CSCE 5580: Computer Network

Homework-2

Name: Kishan Kumar Zalavadia

EUID: 11685261

---

1. **We have said that an application may choose UDP for a transport protocol because UDP offers finer application control (than TCP) of what data is sent in a segment and when.**
   a. **Why does an application have more control over what data is sent in a segment?**

   Ans: In UDP, an application has more control over what data is sent in a segment than in TCP because it is mainly a connectionless protocol, which means that to send messages, the sender need not make any connection with the receiver. UDP is also a stateless protocol, which means the server need not maintain any state related to the client. UDP does not require the data to be structured or segmented. It totally depends on the application whether it wants a large or small amount of data in the UDP segment. Whereas the TCP transport layer handles segmentation and recombining of the data in trying to optimize the recommendation. UDP does not provide any error checking or require any acknowledgment, so it can focus on data transfer other than reliability.

   b. **Why does an application have more control over when the segment is sent?**

   Ans: An application has more control over the segment, which is sent in UDP because UDP is a connectionless and stateless protocol that does not require any connection establishment before communication or any state information stored in the server. UDP also does not require any acknowledgment. In TCP, the transport layer handles the flow control, congestion control, and re-transmission of packets, whereas, in UDP, the transmission is entirely on the application. The application can choose to send UDP segmentation immediately, with delay, or batch multiple application-level messages into a single UDP check without the interference of the transport layer.

---

2. **Do a bit of research: Describe SYN flood attacks and how SYN cookies are used to help alleviate the problem. Use your own words rather than cut and paste from Wikipedia or other source.**

Ans:

SYN flood attacks:

SYN flood is a form of denial-of-service attack that exploits the TCP three-way handshake protocol. The aim of the SYN flood attack is to exhaust the server resources. by sending requests continuously so that the server is not available to legitimate users.

In a three-way handshake protocol, first, the client sends an SYN packet to initialize the connection after that, the server responds with a SYN-ACK packet, which indicates that the server is ready to make a connection. Finally, the client responds with an ACK packet to complete the handshake. Now, the connection is successful, and the client-server communicates with each other.

In a SYN attack, the attacker keeps sending the SYN packet to the server but never finalizes any connection. To establish a connection, the server keeps waiting for the acknowledgment from the client(attacker), but the attacker never sends the SYN acknowledgment, so the server keeps waiting and jumping over multiple connection requests made by the attacker, which eventually exhausts all the resources of the server.

SYN Cookies:

SYN cookies are used to mitigate the SYN flood attack. When a server gets a connection request, instead of allocating the resources immediately, it will assign the request a unique number, which is a sequence number called SYN cookie. SYN cookie is a hash that contains information about the connection, for example, client IP address, port number, and other parameters. When the client receives SYN acknowledgment along with the SYN cookie, it will send an acknowledgment to the server, which will represent that the client genuinely wants to make or complete a connection to the server, and then verify the SYN cookie to validate the client's request before allocating the resources to the communication. In this way, the SYN cookie allows the server to allocate resources only to genuine users.

3. What is the issue with not using sequence numbers in the TCP/IP protocol? List at least three example scenarios. Additionally, respond as to whether a reliable data transfer may be implemented on other layers of the TCP/IP protocol stack and issues that may arise from each.

Ans: Sequence numbers in TCP/IP are mainly used to ensure data integrity, reliable data, data transfer, and correct ordering. All the packets must be received in the correct order in which it is sent without sequence number. There can be several issues that can impact communication.

1. Packet loss: To identify the missing packets, we can use sequence numbers. Packet loss can happen due to congestion or other issues. To identify the missing packets, we can use sequence numbers. If there is no sequence number and If some part of a packet or file is lost, the receiver will have an incomplete or corrupted file without knowing that some file has been lost.

2. Duplicate packets: Due to some error, some packets can be re-transmitted, causing the packets to be duplicated. In this case, the sequence number will be useful to identify the duplicate packets so that they can be discarded. Without sequence numbers, you would not be able to recognize the duplicate packets, which will lead to unnecessary packets.

3. Packet re-ordering: network the data is divided into packets, and each packet can reach its destination through different parts, hence arriving in different order. Without the sequence number, we would not be able to recognize the correct order of the packets, and hence, the receiver will not be able to reassemble the package in the correct order, which will lead to ordered or scrambled data. For example, the message "Hello World" could arrive as "World Hello," which is a serious issue.

TCP provides allowable data transfer through sequence number and other mechanisms. We can also implement reliability on other layers, but each layer has its challenges.

1. Application layer: to implement reliability on the application layer, the application itself must handle re-transmission, acknowledgment, and packet ordering. This would make the application complex. It would also lead to inefficiency of the application because every application would need its own version of reliability control.

2. Transport layer: TCP provides a reliable data transfer whereas UDP doesn't provide built-in reliability, so all the applications that are using UDP would need to have their own protocols to manage acknowledgment, transmission, and sequence numbers. This will allow the application to maintain a low level of communication and control to implement such reliability. It can be challenging and error-prone because each application can have its own way of handling reliability, which may lead to inconsistency.

3. Network layer: to ensure reliable data transfer through the network layer, it will require redesigning the Internet protocol (IP). If we add reliability to internal protocol, it could be complicated for all network devices, such as routers and switches, because they would need to track packet sequence, manage re-transmission, and handle acknowledgments, which would reduce the efficiency due to its complexity.

4. **Suppose that you have been assigned the 138.90.160.0/19 network address block, and you need to establish four equally-sized subnets from this block.**
   1) List the network addresses for each of the four subnets.
   2) How many hosts can be in each subnet?
   3) List the range of host IP addresses for the first subnet.

Ans:

4) Given n/w address block: 138.90.160.0/19

A /19 network has a subnet mask of 255.255.224.0

⇒ 32 - 19 = 13 host bits.

To divide /19 network into 4 equal subnets, we need to borrow 2 more bits => /21

A /21 network has 32 - 21 = 11 host bits. So, each subnet have

$2^{11} - 2 = 2046$ hosts.

First subnet : 138.90.168.0/21

Second subnet : 138.90.168.0/21 (160.0 + 8.0)

Third subnet : 138.90.176.0/21 (160.0 + 16.0)

Fourth subnet : 138.90.184.0/21 (160.0 + 24.0)

Range of Host IP address for the first subnet:

N/w address : 138.90.160.0

Brodcast address: 138.90.167.255

Range of host IP addresss : 138.90.160.1 to 138.90.167.254.

Answers:

1) 138.90.160.0/21

138.90.168.0/21

138.90.176.0/21

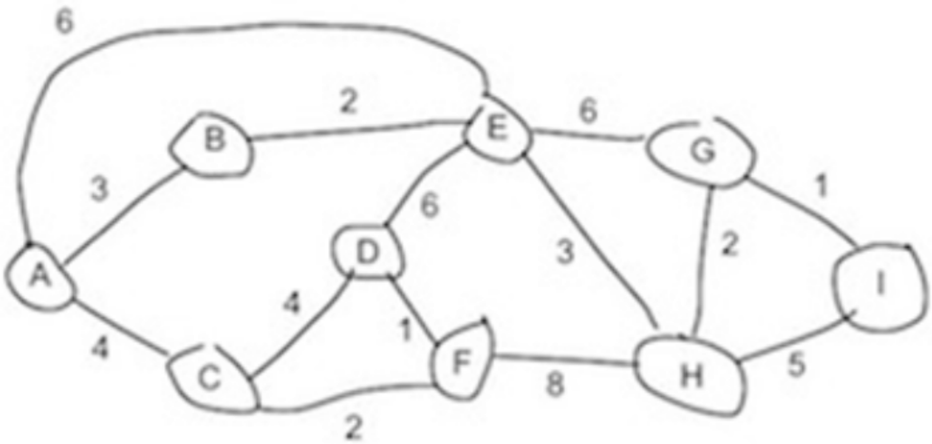
138.90.184.0/21

2) No. of hosts per subnet: 2046 hosts

3) Range of host IP addresses for the first subnet:

138.90.160.1 to 138.90.167.254.

5. **Given the following computer networks,**

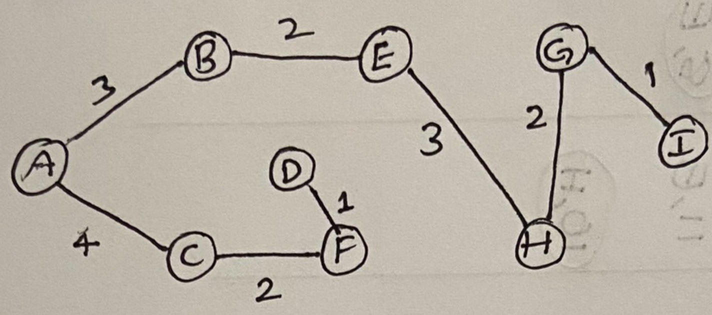


1) **Use Dijkstra's shortest path algorithm to compute the shortest path from A to all network nodes. Show work/table. Show the final next hop routing table/forwarding table for A.**

Ans:

Step	N'	B D(B),P(B)	C D(C),P(C)	D D(D),P(D)	E D(E),P(E)	F D(F),P(F)	G D(G),P(G)	H D(H),P(H)	I D(I),P(I)
0	A	3,A	4,A	∞	6,A	∞	∞	∞	∞
1	AB		4,A	∞	5,B	∞	∞	∞	∞
2	ABC			8,C	5,B	6,C	∞	∞	∞
3	ABCE			8,C		6,C	11,E	8,E	∞
4	ABCEF			7,F			11,E	8,E	∞
5	ABCEFD						11,E	8,E	∞
6	ABCEFDH						10,H		13,H
7	ABCEFDHG								11,G

Least cost path from A :



Forwarding table for A :

Destination	Outgoing Link
B	(A,B)
C	(A,C)
D	(A,C)
E	(A,B)
F	(A,C)
G	(A,B)
H	(A,B)
I	(A,B)

2) **Use the Bellman-Ford algorithm to show the shortest distance from each node to A. Show work/table. Show final next hop routing table values for each node to A. (Assuming that the algorithm begins with each node knowing only the costs to its immediate neighbors)**

Ans:

5) 2) <u>Bellman–Ford Algorithm</u>

$t=0$

DV in A
$D_A(B) = 3$
$D_A(C) = 4$
$D_A(D) = \infty$
$D_A(E) = 6$
$D_A(F) = \infty$
$D_A(G) = \infty$
$D_A(H) = \infty$
$D_A(I) = \infty$

$t=1$, B recives DVs from A, E

DV in B	update 1
$D_B(A) = 3$	3
$D_B(B) = 0$	0
$D_B(C) = \infty$	7
$D_B(D) = \infty$	8
$D_B(E) = 2$	2
$D_B(F) = \infty$	∞
$D_B(G) = \infty$	8
$D_B(H) = \infty$	5
$D_B(I) = \infty$	∞

DV in E	Update 4
$D_E(A) = 6$	5
$D_E(B) = 2$	2
$D_E(C) = \infty$	9
$D_E(D) = 6$	6
$D_E(E) = 0$	0
$D_E(F) = \infty$	7
$D_E(G) = 6$	5
$D_E(H) = 3$	3
$D_E(I) = \infty$	7

$D_B(A) = \min\{C_{B,A} + D_A(A), C_{B,E} + D_E(A)\} = \min\{3, 2+6\} = 3$

$D_B(C) = \min\{C_{B,A} + D_A(C), C_{B,E} + D_e(C)\} = \min\{3+4, 2+\infty\} = 7$

$D_B(D) = \min\{C_{B,A} + D_A(D), C_{B,E} + D_E(D)\} = \min\{3+\infty, 2+6\} = 8$

$D_B(E) = \min\{C_{B,A} + D_A(E), C_{B,E} + D_E(E)\} = \min\{3+6, 2+0\} = 2$

$D_B(F) = \min\{C_{B,A} + D_A(F), C_{B,E} + D_E(F)\} = \min\{3+\infty, 2+\infty\} = \infty$

$D_B(G) = \min\{C_{B,A} + D_A(G), C_{B,E} + D_E(G)\} = \min\{3+\infty, 2+6\} = 8$

$D_B(H) = \min\{C_{B,A} + D_A(H), C_{B,E} + D_E(H)\} = \min\{3+\infty, 2+3\} = 5$

$D_B(I) = \min\{C_{B,A} + D_A(I), C_{B,E} + D_E(I)\} = \min\{3+\infty, 2+\infty\} = \infty.$

Updete 1 Complete.

=2

C recives DVs from A, D, F

DV in D	update 3
$D_D(A) = \infty$	8
$D_D(B) = \infty$	8
$D_D(C) = 4$	3
$D_D(D) = 0$	0
$D_D(E) = 6$	6
$D_D(F) = 1$	1
$D_D(G) = \infty$	12
$D_D(H) = \infty$	9
$D_D(I) = \infty$	∞

DV in F	Update 5
$D_F(A) = \infty$	6
$D_F(B) = \infty$	9
$D_F(C) = 2$	2
$D_F(D) = 1$	1
$D_F(E) = \infty$	7
$D_F(F) = 0$	0
$D_F(G) = \infty$	10
$D_F(H) = 8$	8
$D_F(I) = \infty$	13

$D_c(\cancel{B}A) = \min\{C_{C,A} + D_A(A),\ C_{C,D} + D_D(A),\ C_{C,F} + D_F(A)\}$

$= \min\{4+0,\ 4+\infty,\ 2+\infty\} = 4$

$D_c(B) = \min\{4+3,\ 4+\infty,\ 2+\infty\} = 7$

$D_c(D) = \min\{4+\infty,\ 4+0,\ 2+1\} = 3$

$D_c(E) = \min\{4+6,\ 4+6,\ 2+\infty\} = 10$

$D_c(F) = \min\{4+\infty,\ 4+1,\ 2+0\} = 2$

$D_c(G) = \min\{4+\infty,\ 4+\infty,\ 2+\infty\} = \infty$

$D_c(H) = \min\{4+\infty,\ 4+\infty,\ 2+8\} = 10$

$D_c(I) = \min\{4+\infty,\ 4+\infty,\ 2+\infty\} = \infty$

Update 2 complete.

Dv in C	update 2
$D_c(A) = 4$	4
$D_c(B) = \infty$	7
$D_c(C) = 0$	0
$D_c(D) = 4$	3
$D_c(E) = \infty$	10
$D_c(F) = 2$	2
$D_c(G) = \infty$	∞
$D_c(H) = \infty$	10
$D_c(I) = \infty$	∞

t=3

D recives DVs from C, E, F

$D_D(A) = \min\{4+4, 6+6, 1+\infty\} = 8$

$D_D(B) = \min\{4+7, 6+2, 1+\infty\} = 8$

$D_D(C) = \min\{4+0, 6+\infty, 1+2\} = 3$

$D_D(E) = \min\{4+10, 6+0, 1+\infty\} = 6$

$D_D(F) = \min\{4+2, 6+\infty, 1+0\} = 1$

$D_D(G) = \min\{4+\infty, 6+\cancel{\infty}\,6, 1+\infty\} = 12$

$D_D(H) = \min\{4+10, 6+3, 1+8\} = 9$

$D_D(I) = \min\{4+\infty, 6+\infty, 1+\infty\} = \infty$

Update 3 complete.

t=4

E recive DVs from A, B, D, H, G.

DV in G	update 6
$D_G(A) = \infty$	11
$D_G(B) = \infty$	8
$D_G(C) = \infty$	15
$D_G(D) = \infty$	12
$D_G(E) = 6$	6
$D_G(F) = \infty$	10
$D_G(G) = 0$	0
$D_G(H) = 2$	2
$D_G(I) = 1$	1

DV in H	Update 7
$D_H(A) = \infty$	8
$D_H(B) = \infty$	5
$D_H(C) = \infty$	12
$D_H(D) = \infty$	9
$D_H(E) = \infty$	3
$D_H(F) = 8$	8
$D_H(G) = 2$	2
$D_H(H) = 0$	0
$D_H(I) = 5$	3

$D_E(A) = \min\{6+0, 2+3, 6+8, 3+\infty, 6+\infty\} = 5$

$D_E(B) = \min\{6+3, 2+0, 6+8, 3+\infty, 6+\infty\} = 2$

$D_E(C) = \min\{6+4, 2+\cancel{\infty}\,7, 6+3, 3+\infty, 6+\infty\} = 9$

$D_E(D) = \min\{6+\infty, 2+\cancel{\infty}\,8, 6+0, 3+\infty, 6+\infty\} = 6$

$D_E(F) = \min\{6+\infty, 2+\infty, 6+1, 3+8, 6+\infty\} = 7$

$D_E(G) = \min\{6+\infty, 2+8, 6+12, 3+2, 6+0\} = 5$

$D_E(H) = \min\{6+\infty, 2+5, 6+9, 3+0, 6+2\} = 3$

$D_E(I) = \min\{6+\infty, 2+\infty, 6+\infty, 3+5, 6+1\} = 7$

Update 4 complete.

$t=5$

F receive DV's from C, D, H.

$D_F(A) = \min\{2+4, 1+8, 8+\infty\} = 6$

$D_F(B) = \min\{2+7, 1+8, 8+\infty\} = 9$

$D_F(C) = \min\{2+0, 1+3, 8+\infty\} = 2$

$D_F(D) = \min\{2+3, 1+0, 8+\infty\} = 1$

$D_F(E) = \min\{2+10, 1+6, 8+\infty\} = 7$

$D_F(G) = \min\{2+\infty, 1+12, 8+2\} = 10$

$D_F(H) = \min\{2+10, 1+9, 8+0\} = 8$

$D_F(I) = \min\{2+\infty, 1+\infty, 8+5\} = 13.$

Update 5 complete

t=6

G recives DVs from E, H, I.

(12)

D_x in I	Update 8
$D_I(A) = \infty$	12
$D_I(B) = \infty$	9
$D_I(C) = \infty$	16
$D_I(D) = \infty$	13
$D_I(E) = \infty$	7
$D_I(F) = \infty$	11
$D_I(G) = 1$	1
$D_I(H) = 5$	3
$D_I(I) = 0$	0

$D_G(A) = \min\{6+5, 2+\infty, 1+\infty\} = 11$

$D_G(B) = \min\{6+2, 2+\infty, 1+\infty\} = 8$

$D_G(C) = \min\{6+9, 2+\infty, 1+\infty\} = 15$

$D_G(D) = \min\{6+6, 2+\infty, 1+\infty\} = 12$

$D_G(E) = \min\{6+0, 2+\infty, 1+\infty\} = 6$

$D_G(F) = \min\{6+7, 2+8, 1+\infty\} = 10$

$D_G(H) = \min\{6+3, 2+0, 1+5\} = 2$

$D_G(I) = \min\{6+7, 2+5, 1+0\} = 1$

Update 6 complete.

$t = 7$

H recives DV's from E, F, G, I

$D_H(A) = \min\{3+5, 8+6, 2+11, 5+\infty\} = 8$

$D_H(B) = \min\{3+2, 8+9, 2+8, 5+\infty\} = 5$

$D_H(C) = \min\{3+9, 8+2, 2+15, 5+\infty\} = 12$

$D_H(D) = \min\{3+6, 8+1, 2+12, 5+\infty\} = 9$

$D_H(E) = \min\{3+0, 8+7, 2+6, 5+\infty\} = 3$

$D_H(F) = \min\{3+7, 8+0, 2+10, 5+\infty\} = 8$

$D_H(G) = \min\{3+5, 8+10, 2+0, 5+1\} = 2$

$D_H(I) = \min\{3+7, 8+13, 2+1, 5+0\} = 3$

Update 7 complete.

$t = 8$

I recives DV's from G, H.

$D_I(A) = \min\{1+11; 5+8\} = 12$

$D_I(B) = \min\{1+8, 5+5\} = 9$

$D_I(C) = \min\{1+15, 5+12\} = 16$

$D_I(D) = \min\{1+12, 5+9\} = 13$

$D_I(E) = \min\{1+6, 5+3\} = 7$

$D_I(F) = \min\{1+10, 5+8\} = 11$

$D_I(G) = \min\{1+0, 5+2\} = 1$

$D_I(H) = \min\{1+2, 5+0\} = 3$

Update 8 complete.

6. **Suppose a router interconnects three subnets. All of the subnets will have the prefix 211.1.16/24. Subnet 1 is required to support 62 interfaces, Subnet 2 will need to support 95 interfaces, and Subnet 3 will need to support up to 16 interfaces. Provide three network addresses of the form a.b.c.d/x that satisfy these constraints.**

Ans:

6) Given network: 211.1.16.0/24.

A /24 subnet provides 8 bits for hosts,

$2^8 - 2 = 256 - 2 = 254$ usable IP addresses.

Subnet 1: Needs 62 interfaces.

This requirs 6 host bits, since $2^6 = 64$.

[Note: $2^5 = 32 < 62$, so the next is 2^6]. ↳ /26 Subnet mask

Subnet 2: Needs 95 interfaces.

This requires 7 host bits, since $2^7 = 128$

↳ /25

Subnet 3: Needs 16 interfaces.

This requires 5 host bits, since $2^5 = 32$

↳ /27

Lets Allocate subnet addresses.

Starting with 211.1.16.0/24

Subnet 2: Needs /25 subnets.

N/w address: 211.1.16.0/25

This supports 126 hosts

Subnet 1: Needs /26 subnets.

N/w address: 211.1.16.128/26, supports 62 hosts

Subnet 3: Needs /27 subnets.

N/w address: 211.1.16.192/27, supports 30 hosts.

7. **You use Dijkstra's algorithm for finding the shortest path in a graph between all pairs of nodes. Can you use the same algorithm and change it in some way to find the longest path?**

Ans: The actual Dijkstra algorithm is used to find the shortest path in the graph with non-negative edge weights, so it minimizes the path length to travel the nodes. Unfortunately, the Dijkstra algorithm is not suitable for finding the longest path for all kinds of graphs. Dijkstra algorithm uses a priority queue, which will always expand the shortest path note 1^{st}, and if we try to reverse the algorithm, it won't work reliably because of cycles. Any path with cycles can infinitely be extended by looping through the cycle, hence making the longest path infinite.

Dijkstra's algorithm can find the longest path only for certain types of graphs. For example, Acyclic directed graph. In a Direct acyclic graph, there are no cycles present, so it is feasible to find the longest path. We can reverse the weight of the edges by multiplying by one and using the topological sort approach to find the longest path.

---

8. **Consider a general topology and a synchronous version of the distance-vector algorithm. Suppose that at each iteration, a node exchanges its distance vectors with its neighbors and receives their distance vectors. Assuming the algorithm begins with each node knowing only the costs to its immediate neighbors, what is the maximum number of iterations required before the distributed algorithm converges? Justify your answer**

Ans: First, let us look at the things that are given. Nodes only know the cost of immediate nodes in each iteration. Nodes exchange distance vectors with their neighbors and all nodes exchange information simultaneously. We want to find out the maximum number of iterations needed for convergence.

For every iteration, the information about the parts propagates only one hop. All the nodes already know about the destinations that are one hop away in each iteration. For iteration-1, nodes learn about the other nodes that are one hop away from it, which is already known, for the second iteration, the nodes learn about the nodes that are two hops away from it, similar to the third-iteration, nodes learn about the other nodes, which are three hops away from it and so on.

The maximum number of iterations needed will be equal to the longest shortest path between two nodes in the network. After an iteration, where N is the length of the longest shortest path, all nodes will have information about all possible destinations. No additional iterations will be needed because no further shortest path can be discovered. So, the **maximum iteration is equal to N-1**, where N is the diameter of the network. We subtract one because all the notes already know about their immediate nodes.

---

9. **Suppose you purchase a wireless router and connect it to your cable modem. Also, suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also, suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?**

Ans: When we connect multiple devices to a single wireless router with only one IP address assigned by the ISP, the router uses network address translation in NAT to manage these connections.

The ISP will provide a single assigned IP address to the wireless router. The wireless router can manage a private network within the home to allow each device to communicate within the network and to the Internet. The router will assign a unique private IP address to each device in the local network. This private IP address is not access accessible outside the home network.

When the PC connects to the router using Wi-Fi, the router's DHCP server will assign unique private IP addresses to each device as follows.

PC 1: 192.1 68.0.2

PC 2: 192.1 68.0.3

PC 3: 192.1 68.0.4

PC 4: 192.1 68.0.5

PC 5: 192.1 68.0.6

NAT will map multiple private IP addresses to a single public IP address. NAT is useful because it enables all five pieces to share a single public IP address, which is provided by the ISP. When any device present in the home network initializes a connection to the Internet, NAT converts the device's private IP address to the public IP address. NT will also keep track of communication so that the initiating device is correct correctly. NAT will also provide security by hiding private IP addresses from the external network.