NWEN 302 LAB 2 REPORT

Instructions: Use this editable form to answer the questions and other discussion points. Type your answers in the boxes provided, ensuring that your answers do not go beyond the boxes. Once completed, save the file as `lab2-report.pdf` and submit to the ECS Submission System.

Last Name	Burroughs	First Name	Will
Student ID	300467432		

Lab Objectives

1. To gain understanding of how the Address Resolution Protocol (ARP) works in IPv4.
2. To implement a learning switch.

Objective 1: Understanding Address Resolution Protocol

Q1.1: What triggered the ARP query "`Who has 10.0.0.21? Tell 10.0.0.20`"?

The ARP query is triggered by node n1(10.0.0.20) when it wants to communicate with node n2(10.0.0.21) but does not have n2's MAC adress in its ARP cache.

Q1.2: What is the link layer destination address of the ARP query "`Who has 10.0.0.21? Tell 10.0.0.20`"? Why?

The link layer destination address of the ARP query is the MAC address: (ff:ff:ff:ff:ff:ff). This is because the sender (n1) doesn't know the MAC address of the destination (n2) and it wants to reach all of the devices on the LAN to find the MAC address of the device which has the IP if 10.0.0.21.

Q1.3: Which node generated the ARP reply "`10.0.0.21 is at 00:00:00:aa:00:01`"? Why?

Node n2(10.0.0.0.21) generated the ARP reply because it recived the ARP boradcast.

Q1.4: What is the link layer destination address of the ARP reply "`10.0.0.21 is at 00:00:00:aa:00:01`"? Why did node n3 receive the reply as well?

The link layer destination address of the ARP reply is the MAC address of the sender of the ARP query (n1). Node n3 received the ARP reply because the Ethernet hub floods the ARP reply to all connected devices.

Q1.5: What triggered the ARP query "`Who has 10.0.0.20? Tell 10.0.0.21`"?

After the intial ARP query, n1 knows the MAC address of n2, but n2 does not know the MAC address of n1. Therefore, in order to send an ARP reply query n2 has to send its own ARP query in order to resolve the MAC address of n1.

Q1.6: Compare the delays encountered by the first ping and second ping. Explain why the delays are different.

The delay encountered by the first ping is longer than the delay encountered by the second ping. In the first ping, there is an ARP query sent by n1 to resolve the MAC address of n2, which takes more time. In the second ping, the ARP cache of node n1 already has the MAC address of node n2, so there's no need for an ARP query, resulting in a shorter delay.

Q1.7: How many ARP requests are sent by the sender before it gives up?

There were 3 ARP requests sent from n1 before it gave up.

Running "sysctl net.ipv4.neigh.default.mcast_solicit" in VM console also gives 3.

Therefore, max attempts == 3.

Q1.8: Did you notice any difference in the packets captured at the two nodes? Explain your observation.

In both the ethernet hub & ethernet switch networks, nodes n2 & n3 both received the ARP broadcast requests for both pings(The initial ping as well as the ping for the IP address that didnt exist).

However, in the network that used an ethernet switch n3 did not receive the ping requests/replies.

This is because a network hub will forward all of the ARP requests & pings to all of the devices it is connected to (they are all part of the same collision domain) - it essentially floods the interfaces with frames because it doesnt have any

Discussion 1.1: Unused and stale ARP cache entries are expunged from the cache. Discuss why this is necessary.

Unused & stale ARP cache entries are periodically removed from the ARP cache to help optomize network performance and accuracy. Clearing these caches helps manage memory resources and ensures that devices are using up to date information for the mapping of IP to MAS addresses. It also helps prevent vulnerabilities that might arise from expired or incorrect cache entries (e.g. cache poisioning etc).

Discussion 1.2: Discuss what happens if a node with a different MAC address re-uses an IP address and the old ARP cache entry is not yet expunged.

If a different node re-uses an IP address while an old ARP cache entry is still present, communication involving that node probably wont work as intended / lead to unpredictable behaviour. This could lead to communication and security issues as packets will be sent to the wrong MAC address. This essentially means the data might not be going where the Host was intending it to go, this data will be lost and / or potentially intercepted by someone else.

Objective 2: Implementing a Learning Switch

Q2.1: What are the entries of the table? Provide a brief explanation of the columns.

port no:
This is the port number associated with the corresponding MAC address.

mac addr:
This column shows the MAC address of a network device.

is local?:

Q2.2: What did you notice about the packets captures at the 3 interfaces of n4? Explain.

All three of the interfaces of n4 recived the same packets (Two ARP packets --> Two ICMP packets --> Two ARP packets again).

This is because, as stated in the handout, the switch is by default not configured to learn - This therefore means that it does not update its ARP cache when it learns from traffic, instead it just floods all of the interfaces with the messages just like how a hub would.

Q2.3: What did you notice about the packets captured at the 3 interfaces of n4? Explain.

The interfaces leading to n1(eth0) and n3(eth2) both recieved these frames:

-Ping request
-Ping reply
-ARP request
-ARP request
-ARP reply

Q2.4: What are the new entries in the table?

As well as the entries that were previously there, we have two new entries:

port no	mac addr	is local?	ageing timer
1	00:00:00:aa:00:01	no	18.70
2	00:00:00:aa:00:05	no	18.70

This shows that the switch has learned which ports lead to two particular MAC addresses

Discussion 2.1: Unused and stale entries are expunged from the switch table. Discuss why this is necessary.

Switches have MAC address tables to allow them to perform accurate fram forwarding. However, over time devices can change, which may lead to the entries in these tables being outdated.

Expungingg unused and stale entries is important to optomise memory usage, ensure accurate frame forwarding, reduce network congestion, help maintain the security of frames, and make sure the switch's switch table is concurrent.

Discussion 2.2: Discuss the trade-offs of switch table age/expiration time (i.e. pros and cons of long vs short expiration time).

Long expiration times reduce the frequency of updates, which can lead to better stability. However this can lead to delays in reflecting changes in the network.

Short expiration times allow for more accurate and adaptable forwarding, but this obviously requires a greater amount of computation from the switch which can impact performance.

I think in order to acheive an optimal expiration time for a switch table you would need to take the pros and cons of the afformentioned statements into consideration as well as to consider the level of accuracy and responsiveness required of the specific network.