ECE435: Network Engineering – Homework 7
Internet Protocol v4

Due: Friday, 14 March 2025, 5:00pm

For this homework short answers will suffice.

To submit, create a document with your answers (text, pdf, libreoffice, MS Office if you must) and e-mail them to *vincent.weaver@maine.edu* by the homework deadline. Title your e-mail "ECE435 Homework 7" and be sure your name is included in the document.

1. If you recall from previous homeworks we looked at a packet similar to this:

```
0x0000:  0013 3b10 667f b827 ebaf 3711 0800 4500  ..;.f..'..7...E.
0x0010:  0038 572a 4000 4006 69cc c0a8 0833 826f  .8W*@.@.i....3.o
0x0020:  2e7f bda5 0050 cdc4 6a49 3c7b 6ca5 8018  .....P..jI<{l...
0x0030:  00e5 79f4 0000 0101 080a 0104 3e58 34a8  ..y.........>X4.
0x0040:  7bc3 4745 540a                           {.GET.
```

   The IPv4 header begins at offset 0xe.
   Fill in the table with the name of the field as well as the decoded value. Use decimal when decoding if it makes sense, provide units if necessary, and if the value decoded has a meaning (such as a flag of pre-defined value) say what it means. Give sizes in bytes if possible, and any IPv4 addresses show in dotted decimal.
   For help decoding the IPv4 header see the class notes or else RFC791.

BEGIN IPv4 HEADER	Name of Field	Decoded Value
`0x000e:` 4		
`0x000e:` 5		
`0x000f:` 00		
`0x0010:` 0038		
`0x0012:` 572a		
`0x0014:` 4000		
`0x0016:` 40		
`0x0017:` 06		
`0x0018:` 69cc		
`0x001a:` c0a8 0833		
`0x001e:` 826f 2e7f		
END IPv4 HEADER		

2. Which of the following are valid IPv4 addresses?

   (a) 1.1.1.1

   (b) 123.67.267.44

   (c) 192.168.8.1

   (d) 3232237569

   (e) 0xc0a80801

3. Early internet adopters got large IPv4 allocations. For example Ford (the car company) owned all of 19.0.0.0/8. What percentage of the entire IPv4 space is that? (Somewhat related, this old xkcd comic gives an interesting map of the IPv4 situation at the time: `https://xkcd.com/195/`)

4. A network is described as 192.168.13.0/24.

   (a) What would be the subnet mask for this subnet?

   (b) What would be the lowest IP address you could assign on this subnet?

   (c) What would be the highest IP address you could assign on this subnet?

5. Traditionally on Linux you could use the `route` command to find out the IP routing information for a system. Here are the results from a Raspberry Pi on one of my networks.

```
pi3:~$ /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.8.2     0.0.0.0         UG    0      0        0 eth0
192.168.8.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

   The `route` command is now considered deprecated and you can now find the same info with the `ip route` command

```
pi3:~$ ip route
default via 192.168.8.2 dev eth0 proto dhcp src 192.168.8.138 metric 202
192.168.8.0/24 dev eth0 proto dhcp scope link src 192.168.8.138 metric 202
```

   (a) If a packet is sent to 216.58.192.132, what is its first "hop" on the way?

   (b) If a packet is sent to 192.168.8.50 what is its first "hop" on the way?

6. Use the "ping" command on a network connected machine to `ping www.google.com`. (If you don't have access to a machine with ping on traceroute available, let me know and I can provide access)

   (a) What is the round-trip packet time?

   (b) Do you notice anything odd about the hostname that responds?

7. Use the "traceroute" command. It's tracert on Windows.

   (a) `traceroute www.maine.edu`.
   How many hops away is it? Do you recognize any of the names in the hops along the way?

   (b) `traceroute www.facebook.com`.
   How many hops away is it? Do the response times gradually go up for each further hop?

8. **Network Address Translation**

   (a) You use tcpdump to monitor your network and see packets such as this go by:

```
16:58:49.108396 00:13:3b:10:66:7f (oui Unknown) >
00:50:b6:47:1c:de (oui Unknown), ethertype IPv4 (0x0800),
length 141: google-public-dns-a.google.com.domain >
macbook-air.51415: 30858 2/0/0
CNAME pagead46.l.doubleclick.net., A 172.217.7.130 (99)
```

   where macbook-air has the address 192.168.8.38 and it is connecting to IP 8.8.8.8.
   Is it normal for an address like 192.168.8.38 to be able to connect directly to 8.8.8.8? Why or why not?

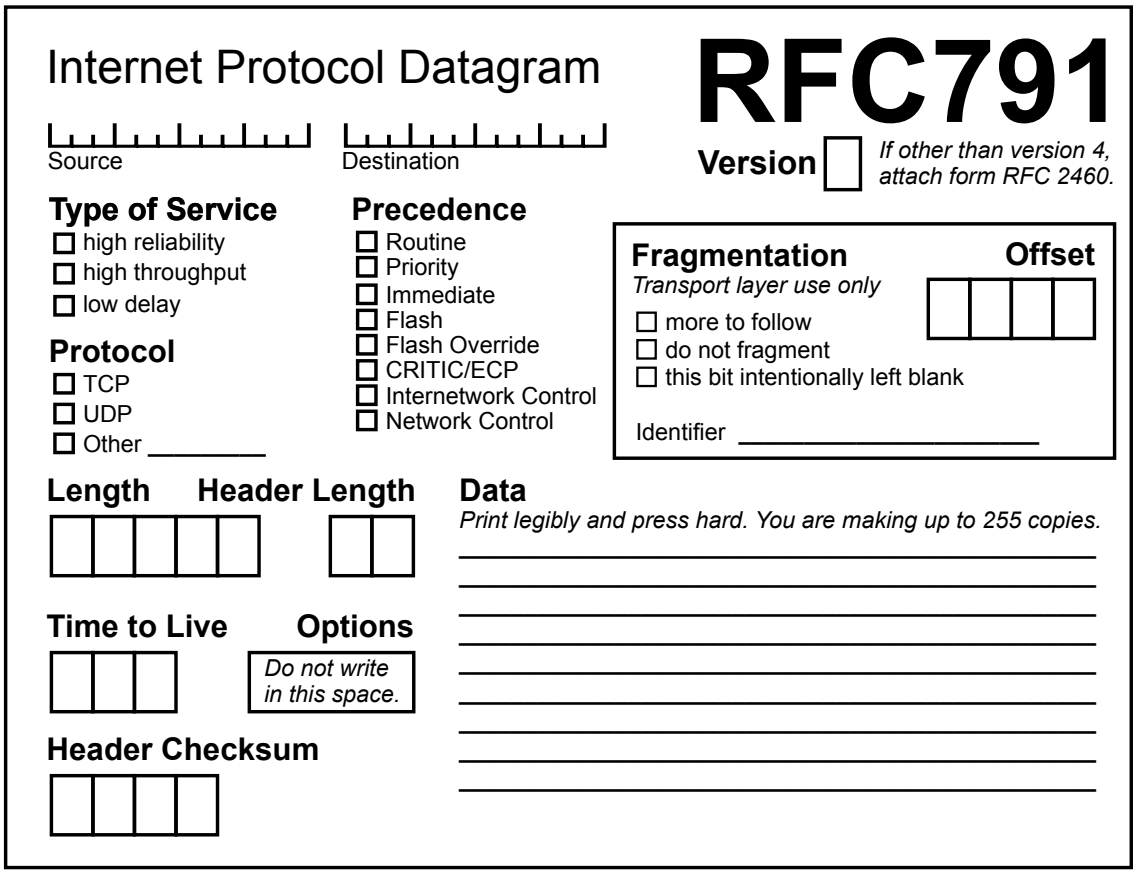
   What is the likely reason this is working?

   (b) You can use the `netstat-nat` command on a Linux machine doing NAT to see all of the nat connections. Some sample output is below.

```
Proto NATed Address              Destination Address            State
tcp   macbook-air:51908          49.246.178.107.bc.google:https ESTABLISHED
tcp   macbook-air:55194          iad23s63-in-f19.1e100.ne:https ESTABLISHED
tcp   macbook-air:42334          206-140.amazon.com:https       TIME_WAIT
tcp   macbook-air:52930          104.16.78.166:https            ESTABLISHED
tcp   macbook-air:57928          akamai-1-s.net.maine.edu:http  ESTABLISHED
udp   macbook-air:58903          google-public-dns-a.goo:domain ASSURED
udp   macbook-air:49779          google-public-dns-a.goo:domain ASSURED
udp   macbook-air:44416          google-public-dns-a.goo:domain UNREPLIED
udp   pi2:ntp                    clock.xmission.com:ntp         ASSURED
udp   pi2:ntp                    38.88.18.251:ntp               ASSURED
udp   pi2:ntp                    tock.no-such-agency.net:ntp    ASSURED
```

   One of the UDP connections is listed as UNREPLIED. Why might the NAT firewall track whether a UDP packet has been replied to or not?

Not a question, but you might know enough about IP networking now to find this novelty notepad amusing:



for more info, check IPv4 specifications at http://www.ietf.org/rfc/rfc0791.txt