scenario 2 Q2:

Throughput = $F / [(2L + F) / R + d/s] = FR / [2L + F + dR/s]$

Week 2:

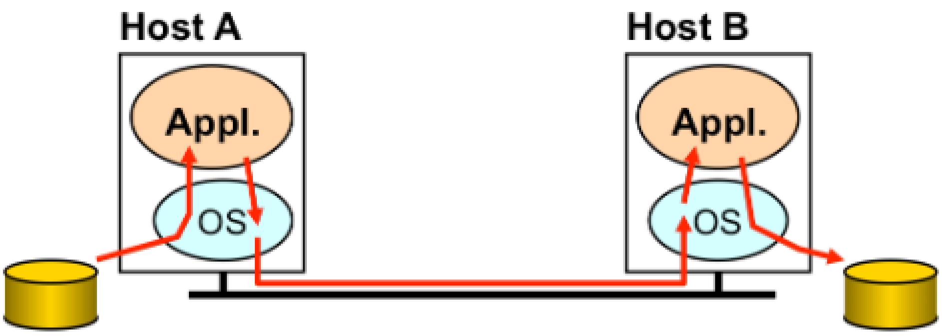
1. Sample Question: End-to-End Arguments in System Design (Please discuss)

Q:

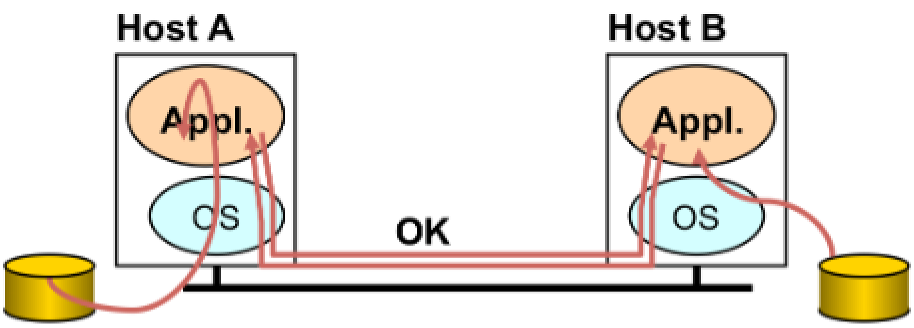
This question is centered around the fundamental paper "End-to-end Arguments in System Design" which is available here - Saltzer

Consider that we wish to implement a reliable file transfer protocol.

One approach is to make each step of the file transfer reliable and string them together to make the end-to-end process reliable. This is depicted in the figure below.



The other approach is to not worry about implementing reliability along each step but rather to make use of an end-to-end check (simple error detection at the receiver followed by feedback to the sender) and retransmit if necessary. This is depicted in the figure below.



Discuss the pros and cons of each approach. Which approach would you pick?

Solution: (We are only emphasising the networking context of the problem setting)

Providing hop by hop reliability checks increases the required processing but frees up resources that would be wasted in transmitting an erroneous packet upstream. In approach 2, an erroneous packet may travel all the way to the destination and gets dropped there. The actual design used for TCP/IP stack uses reliability checks at each hop as well as final end-to-end check at the transport layer.

2. Sample Question: Headers, Headers and Headers (Please discuss)

Q: Consider a 4-layer protocol stack with the following layers (numbered 1 to 4 from bottom to top), physical, network, transport and application. The physical layer uses one-byte flags to mark both the start and end of a packet (i.e. a one-byte header and a one-byte trailer). The other three layers all use four-byte headers. The transport layer imposes that the maximum packet payload (i.e. data portion) for this layer must not exceed 100 bytes. Other layers have no such restrictions. Calculate the percentage overhead assuming an application message of 450 bytes is to be transmitted.

Solution:

The application layer message size is 450 bytes + 4 bytes of header = 454 bytes.

Split into 4 transport layer segments, each with payload 100, 100, 100, 100 and 54 bytes, respectively.

We get transport layer segments of the following sizes: 104, 104, 104, 104 and 58 bytes.

Adding the 4-byte network layer header to each datagram gives us: 108, 108, 108 and 62 bytes.

Adding 2-byte physical layer header to each packet gives us: 110, 110, 110, 110 and 64 bytes.

Summing up all the packets, we have 504 bytes to transmit a 450-byte application message. Thus, total headers = 54 bytes.

Percentage overhead = 54/450 = 12%

3. Sample Question: Persistent HTTP (Please discuss)

Q : Among the following, in which case would you get the greatest improvement in performance with persistence HTTP as compared to non-persistence HTTP?

a) Low throughput network paths (irrespective of distance)

b) High throughput network paths (irrespective of distance)

c) Long distance network paths (irrespective of throughput)

d) High throughput, short-distance network paths

e) High throughput, long-distance network paths

Solution:

Total delay = some multiple of RTT + Transmission Time of object.

Remember that persistence HTTP will only reduce the first component of the delay. The time to transmit the object does not change whether we use persistent or non-persistent HTTP.

First up, for a low throughput path, the second component of the delay will make up a non-trivial component of the total delay. As such, the savings achieved by using persistence will not be as much as over a high throughout path when the transmission time is very negligible.

This suggests that we will get the most savings over a high throughput path. The question now is whether the distance of the path matters.

Let us say persistent HTTP achieves a n(RTT) savings over non-persistent HTTP.

Q: when will this saving have the MOST impact?

A: This will be when the RTT value is large.

Q: When is the RTT large?

A: When the network path is long.
So, the answer is E.

4. Sample Question: Conditional GET (Please discuss)

Q: We explored two ways that a web cache can use to check if the cached content is up-to-date: (i) If-Modified-Since and (ii) E-tag.

(Reference: Slides 58-59 in the Week 2 slide set on HTTP). Can you think of a scenario where one method would be preferred to another?

Solution:

Suppose an object is modified and then changed back to its original state. In this case, if-modified-since treats it as a new object as the last modified time has changed while E-tag would be the same for the object.

Shaowei Ma mentioned (in comments below) the case of multiple updating of an object within a second because If-Modified-Since has a 1 second granularity.

5. Sample Question: Web Caching (Please discuss)

Q:

Consider the figure below for which there is an institutional network connected to the Internet. Suppose that the average object size is 900,000 bits and that that the average request rate from the institution's browsers to the origin server is 1.5 requests per second. Suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response in two seconds on average. Model the total average response time as the sum of the average access delay and the average Internet delay. For the average access delay, use A/(1-AB) where A is the average time required to send an object over the access link and B is the arrival rate of objects to the access link. You can assume that the HTTP request messages are negligibly small and thus create no traffic on the network or the access link.

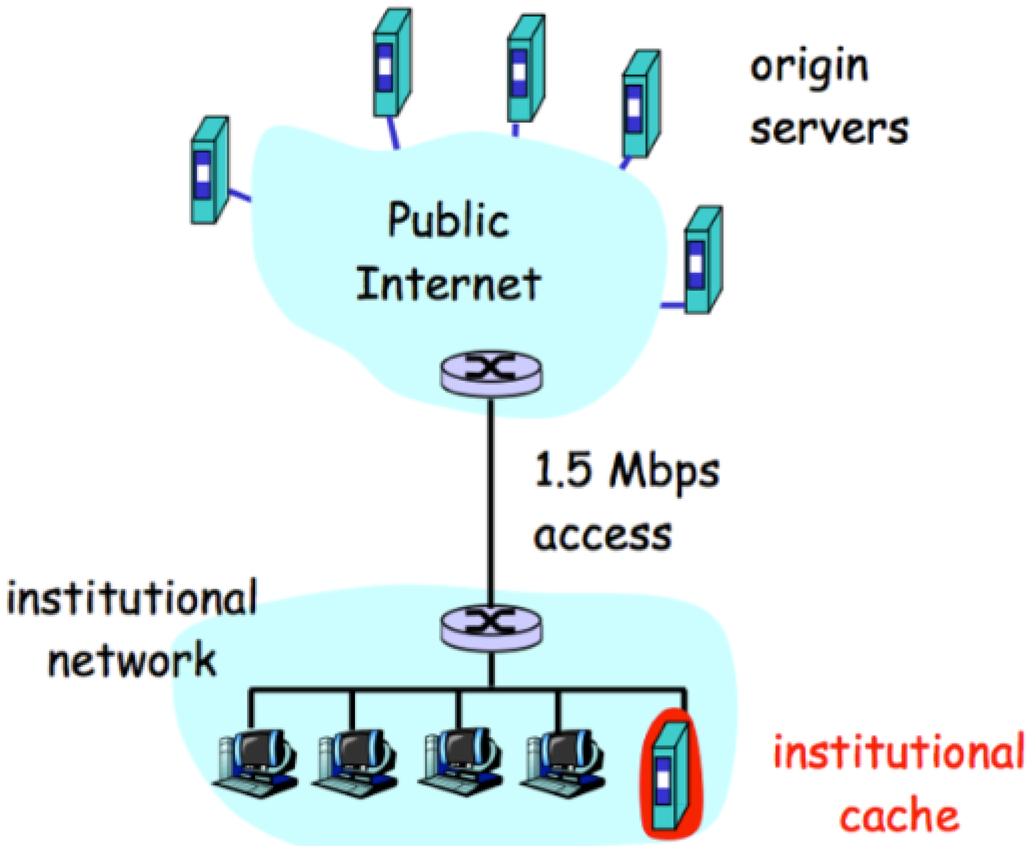


Figure 1: Figure for web cache problem

Question 1. Find the total average response time?

Question 2. Now suppose a cache is installed in the institutional LAN. Suppose the cache hit rate is 0.4. Find the total response time.

Solution:

Q1.

A = 0.9 Mb / 1.5Mbps = 0.6sec

B = 1.5 requests per second

Average access delay = A / (1-AB) = 0.6 / (1 - 0.6 x 1.5) = 6 sec

Average delay = average access delay + average Internet delay = 6 + 2 = 8 sec

Q2.

B = 0.9 requests / sec [60% cache miss rate]

Average access delay = A / (1 - AB) = 0.6 / (1 - 0.6 x 0.9) = 1.3 sec

Response time for cache miss cases = 1.3 + 2 = 3.3 sec

The average response time would be 0.6*3.3 +0.4(0) = 1.98 sec (neglecting the delay for the cache hit case, 40% of the time object transferred from the institutional cache).

Week 3:

1. Sample Question: DNS delays can be significant (please discuss)

Q: Suppose within your Web browser you click on a link to obtain a web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host receives the IP address from DNS and that iterative queries are used. Let the successive visits to the DNS servers incur an RTT of RTT1, ….., RTTn. Further suppose that the webpage associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

To lay the solution out in sequence of events:

RTT1 : DNS server 1

RTT2 : DNS server 2

.

.

.

RTTn: DNS server n

Now we have the IP address of the given URL

We don't have to assume whether it's a persistent connection or not since we're only looking at one object.

RTT0: Establish a TCP connection

RTT0: Retrieve the single Object/html

2 * RTT0 + RTT1 + ... + RTTn

Also, note that When fetching a website, your browser is almost always fetching the index page first unless explicitly indicated otherwise

2. Sample Question: Using public DNS servers not always a good thing? (Please Discuss)

Q: You will often find recommendations on the Internet to use openly available DNS servers such as Google DNS (8.8.8.8, 8.8.4.4) or OpenDNS instead of your local ISP's DNS servers. This is because these servers are typically able to resolve queries much faster than your local ISP DNS servers. Can you explain why using such openly available DNS servers may actually lead to poor download speeds and increased latency, e.g. when downloading a large movie from iTunes (as compared to using your local ISP's DNS servers)?

Solution:

Have to consider two scenarios. Latency in DNS resolution and latency in actual data transfer phase. Public DNS servers are typically located far from a client as compared to the local ISP thus would involve more latency in the name resolution process. On the other hand, the chances of a 'cache hit' at public DNS servers is higher than your LNS. Thus, it would also depend on whether the DNS mapping is cached at LNS or not.

Latency in actual data transfer would depend on the instance of server returned by the DNS process (think of a CDN). If you are using public DNS system, you would get back a CDN node located close to the public DNS server that may not be the closest CDN node to your (client) actual location.

Week 4:

1. Sample Question: Reliable Data Transfer (please discuss)

   Q:

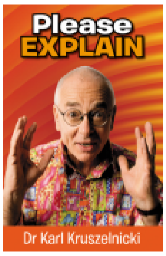
Quiz: RDT 2.2 with only NACKs

- RDT 2.2 uses only ACKs and is robust against packet errors. Would it be possible to implement RDT 2.2 with only NACKs?

  A. YES

  B. NO

  Please explain your rationale



Note that, The NACKs are sent by the receiver and not the sender.

Solution:

Yes. One possible design:

1. For correctly received packets, send NACK (1) when you receive packet 0 and NACK (0) for packet 1.

2. For corrupted packets or receipt of a packet that is not expected, send NACK (0) if expecting packet 0 or NACK (1) if expecting packet 1.

The sender need to keep track of what packet it has transmitted and what NACK it is expecting (NACK (0) or NACK (1))

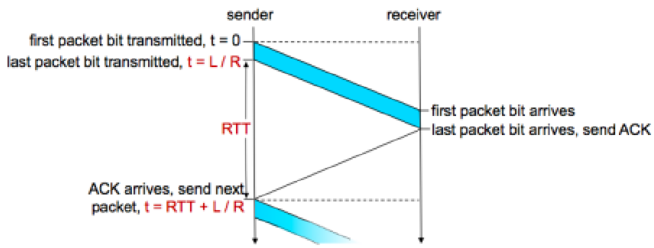
2. Sample Question: Comparing Sliding Window with Stop-and-Wait (please discuss)

Q:

Quiz: Sliding Window Protocols

- Consider a path of bottleneck capacity R, round-trip time T, and maximum segment size L. What is the greatest throughput improvement factor that an ideal pipelined protocol (assuming corruptions and loss are negligible) can provide compared to a stop-and-wait protocol?
  A. 2L/(RT+L)
  B. (L/R)/(T+L/R)
  C. (RT+L)/L
  D. $(TR/L)^2$

rdt3.0: stop-and-wait operation



$$U = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

Solution:

C.

Utilisation for stop and wait: (L/R) / {(L/R) +T}

Utilisation for pipelined protocol = 1 = {(L/R) +T} / {(L/R) +T}

Improvement = {utilisation for pipelined / utilisation for stop and wait}

= {{(L/R) +T} / {(L/R) +T}} / {(L/R) / {(L/R) +T}}

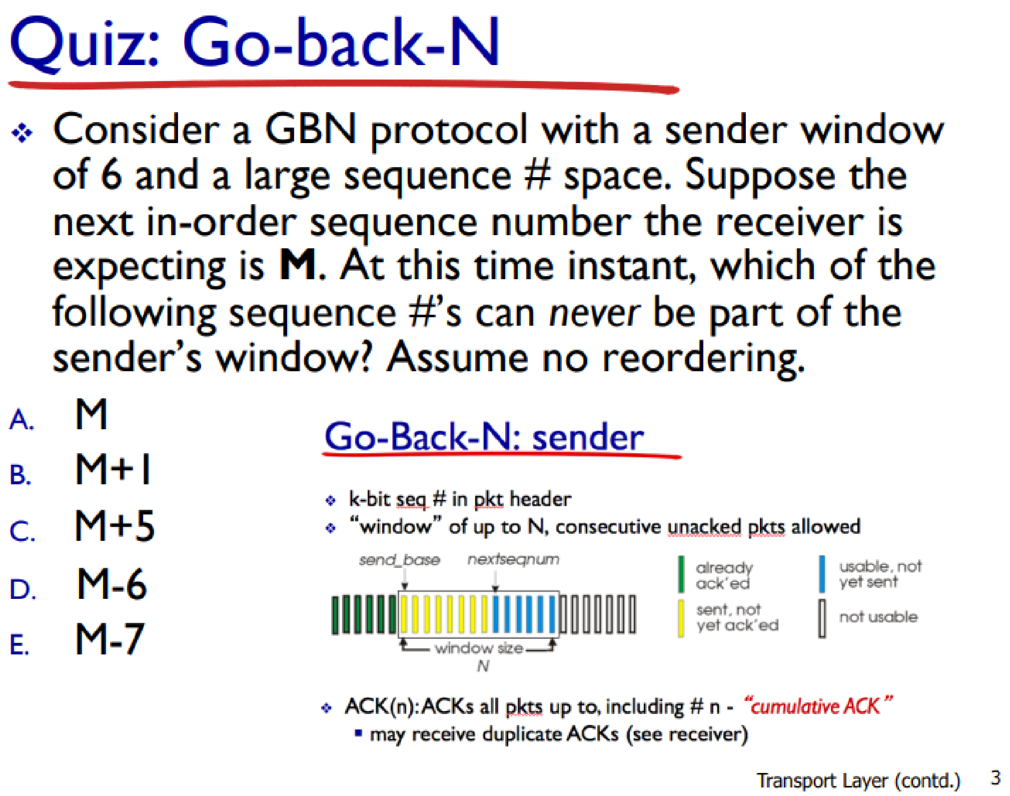
= (RT+L) / L

why the utilization for pipelined protocol is equal to 1?

because the sender sends sufficient # of back to back packets such that it starts receiving the ACK for the first packet in the window as it finishes sending the last packet in the window.

3. Sample Question: GBN (Please discuss)

Q:



Solution: E. M-7

4. Sample Question: Compare GBN and SR (Please discuss)

Q:

Quiz: GBN vs. SR

- Suppose a receiver that has received all packets up to and including sequence number 24 and next receives packet 27 and 28. In response, what are the sequence numbers in the ACK(s) sent out by the GBN and SR receiver respectively?

A. [27,28], [28]
B. [24, 24], [27,28]
C. [27,28], [27,28]
D. [25], [25]
E. [nothing], [27, 28]

Transport Layer 90

Solution: B.

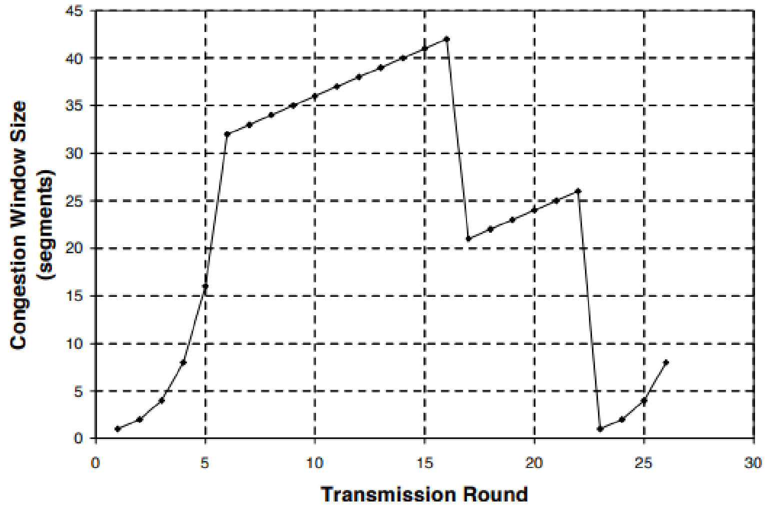
Week 6:

1. Sample Question: TCP Congestion Control (please discuss)

   Q;

Practice Problem

Consider the following plot of TCP window size as a function of time. Assume that the version is Reno.



Network Layer 61

Practice Problem

- Identify the intervals of time when TCP slow start is operating.
- Identify the intervals of time when TCP congestion avoidance is operating.
- After the 16th transmission round, is segment loss detected by a triple duplicate ACK or a timeout?
- What is the initial value of ssthreshold at the first transmission round?

Practice Problem

- What is the value of ssthreshold at the 18th transmission round?
- What is the value of ssthreshold at the 24th transmission round?
- During which transmission round is the 70th segment sent?
- Assuming a packet loss is detected after the 26th round by the receipt of triple duplicate ACK, what will be value of the congestion window size and of ssthreshold?

Solution:

1. 1- 6 and 23-26

2. 7-16, 17-22

3. 3 DupAcks

4. 32 Segments

5. 21 Segments

6. 13 Segments

7. 7th transmission round

8. Both set to 4 Segments.

Week 7:

1. Sample Question: IP Fragmentation

Q: An IP datagram that is 5540 bytes large is sent over a link with an MTU of 1500 bytes and therefore has to be fragmented. State the identifier, fragmentation offset and the value of the MF flag in the IP-header for the different fragments in the table below. You should assume that the IP-header is 20 bytes long and that the identifier field set to 2222. Please note, the table might contain more rows than is absolutely necessary, just so that the task doesn't become too easy.

Solution:

MTU=1500 bytes

IP datagram = 5540 bytes

IP header = 20 bytes

data = 5540-20 = 5520 bytes

Make 4 fragments as follows:

F1: 1480, F2: 1480, F3: 1480, F4: 1080

Identifier Offset Flag (MF)

2222 0 1

2222 185 1

2222 370 1

2222 555 0

2. Sample Question: IP Fragmentation

Q: Why does the re-assembly of IP fragments is only allowed at the destination? Can't a router re-assemble IP fragments if it finds out that the outgoing link can carry a bigger segment (has higher MTU)?

Solution:

Each fragment becomes an independent datagram and may be routed along different paths. A router may not receive all fragments and hence may fail to reassemble a segment.

We also want to keep our core simple and push all complex functionality to the edge devices. Hence, the reassembly is only allowed at the destination without involving the routers in the re-assembly process.

3. Sample Question: Forwarding

Q:

Forwarding

- A forwarding table for a router is as follows:

Address prefix	Next hop
196.94.2.0/24	A
196.94.2.128/25	B
196.94.0.0/16	C
196.94.64.0/18	D
196.76.0.0/14	E
140.0.0.0/8	F
128.0.0.0/2	G
0.0.0.0/1	H

- State the next hop for the following destinations:
  - 139.1.1.1
  - 196.94.2.100
  - 196.94.2.200
  - 196.94.3.100

Solution:

Using longest prefix match:

1. G

2. A

3. B

4. C

4. Sample Question: Network Design

Q; Suppose an ISP has a block of IP addresses starting with 144.4.64.0/19. It wants to create sub-blocks out of its available addresses such that it can meet the following requirements:

i. 32 blocks of 64 IP addresses

ii. 32 blocks of 32 IP addresses

iii. 64 blocks of 16 IP addresses

iv. 128 blocks of 8 IP addresses

v. 256 blocks of 4 IP addresses

Design the sub-blocks and give the slash notation for network address and the broadcast address for each sub-block. How many addresses are still available with the ISP after this allocation?

Solution:

144.4.64.0/19 -> 144.4.64.0 to 144.4.95.255 : Total 8192 Addresses

i. 64 IP addresses -> 2^6 : 6 bits for host and (32-6) 26 bits for the network prefix.

144.4.64.0/26 to 144.4.64.63/26

144.4.64.64/26 to 144.4.64.127/26

....

144.4.71.128/26 to 144.4.71.191/26

144.4.71.192/26 to 144.4.71.255/26

32 * 64 = 2048 addresses

ii. 32 IP addresses -> 2^5: 5 bits for host and (32-5) 27 bits for the network prefix.

144.4.72.0/27 to 144.4.72.31/27

144.4.72.32/27 to 144.4.72.63/27

...

144.4.75.192/27 to 144.4.75.223/27

144.4.75.224/27 to 144.4.75.255/27

32 * 32 = 1024 addresses

iii. 16 IP addresses -> 2^4: 4 bits for host and (32-4) 28 bits for the network prefix.

144.4.76.0/28 to 144.4.76.15/28

144.4.76.16/28 to 144.4.76.31/28

…

144.4.79.224/28 to 144.4.79.239/28

144.4.79.240/28 to 144.4.79.255/28

64 * 16 = 1024 addresses

iv. 8 IP addresses -> 2^3: 3 bits for host and (32-3) 29 bits for the network prefix.

144.4.80.0/29 to 144.4.80.7/29

144.4.80.8/29 to 144.4.80.15/29

…

144.4.83.240/29 to 144.4.83.247/29

144.4.83.248.29 to 144.4.83.255/29

128 * 8 = 1024 addresses

v. 4 IP addresses >> 2^2: 2 bits for host and (32-2) 30 bits for the network prefix.

144.4.84.0/30 to 144.4.84.3/30

144.4.84.4/30 to 144.4.84.7/30

…

144.4.87.248/30 to 144.4.87.251/30

144.4.87.252/30 to 144.4.87.255/30

256 * 4 = 1024

Total used: 2048 + 4 * 1024 = 6144

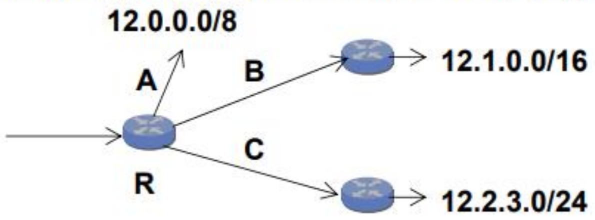
Remaining: 8192 -6144 = 2048

5. Sample Question: Forwarding

Q:

Sample Question

Consider the following 3 routers, where router R has outgoing interfaces A, B and C



- How many IP addresses does the prefix 12.2.3.0/24 represent?

- List the forwarding-table entries for router R?

1

Sample Question (contd.)

- Which outgoing interface does R use for a packet with destination 12.2.3.1

- Which outgoing interface does R use for a packet with destination 12.1.2.3?

- Which outgoing interface does R use for a packet with destination 12.2.4.5?

2

Solution:

1. 256

2. 12.0.0.0/8 A

12.1.0.0/16 B

12.2.3.0/24 C

3. C

4. B

5. A

6. Sample Question: NAT

Q: In some cases, why may a NAT router need to change TCP sequence number of a packet? Can you provide an example?

Solution:

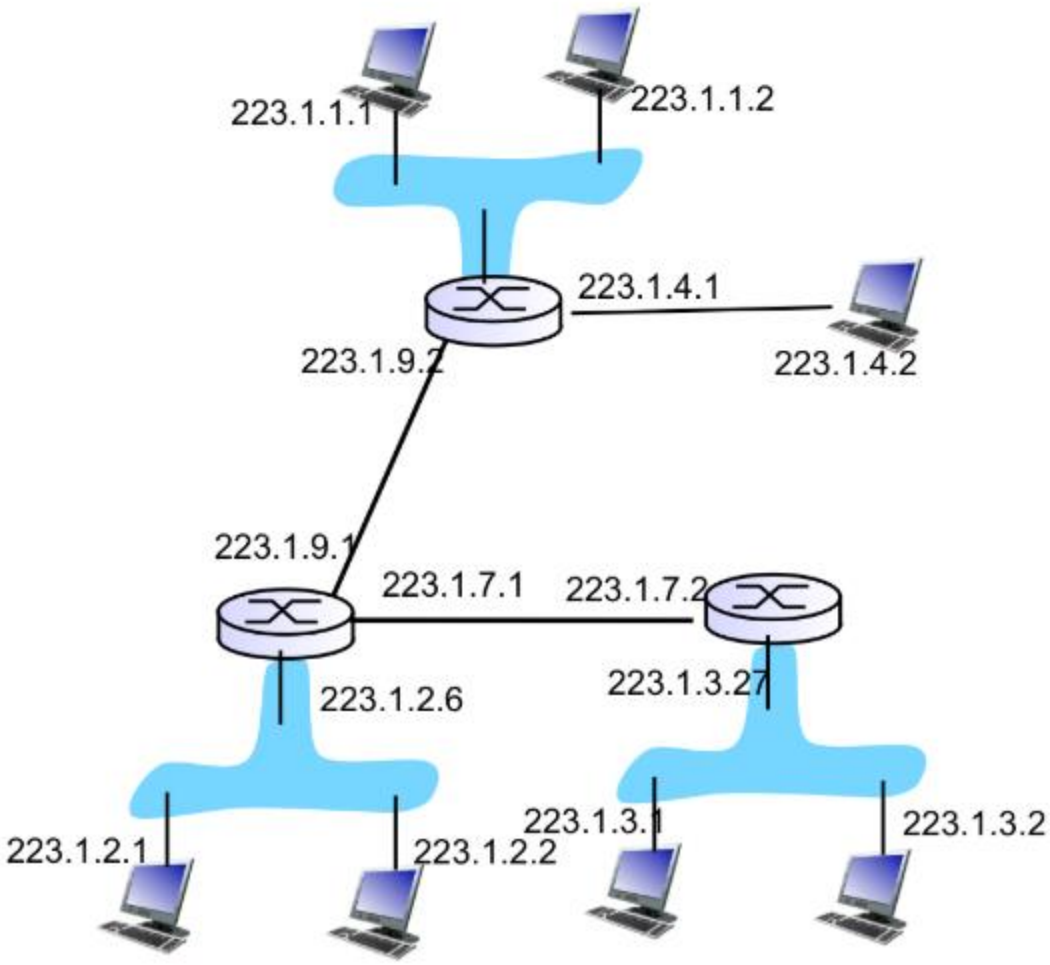
Some applications (such as FTP) may include IP address or Port number in data part. The NAT box peeks in the data and changes the IP / Port number. This change may result in different data length (e.g., For FTP, IP addresses are encoded in ASCII in data) requiring the TCP sequence number to be changed as well.

7. Sample Question: Forwarding

Q:

Complete the forwarding table of all routers shown in the topology below. You can assume that the network portion of all addresses is 24 bits long.

Destination Network	Next Router	# of hops	Interface



Solution: (Assume the missing interface IP for top router is 223.1.1.3)

Destination network	Next Router	# of hops	Interface
Top Router			
223.1.1.0/24	Direct	0	223.1.1.3
223.1.4.0/24	Direct	0	223.1.4.1
223.1.9.0/24	Direct	0	223.1.9.2
223.1.7.0/24	223.1.9.1	1	223.1.9.2
223.1.2.0/24	223.1.9.1	1	223.1.9.2
22.1.3.0/24	223.1.9.1	2	223.1.9.2
Left Router			
223.1.2.0/24	Direct	0	223.1.2.6
223.1.7.0/24	Direct	0	223.1.7.1
223.1.9.0/24	Direct	0	223.1.9.1
223.1.3.0/24	223.1.7.2	1	223.1.7.1
223.1.1.0/24	223.1.9.2	1	223.1.9.1
223.1.4.0/24	223.1.9.2	1	223.1.9.1

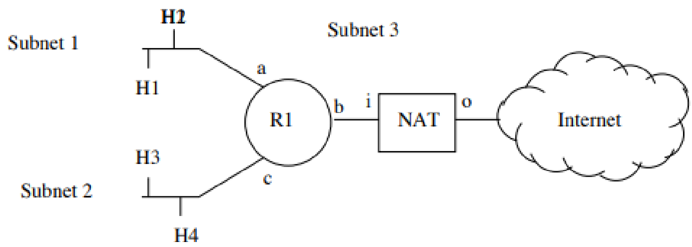
Right Router			
223.1.3.0/24	Direct	0	223.1.3.27
223.1.7.0/24	Direct	0	223.1.7.2
223.1.2.0/24	223.1.7.1	1	223.1.7.2
223.1.9.0/24	223.1.7.1	1	223.1.7.2
223.1.4.0/24	223.1.7.1	2	223.1.7.2
223.1.1.0/24	223.1.7.1	2	223.1.7.2

8. Sample Question: NAT

Q:

Sample Question

Leonard Hofstadter runs a large network at his house and wants to subnet it to separate his 2 work computers (H1 and H2) from the network that controls his toaster (H3) and fridge (H4). He has purchased a NAT box, and divided his network as follows:



His ISP has given him an IP address that he assigns to NAT-o (the WAN side interface of the NAT box). Your job is to assign addresses to the subnets, router interfaces (a, b, c), 4 hosts (H1-H4) and the NAT-i interface. Use the private address range 10.x.x.x.

Link Layer 3

Solution:

Subnet 1& 2 requires 5 IP addresses [3 interface addresses for devices & router, 1 address for network and 1 address for broadcast]. We need at least /29 giving you 2^3 = 8 IP addresses. As the question does not ask for an efficient address allocation, we can use the following simple scheme.

Subnet1: 10.0.0.0/24 >> H1: 10.0.0.2, H2: 10.0.0.3, R1a: 10.0.0.1

Subnet 2: 10.0.1.0/24 >> H3: 10.0.1.2, H4: 10.0.1.3, R1c: 10.0.1.1

Subnet 3: 10.0.2.0/24 >> R1b: 10.0.2.1, NATi: 10.0.2.2

Week 8:

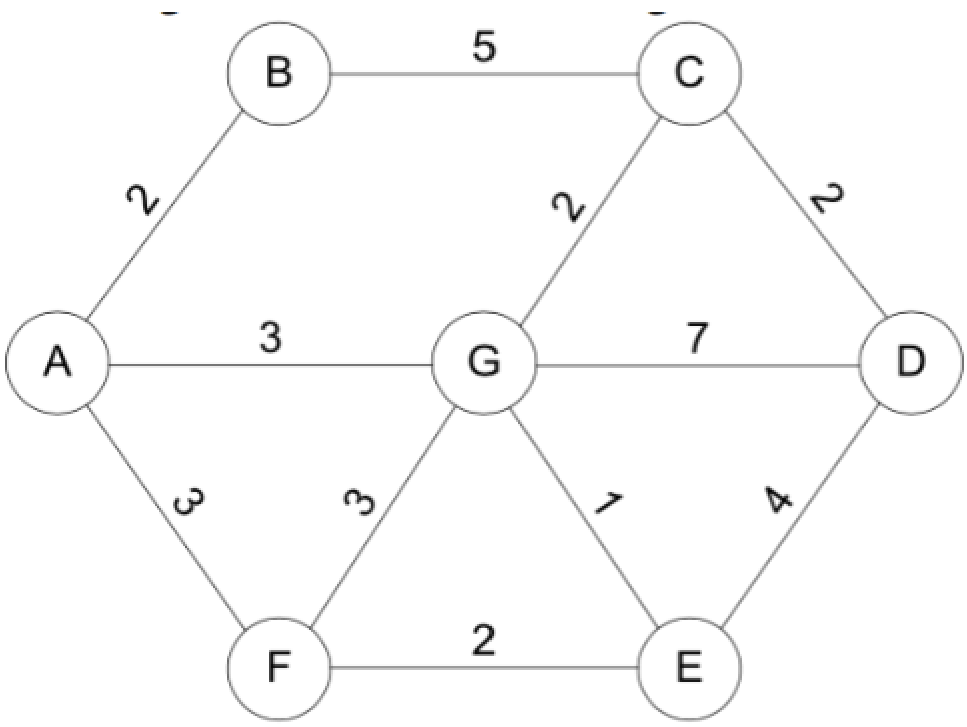
1. Sample Question: Routing

Q: (a) For the network shown below with the given link costs, use Dijkstra's algorithm to determine the shortest path from A to all other nodes

(b) Based on your answer to (a), what would the routing table at A look like?

(c) Consider the distance-vector routing algorithm applied to this network.

(i) Show the initial forwarding tables for nodes A, B, F and G (i.e. when each node is only aware of its immediate neighbours). (ii) Show the forwarding table for node A after the first exchange of forwarding information between neighbours.



Solution:

a)

	B	C	D	E	F	G
A	2,A	inf	inf	inf	3,A	3,A
AB		7,B	inf	inf	3,A	3,A
ABF		7,B	inf	5,F		3,A
ABFG		5,G	10,G	4,G		
ABFGE		5,G	8,E			
ABFGEC			7,C			
ABFGECD						

b)

RT at A	
Dest	NextHop
B	direct
F	direct
G	direct
C	G
D	G
E	G

c)

Node A				
	B	F	G	mindist
B	2	inf	inf	2
F	inf	3	inf	3
G	inf	inf	3	3

Node B			
	A	C	mindist
A	2	inf	2
C	inf	5	5

Node F				
	A	E	G	mindist
A	3	inf	inf	3
E	inf	2	inf	2
G	inf	inf	3	3

Node G						
	A	C	D	E	F	mindist
A	3	inf	inf	inf	inf	3
C	inf	2	inf	inf	inf	2
D	inf	inf	7	inf	inf	7
E	inf	inf	inf	1	inf	1
F	inf	inf	inf	inf	3	3

Node A				
	B	F	G	mindist
B	2	inf	inf	2
F	inf	3	6	3
G	inf	6	3	3
C	7	inf	5	5
D	inf	inf	10	10
E	inf	5	4	4

2. Sample Question: Parity Check

Q: Give an example of a 4-bit error that would not be detected by 2-dimensional parity. What is the general set of circumstances under which 4-bit errors will be undetected.

Solution:

Example:

1 0 1 0 1 1 0 0

1 0 0 1 0 0 1 1

0 0 0 1 0 1 0 0

0 0 1 0 1 0 1 1

Flipped bits that can go undetected:

1 0 1 0 1 1 0 0

1 1 1 1 0 0 1 1

0 1 1 1 0 1 0 0

0 0 1 0 1 0 1 1

General Rule: There must be even Nos of bits that must be changed in two rows and these changes must correspond to the same columns in the two rows.

3. Sample Question: CRC

Q: Suppose a sender and receiver use Cyclic Redundancy Check (CRC) code for error detection. They agree to use generator 10011. The sender transmits a message and the receiver receives the following message (including CRC bits): 01101011100110. Find whether the message was received correctly or whether it is in error.

Solution:

The remainder of dividing the received message and the generator should be 0 if there has been no error. The remainder comes out to be 1100 so the received packet is declared as corrupted.

Week 9:

1. Sample Question: ARP

Q: Nodes 1, 2, 3 and 4 on an Ethernet LAN have Ethernet addresses E1, E2, E3 and E4, and IP addresses I1, I2, I3 and I4, respectively. Assume that the nodes are connected via a star topology via a central Ethernet switch (typical star topology as shown in the lecture slides) All nodes maintain an ARP cache and respond to ARP queries immediately. Unless refreshed, ARP cache entries are timed out and removed if more than 15 minutes old. At time t = 0, assume that all the ARP caches are empty. After ARP responses are received subsequent to each of the following ARP queries, state which nodes have an Ethernet-IP address mapping for which other nodes in their ARP caches.

1. At time t = 5 minutes, node 1 broadcasts an ARP query for IP address I3, and node 2 broadcasts an ARP query for IP address I1.
2. At time t = 15 minutes, node 3 broadcasts an ARP query for IP address I2.
3. At time t = 25 minutes, node 4 broadcasts an ARP query for IP address I2.

Solution:

At time t = 5min

N1 has (I2:E2 TTL 15, I3:E3 TTL 15)

N2 has (I1:E1 TTL 15)

N3 has (I1:E1 TTL 15)

N4 has NIL.

At time t = 15min

N1 has (I2:E2 TTL 5, I3:E3 TTL 5)

N2 has (I1:E1 TTL 5, I3:E3 TTL 15)

N3 has (I1:E1 TTL 5, I2:E2 TTL 15)

N4 has NIL

At time t = 25min

N1 has NIL

N2 has (I3:E3 TTL 5, I4:E4 TTL 15)

N3 has (I2:E2 TTL 5)

N4 has (I2:E2 TTL 15)

2. Sample Question: Hidden Terminal

Q: What is the "hidden" terminal problem? How is it solved in 802.11?

Solution:

Hidden terminal problem refers to a scenario whereby two nodes that cannot hear each other attempt to communicate to an intermediate node that can hear both of them. This means carrier sensing does not detect whether a transmission to the intermediate node will face any collision as the channel is found clear by both the nodes.

It is solved using Using CSMA/CA with RTS/CTS in 802.11.

3. Sample Question: Co-Existence of WIFi

Q: There are two coffee shops next to each other. Two different ISPs (ISP-1 and ISP-2) each with a unique SSID provide 802.11 based Wi-Fi access to the coffee shops. Accidentally, both the ISPs configure their APs to operate on channel 11 not aware that their wireless coverage almost completely overlaps being too close to each other. Now consider two wireless hosts, one associated with ISP-1 and the other associated with ISP-2, that try to access the Internet simultaneously. Will communication be at all possible for the wireless hosts? Answer YES or NO and BRIEFLY explain why.

Solution:

Yes. Communication can still be possible because of 802.11 using CSMA/CA on a shared medium. RTS-CTS will be used between each hosts and APs to avoid collisions, or any collisions will be handled with binary exponential back off.

Week 10:

"Security is not included in the exam"