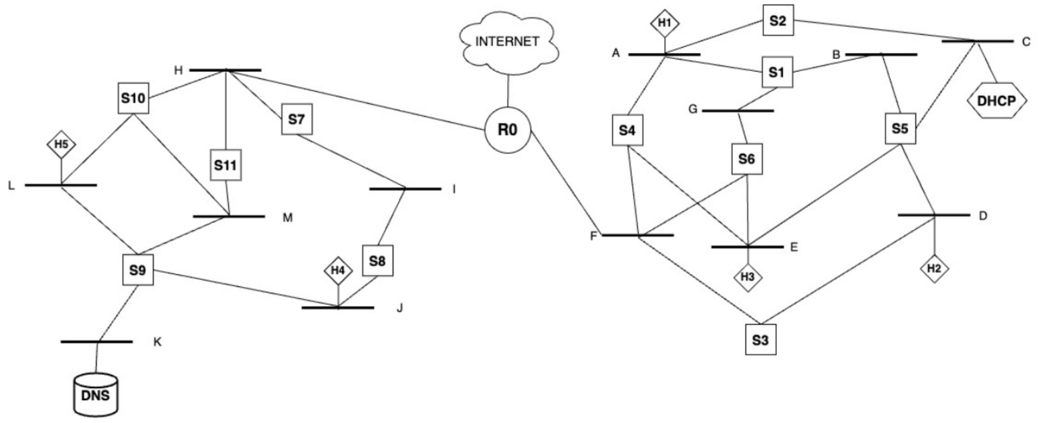
Ex1 — Introduction to Networks

Eran Ston (206704512)   Oded Vaalany (208230474)

June 30, 2024

1 Question 1

Given the following network graph:



1.1 The end unit H5 connects to the left network. Within it LAN there is noe DHCP server explain how the end unit H5 can get an IP address automaticlly.

1.1.1 Elaborate on the needed configuration such the end unit H5 will be able to get an IP address automatically.

Firstly we need to configure R0 to transfer DHCP requests to the other network. and vice versa.

1.1.2 named the massages that are used when H5 connects and request an IP address.

- DHCP Discover in the left network.
- R0 forwards the DHCP Discover to the right network.
- DHCP Offer in the right network.
- R0 forwards the DHCP Offer to the left network.
- DHCP Request in the left network.
- R0 forwards the DHCP Request to the right network.
- DHCP Ack in the right network.
- R0 forwards the DHCP Ack to the left network.
- DHCP Ack in the left network.

1.2 Let's assume that unit H3 activated for the first time, and it want to send one massage to www.huji.ac.il (IP: 123.4.5.6) in the internet. Fill the table with all the massages sends.

The network	Source IP	Destination IP	Source MAC	Destination MAC	Protocol	Massage Type	Explain
Right	0.0.0.0	255.255.255.255	H3	Brodcast	DHCP	DHCP Discover	BC3
Right	DHCP	255.255.255.255	DHCP	Brodcast	DHCP	DHCP Offer	BC3
Right	0.0.0.0	255.255.255.255	H3	Brodcast	DHCP	DHCP Request	BC3
Right	DHCP	255.255.255.255	DHCP	Brodcast	DHCP	DHCP ACK	BC3
Right	-	-	H3	R0(1)	ARP	ARP Request - DNS Server	-
Left	-	-	R0(2)	Brodcast	ARP	ARP Request - H3	transfer the request for H3
Left	-	-	DNS	R0(2)	ARP	ARP Response - DNS Server	R0 will transfer this
Right	-	-	R0(1)	H3	ARP	ARP Response - DNS Server	-
Right	H3	DNS	H3	R0(1)	DNS	DNS Request - www.huji.ac.il	-
Left	H3	DNS	R0(2)	DNS	DNS	DNS Request - www.huji.ac.il	-
Left	DNS	H3	DNS	R0(2)	DNS	DNS Response - www.huji.ac.il	-
Right	DNS	H3	R0(1)	H3	DNS	DNS Response - www.huji.ac.il	-
Right	H3	123.4.5.6	H3	R0(1)	HTTP	HTTP Request - www.huji.ac.il	-
Right	123.4.5.6	H3	R0(1)	H3	HTTP	HTTP Response - www.huji.ac.il	-

2 Question 2

The parts in this questions are not depends on each other.

2.1 Part 1

2.1.1 subquestion 1

- Host send Request to local DNS server.
- Local DNS server send Request to Root DNS server.
- Root DNS server Response to Local DNS server the location of '.com' DNS server.
- Local DNS server send Request to '.com' DNS server.
- '.com' DNS server Response to Local DNS server the location of 'ns1.drekflix.com' DNS server.
- Local DNS server send Request for 'movies.local.drekflix.com' from 'ns1.drekflix.com' DNS server.
- 'ns1.drekflix.com' DNS server Response to Local DNS server the IP address of 'ns1.israel.drekflix.com'.
- Local DNS server send Request for 'movies.local.drekflix.com' from 'ns1.israel.drekflix.com' DNS server.
- 'ns1.israel.drekflix.com' DNS server Response to Local DNS server the IP address of 'movies.local.drekflix.com'.
- Local DNS server send Response to Host with the IP address of 'movies.local.drekflix.com'.

2.1.2 subquestion 2

When one DNS server received a response from another DNS server and update the response in it chache, the TTL ot this records define by the response DNS server as the Authorative DNS server.

2.1.3 subquestion 3

‘ns1.israel.drekflix.com’ wants to be a load-balancing dns server and therefore it crucial to have a short TTL. The reason is that if the IP address of the server changes, the DNS server needs to update the records in the local DNS resolvers. If the TTL is long, the local DNS resolvers will not update the records and the traffic will not be balanced between the servers. the other dns servers can have a long TTL since they are not the load-balancing servers and the IP address of the servers is not changed frequently.

so we will receive the following TTLs:

- Root DNS server: return records with TTL in days.
- ’.com’ DNS server: return records with TTL in days.
- ‘ns1.drekflix.com’: return records with TTL in days.
- ‘ns1.israel.drekflix.com’: return records with TTL in minutes.

2.2 Part 2

2.2.1 subquestion 1

Since there were a storm that cause dammage to the main DNS server, the DNS server is not available. The DNS server is the one that responsible for the domain ’.mad’. And even tho the Server who host the domain ‘www.brass-monkey-saloon.mad’ in order to resolve this address the Local DNS resolver needs first to resolve the domain ’.mad’. Since the main DNS server is not available the Local DNS resolver can’t resolve the domain ’.mad’ and therefore can’t resolve the domain ‘www.brass-monkey-saloon.mad’.

2.2.2 subquestion 2

The reasone why it takes time after the DNS server dammage until the traffic in Madripor failed is because the cached domain in the local DNS resolvers. The TTL of the records in the local DNS resolvers is not expired yet, and therefore the local DNS resolvers can still resolve the domains end with ’.mad’. Since the resolvers are probabliy local and the storme dammage hit far away from the country, the local DNS resolvers can still resolve the domains.

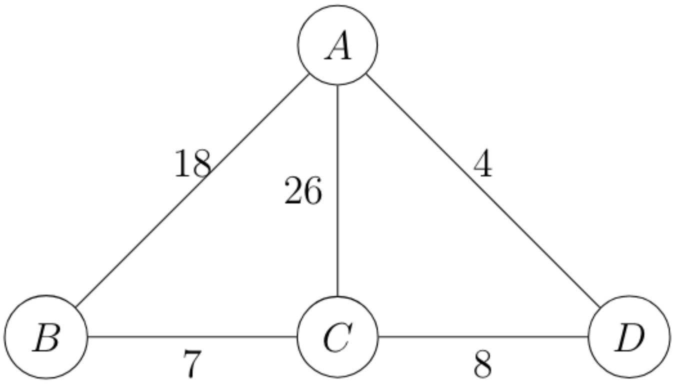
The time it takes until the traffic in Madripor failed is the time it takes until the TTL of the records in the local DNS resolvers expired.

3 Question 3

In Distance Vector Protocol we will split the time to synced time units between all vertexes donates as $t1, t2, t3, \ldots$

- In each time unit t_i each vertex that needs do those steps:
  - Send the distance vector to all neighbors.
  - Receive the distance vector that sent to him.
  - Update its distance vector.
- In Addition, when the weight of an edge changes, the connected vertexes will update their distance vector in the next time unit.

Given the following graph:



3.1 Subquestion 1

assume that the net is stable for t_0 and all the distance vectors are updated. we made a change in edge AB from 18 to 1. we will show the distance vectors of all the vertexes in the t_1, t_2

This is the distance vectors of all the vertexes in the t_0:

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	18	12	4
C	12	7	0	8
D	4	15	8	0

$\Rightarrow$

When the weight of the edge AB changes to 1, A,B imidiately update their distance vector and will send them in the next time unit t_1. Therefore this will be the effect on the changes

Node A	A	B	C	D
A	0	1	8	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	1	0	7	5
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	18	12	4
C	12	7	0	8
D	4	15	8	0

Now we will show the before and after with time unit t_1

After receiving the distance vectors from the neighbors

Node A	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	12	7	0	8

Node C	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	12	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	1	8	4
C	12	7	0	8
D	4	15	8	0

⇒

End of t_1

Node A	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	12	7	0	8

Node C	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	1	8	4
C	12	7	0	8
D	4	5	8	0

Now we will show the before and after with time unit t_2

After Receiving the distance vectors from the neighbors

Node A	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8
D	4	5	8	0

Node B	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8

Node C	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8
D	4	5	8	0

Node D	A	B	C	D
A	0	1	8	4
C	8	7	0	8
D	4	5	8	0

⇒

End of t_2

Node A	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8
D	4	5	8	0

Node B	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8

Node C	A	B	C	D
A	0	1	8	4
B	1	0	7	5
C	8	7	0	8
D	4	5	8	0

Node D	A	B	C	D
A	0	1	8	4
C	8	7	0	8
D	4	5	8	0

3.2 Subquestion 2

Now the edge CD changes from 8 to 32. we will show the distance vectors of all the vertexes in the t_1, t_2, t_3

This is the distance vectors of all the vertexes in the t_0:

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	18	12	4
C	12	7	0	8
D	4	15	8	0

⇒

When the weight of the edge DC changes to 32, D,C imidiately update their distance vector and will send them in the next time unit t_1. Therefore this will be the effect on the changes

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	22
D	4	15	8	0

Node D	A	B	C	D
A	0	18	12	4
C	12	7	0	8
D	4	22	16	0

Now we will show the before and after with time unit t_1

After Receiving the distance vectors from the neighbors

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	22
D	4	22	16	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	22

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	22
D	4	22	16	0

Node D	A	B	C	D
A	0	18	12	4
C	25	7	0	22
D	4	22	16	0

⇒

End of t_1

Node A	A	B	C	D
A	0	18	20	4
B	18	0	7	15
C	25	7	0	22
D	4	22	16	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	22
C	25	7	0	22

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	22
D	4	22	16	0

Node D	A	B	C	D
A	0	18	12	4
C	25	7	0	22
D	4	22	16	0

Now we will show the before and after with time unit t_2

After Receiving the distance vectors from the neighbors

Node A	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	22
D	4	22	16	0

Node B	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	22

Node C	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	22
D	4	22	16	0

Node D	A	B	C	D
A	0	18	20	4
C	25	7	0	22
D	4	22	16	0

⇒

End of t_2

Node A	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	22
D	4	22	16	0

Node B	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	22

Node C	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29
D	4	22	16	0

Node D	A	B	C	D
A	0	18	20	4
C	25	7	0	30
D	4	22	24	0

Now we will show the before and after with time unit t_3

After Receiving the distance vectors from the neighbors

Node A	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29
D	4	22	24	0

Node B	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29

Node C	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29
D	4	22	24	0

Node D	A	B	C	D
A	0	18	20	4
C	25	7	0	29
D	4	22	24	0

⇒

End of t_3

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	24	0

Node B	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29

Node C	A	B	C	D
A	0	18	20	4
B	18	0	7	22
C	25	7	0	29
D	4	22	24	0

Node D	A	B	C	D
A	0	18	20	4
C	25	7	0	29
D	4	22	24	0

3.3 subquestion 3

Now we will show the run using the poisoned reverse technique.

The DV before the change

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node D	A	B	C	D
A	0	18	12	4
C	12	7	0	8
D	4	15	8	0

t_0

⇒

The DV after the change

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8
D	4	15	8	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	12	7	0	8

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	∞
C	25	7	0	30
D	4	15	8	0

Node D	A	B	C	D
A	0	18	∞	4
C	12	7	0	8
D	4	22	32	0

The DV after the change

Node A	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	30
D	4	22	32	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	15
C	25	7	0	30

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	∞
C	25	7	0	30
D	4	22	32	0

Node D	A	B	C	D
A	0	18	∞	4
C	25	7	0	30
D	4	22	32	0

t_1

⇒

The DV after the change

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	15
C	25	7	0	30
D	4	22	32	0

Node B	A	B	C	D
A	0	18	12	4
B	18	0	7	22
C	25	7	0	30

Node C	A	B	C	D
A	0	18	12	4
B	18	0	7	∞
C	25	7	0	30
D	4	22	32	0

Node D	A	B	C	D
A	0	18	∞	4
C	25	7	0	30
D	4	22	32	0

The DV after the change

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	30
D	4	22	32	0

Node B	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	30

Node C	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	30
D	4	22	32	0

Node D	A	B	C	D
A	0	18	25	4
C	25	7	0	30
D	4	22	32	0

t_2

$\Rightarrow$

The DV after the change

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	30
D	4	22	32	0

Node B	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	30

Node C	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	32	0

Node D	A	B	C	D
A	0	18	25	4
C	25	7	0	30
D	4	22	29	0

The DV after the change

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	29	0

Node B	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29

Node C	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	29	0

Node D	A	B	C	D
A	0	18	25	4
C	25	7	0	29
D	4	22	29	0

t_3

$\Rightarrow$

The DV after the change

Node A	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	29	0

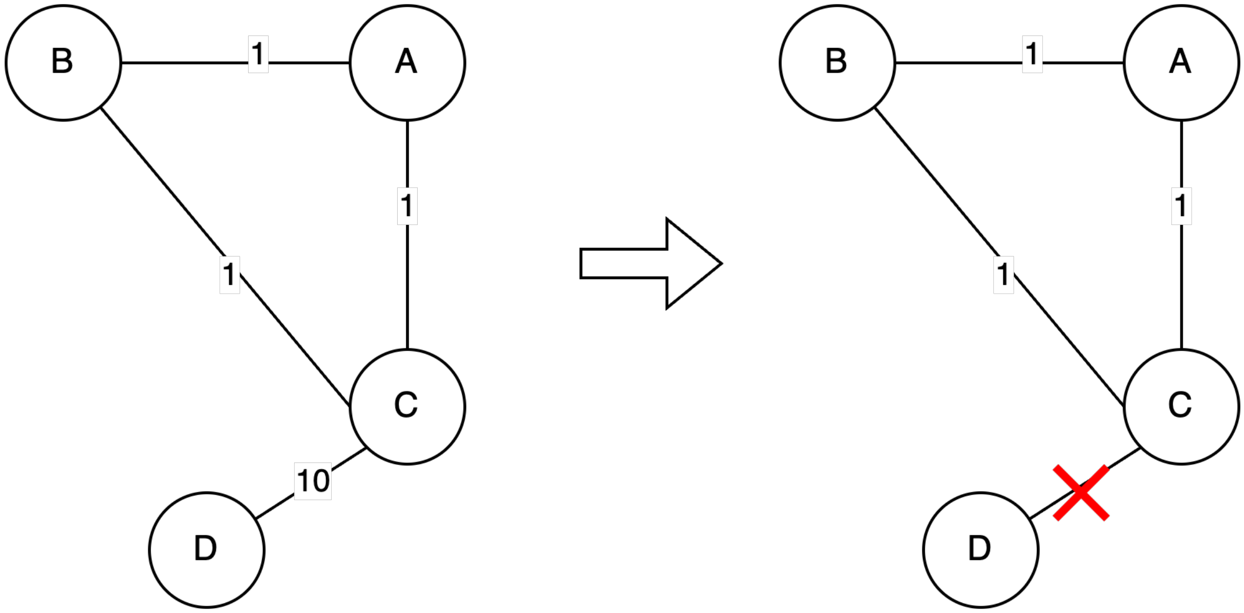
Node B	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29

Node C	A	B	C	D
A	0	18	25	4
B	18	0	7	22
C	25	7	0	29
D	4	22	29	0

Node D	A	B	C	D
A	0	18	25	4
C	25	7	0	29
D	4	22	29	0

3.4 subquestion 4

No, poisoned reverse does not always solve the count to infinity problem in networking. For example If



When link breaks, C marks D as unreachable and reports that to A and B.

- Suppose A learns it first
  - A now thinks best path to D is through B
  - A reports D unreachable to B and a route of cost=3 to C
- C thinks D is reachable through A at cost 4 and reports that to B
- B reports a cost 5 to A who reports new cost to C
- etc...

Note that in this case the loop will occur at each step (it won't halt)