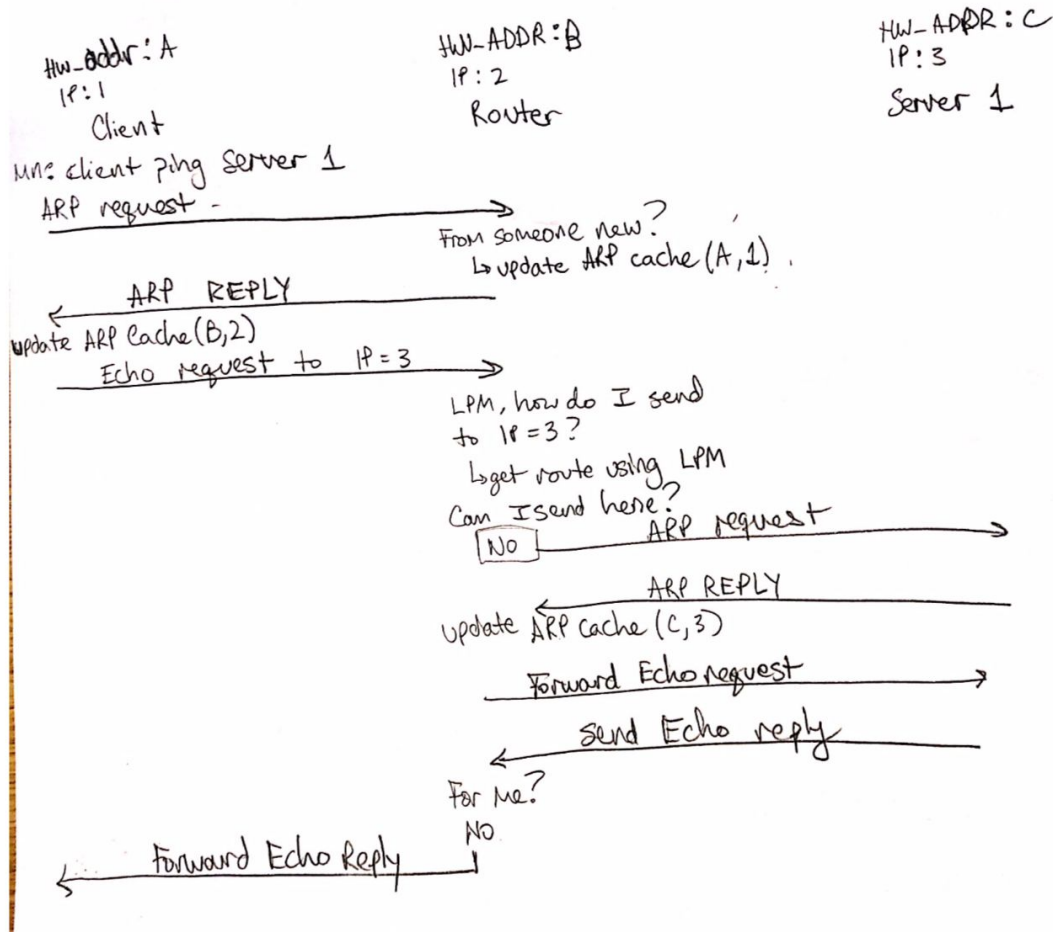


Charlie Andre
CSCI 551
Prof Govindan
Lab 1

Lab 1 Report

Design Structure:



Challenges:

Before doing this lab, I had no idea what ARP was, and I was a bit lost on the entire process of sending an ARP request in order to get the hardware address corresponding to an IP address. This made development really slow at first. Another challenge that I faced was in my basic lack of C coding skills. After a few days of working on this lab, I had refreshed my understanding of pointers and I was writing code successfully.

A big roadblock in developing this was also that I freed an interface that was borrowed from `handlepacket()`. This led to an interface not being asserted. I tried everything to debug this and it took a few days before realizing my mistake.

I coded up basic pinging (without forwarding) in a day after finding this mistake, editing and tweaking code that I already wrote.

Identified source of error:

I believe that my code does not work because I am interrupted by another echo request while I am trying to send the initial echo request to server1.

Explained Output:

The below is my terminal output when running my code. I am going to walk through and explain what is happening and specifically why what I am doing seems correct.

```
cs551@mininet-vm:~/candre97-cs551/lab1$ screen -r mn
```

```
There is no screen to be resumed matching mn.
```

```
cs551@mininet-vm:~/candre97-cs551/lab1$ ./router/sr
```

```
Using VNS sr stub code revised 2009-10-14 (rev 0.20)
```

```
Loading routing table from server, clear local routing table.
```

```
Loading routing table
```

```
-----
Destination    Gateway          Mask  Iface
10.0.1.0        10.0.1.100      255.255.255.0 eth3
192.168.2.2     192.168.2.2     255.255.255.255  eth1
172.64.3.0      172.64.3.10     255.255.255.0 eth2
-----
```

```
Client cs551 connecting to Server localhost:8888
```

```
Requesting topology 0
```

```
successfully authenticated as cs551
```

```
Loading routing table from server, clear local routing table.
```

```
Loading routing table
```

```
-----
Destination    Gateway          Mask  Iface
10.0.1.0        10.0.1.100      255.255.255.0 eth3
192.168.2.2     192.168.2.2     255.255.255.255  eth1
172.64.3.0      172.64.3.10     255.255.255.0 eth2
-----
```

```
Router interfaces:
```

```
eth3  HWaddr0e:32:1b:f6:c3:09
      inet addr 10.0.1.1
```

```
eth2  HWaddr2e:a3:2f:a3:79:bb
      inet addr 172.64.3.1
```

```
eth1  HWaddr ae:d9:a6:44:ca:21
      inet addr 192.168.2.1
```

```
<-- Ready to process packets -->
```

```
*** -> Received packet of length 42
```

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF

source: F2:8A:84:91:B3:70

type: 2054

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 1

sender hardware address: F2:8A:84:91:B3:70

sender ip address: 10.0.1.100

target hardware address: 00:00:00:00:00:00

target ip address: 10.0.1.1

ETHERTYPE: 2054

ETHERTYPE_IP: 2048

Got an ARP Packet

From someone new, adding them to ARP Cache

No requests waiting on this reply

Received ARP request

PACKET THAT I JUST MADE:

ETHERNET header:

destination: F2:8A:84:91:B3:70

source: 0E:32:1B:F6:C3:09

type: 2054

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 2 -- HERE I AM SENDING AN ARP REPLY TO CLIENT

sender hardware address: 0E:32:1B:F6:C3:09

sender ip address: 10.0.1.1

target hardware address: F2:8A:84:91:B3:70

target ip address: 10.0.1.100

sent

*** -> Received packet of length 98

ETHERNET header:

destination: 0E:32:1B:F6:C3:09

source: F2:8A:84:91:B3:70

type: 2048

IP header:

version: 4

header length: 5
type of service: 0
length: 84
id: 49266
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 10412
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 55818

ETHERTYPE: 2048

ETHERTYPE_IP: 2048

Got an IP Packet

Going to handle this IP packet

IP packet received on eth3 needs to be forwarded to: Found a longer mask

Found a longer mask

Found a match!

Found a match: eth1

can route to: 192.168.2.2

via my interface at:

192.168.2.1

DEST IP: 192.168.2.2 -- SUCCESSFULLY PERFORMING LPM, THOUGH MY METHOD IS A
BIT OF A HACK

Adding this packet to request Queue:

ETHERNET header:

destination: 0E:32:1B:F6:C3:09
source: F2:8A:84:91:B3:70
type: 2048

IP header:

version: 4
header length: 5
type of service: 0
length: 84
id: 49266
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1

checksum: 0
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 55818

As

ETHERNET header:

destination: 0E:32:1B:F6:C3:09
source: F2:8A:84:91:B3:70
type: 2048

IP header:

version: 4
header length: 5
type of service: 0
length: 84
id: 6912
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 10412
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527

HERE I ADD A COPY OF THE PACKET TO THE REQUEST QUEUE SO IT IS NOT ERASED
BEFORE I CAN DEAL WITH IT

Added a request to the queue for a MAC to match IP: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF
source: 0E:32:1B:F6:C3:09
type: 2054

ARP header

hardware type: 1
protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 1

sender hardware address: 0E:32:1B:F6:C3:09

sender ip address: 10.0.1.1

target hardware address: 00:00:00:00:00:00

target ip address: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF

source: 2E:A3:2F:A3:79:BB

type: 2054

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 1

sender hardware address: 2E:A3:2F:A3:79:BB

sender ip address: 172.64.3.1

target hardware address: 00:00:00:00:00:00

target ip address: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF

source: AE:D9:A6:44:CA:21

type: 2054

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 1

sender hardware address: AE:D9:A6:44:CA:21

sender ip address: 192.168.2.1

target hardware address: 00:00:00:00:00:00

target ip address: 192.168.2.2

HERE I SEND AN ARP REQUEST TO SERVER1, ASKING WHAT HARDWARE ADDRESS I SHOULD USE TO CONTACT 192.168.2.2

*** -> Received packet of length 42

ETHERNET header:

destination: AE:D9:A6:44:CA:21

source: 42:4A:15:B0:45:47

type: 2054

ARP header

hardware type: 1

protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 2
sender hardware address: 42:4A:15:B0:45:47
sender ip address: 192.168.2.2
target hardware address: AE:D9:A6:44:CA:21
target ip address: 192.168.2.1

ETHERTYPE: 2054

ETHERTYPE_IP: 2048

Got an ARP Packet

Received ARP reply

ARP header

hardware type: 1
protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 2
sender hardware address: 42:4A:15:B0:45:47
sender ip address: 192.168.2.2
target hardware address: AE:D9:A6:44:CA:21
target ip address: 192.168.2.1

Received an ARP reply

Updating ARP cache

New arp entry:

192.168.2.2

42:4A:15:B0:45:47

ADDING THE CORRECT INFORMATION TO MY APR CACHE

Handling a request that was waiting on this response

Handling this packet:

ETHERNET header:

destination: 0E:32:1B:F6:C3:09

source: F2:8A:84:91:B3:70

type: 2048

THIS IS A BACKED UP ECHO REQUEST THAT I CAN NOW FORWARD TO SERVER1

IP header:

version: 4

header length: 5

type of service: 0

length: 84

id: 6912

fragment flag: DF

fragment offset: 0

TTL: 64
protocol: 1
checksum: 10412
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527

Found a longer mask

Found a longer mask

Found a match!

Found a match: eth1

can route to: 192.168.2.2

via my interface at:

192.168.2.1

replying using this packet:

ETHERNET header:

destination: 42:4A:15:B0:45:47
source: AE:D9:A6:44:CA:21 -- FORWARDED TO SERVER1
type: 2048

IP header:

version: 4
header length: 5
type of service: 0
length: 84
id: 6912
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 10412
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527

*** -> Received packet of length 98

BEFORE I CAN RECEIVE AN ECHO REPLY FROM SERVER1, I AM INTERRUPTED WITH
ANOTHER ECHO REQUEST FROM CLIENT

ETHERNET header:

destination: 0E:32:1B:F6:C3:09

source: F2:8A:84:91:B3:70

type: 2048

IP header:

version: 4

header length: 5

type of service: 0

length: 84

id: 49504

fragment flag: DF

fragment offset: 0

TTL: 64

protocol: 1

checksum: 15019

source: 10.0.1.100

destination: 192.168.2.2

ICMP header:

type: 8

code: 0

checksum: 47720

ETHERTYPE: 2048

ETHERTYPE_IP: 2048

Got an IP Packet

Going to handle this IP packet

IP packet received on eth3 needs to be forwarded to: Found a longer mask

Found a longer mask

Found a match!

Found a match: eth1

can route to: 192.168.2.2

via my interface at:

192.168.2.1

DEST IP: 192.168.2.2

Adding this packet to request Queue:

THIS IS ESSENTIALLY THE ERROR I HAVE BEEN STUCK ON ALL DAY-- EVEN THOUGH I UPDATED MY ARP CACHE, I AM EITHER READING FROM ARP CACHE INCORRECTLY OR I AM LOOKING UP AN ARP ENTRY INCORRECTLY. HOWEVER I AM STILL ABLE TO SEND THE ECHO REQUEST TO SERVER 1 EVENTUALLY.

ETHERNET header:

destination: 0E:32:1B:F6:C3:09

source: F2:8A:84:91:B3:70

type: 2048

IP header:

version: 4

header length: 5

type of service: 0
length: 84
id: 49504
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 0
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 47720

as

ETHERNET header:

destination: 0E:32:1B:F6:C3:09
source: F2:8A:84:91:B3:70
type: 2048

IP header:

version: 4
header length: 5
type of service: 0
length: 84
id: 6912
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 15019
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527

Added a request to the queue for a MAC to match IP: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF
source: 0E:32:1B:F6:C3:09
type: 2054

ARP header

hardware type: 1
protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 1
sender hardware address: 0E:32:1B:F6:C3:09
sender ip address: 10.0.1.1
target hardware address: 00:00:00:00:00:00
target ip address: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF
source: 2E:A3:2F:A3:79:BB
type: 2054

ARP header

hardware type: 1
protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 1
sender hardware address: 2E:A3:2F:A3:79:BB
sender ip address: 172.64.3.1
target hardware address: 00:00:00:00:00:00
target ip address: 192.168.2.2

ARP REQUEST THAT I JUST SENT:

ETHERNET header:

destination: FF:FF:FF:FF:FF:FF
source: AE:D9:A6:44:CA:21
type: 2054

ARP header

hardware type: 1
protocol type: 2048
hardware address length: 6
protocol address length: 4
opcode: 1
sender hardware address: AE:D9:A6:44:CA:21
sender ip address: 192.168.2.1
target hardware address: 00:00:00:00:00:00
target ip address: 192.168.2.2

*** -> Received packet of length 42

ETHERNET header:

destination: AE:D9:A6:44:CA:21
source: 42:4A:15:B0:45:47

type: 2054

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 2

sender hardware address: 42:4A:15:B0:45:47

sender ip address: 192.168.2.2

target hardware address: AE:D9:A6:44:CA:21

target ip address: 192.168.2.1

ETHERTYPE: 2054

ETHERTYPE_IP: 2048

Got an ARP Packet

Received ARP reply

ARP header

hardware type: 1

protocol type: 2048

hardware address length: 6

protocol address length: 4

opcode: 2

sender hardware address: 42:4A:15:B0:45:47

sender ip address: 192.168.2.2

target hardware address: AE:D9:A6:44:CA:21

target ip address: 192.168.2.1

Received an ARP reply

Updating ARP cache

New arp entry:

192.168.2.2

42:4A:15:B0:45:47

Handling a request that was waiting on this response

Handling this packet:

ETHERNET header:

destination: 0E:32:1B:F6:C3:09

source: F2:8A:84:91:B3:70

type: 2048

IP header:

version: 4

header length: 5

type of service: 0

length: 84

id: 6912

fragment flag: DF

fragment offset: 0
TTL: 64
protocol: 1
checksum: 15019
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527

Found a longer mask

Found a longer mask

Found a match!

Found a match: eth1

can route to: 192.168.2.2

via my interface at:

192.168.2.1

replying using this packet:

ETHERNET header:

destination: 42:4A:15:B0:45:47
source: AE:D9:A6:44:CA:21
type: 2048

IP header:

version: 4
header length: 5
type of service: 0
length: 84
id: 6912
fragment flag: DF
fragment offset: 0
TTL: 64
protocol: 1
checksum: 15019
source: 10.0.1.100
destination: 192.168.2.2

ICMP header:

type: 8
code: 0
checksum: 65527