Lecture 12

Assignment 4

A code consisting of four (4) 6-bit codewords is formed by taking all possible 2-bit values and replicating them three times. For example, the first codeword is 000000 (binary), the second is 010101, etc.

(a) What are n and k?

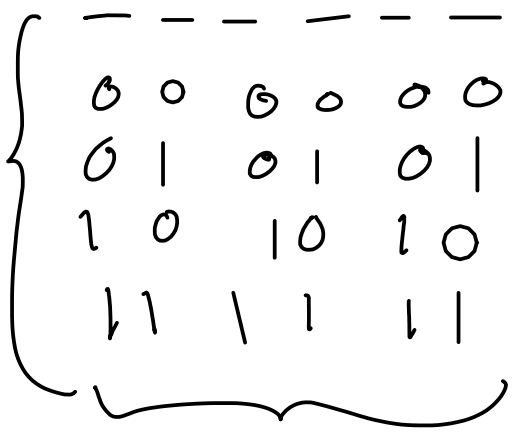
(b) What is the code rate? $\frac{k}{n}$

(c) What is the minimum distance of this code?

(d) How many errors can this code detect? How many can it correct?

(e) You receive the codeword 001010. What codeword was most likely transmitted? Why? How many errors were corrected?

(f) You receive the codeword 001001. How many errors were detected? Can you tell what they are?



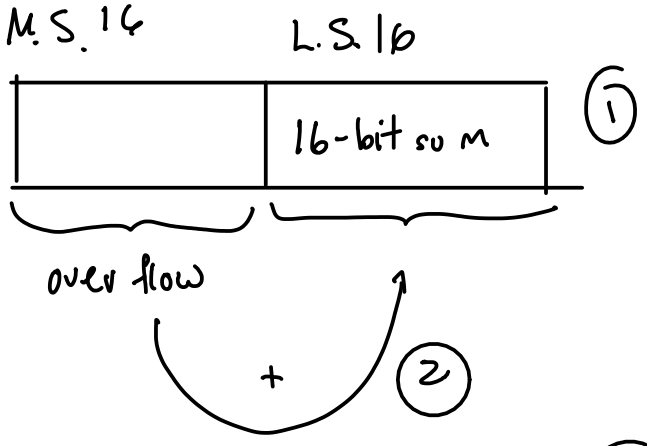
$n = 6$

$k = 2 = \log_2 4$

$M \rightarrow \log_2 M$ bits of information

$\log_2 8 = 3$ $\quad 2^3 = 8$

$\log_2 16 =$ $\quad 2^4 = 16$



Exercise 1: What is the difference between IP and "The Internet"? Does a network using IP have to be on the Internet? Does someone using the Internet have to use IP?

IP = a protocol

Internet = a network using IP

no. private networks can (and do) use IP.

yes. to communicate with other hosts on the internet you must use IP.

Exercise 2: A protocol header contains four 16-bit fields with decimal values 65535, 1, 2, and 3 that are to included in an IPv4 checksum. What is the value of the header checksum?

(1)

```
  65535
      1
      2
      3
-------
  65541
```

0x 0001 0005

(2)

```
  0x 0001
+ 0x 0005
---------
  0x 0006
```

6 → 0110

(3)

1111 1111 1111 1001

Exercise 3: What is the netmask in binary for a /24 network? What is it in decimal? How can the netmask be used to determine if one IP address is on the same network as another? Is the address 192.168.2.200 in the 192.168.2.0/25 network?

/24 ⇒ 24 MS bits are 1

1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

255 . 255 . 255 . 0

bitwise - AND netmask & IP address gives the network address

bitmask for /25 network is 255.255.255.128

192.168.2.200
AND 255.255.255.128

192.168.2.128 /25

0x C8
AND 0x 80

0x 80

NOT in 192.168.2.0 /25

Exercise 4: Who "owns" the 204.191.0.0/16 network?

Telus

Exercise 5: For the routing table above, what port ("Interface") would be used by frames with the following destination IP addresses: 127.0.0.255? 192.168.1.1? 192.168.2.1? 204.191.10.32?

outer:

Destination	Gateway	Subnet Mask	Metric	Interface
192.168.1.0	*	255.255.255.0	0	br0 (LAN)
204.191.0.0	*	255.255.0.0	0	vlan1 (WAN)
127.0.0.0	*	255.0.0.0	0	lo
default	204.191.1.1	0.0.0.0	0	vlan1 (WAN)

127.0.0.255 → lo

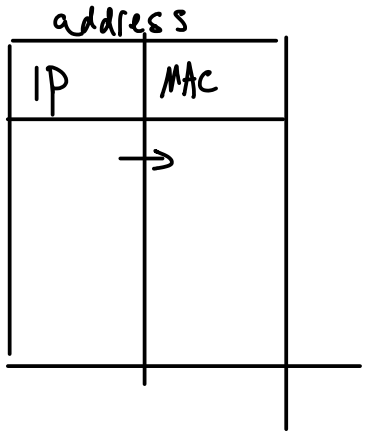
192.168.1.1 → 192.168.1.0 network → br0

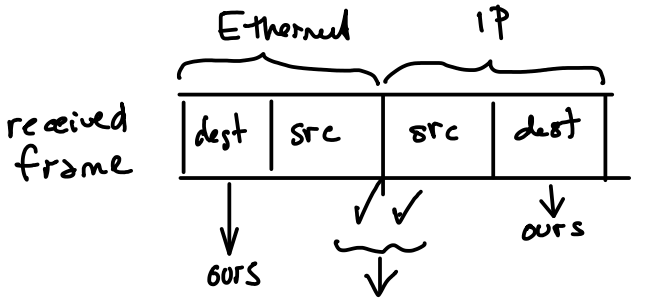
192.168.2.1 → vlan1 (to gateway 204.191.1.1)

204.191.10.32 → 204.191.0.0 network → vlan1

Exercise 6: What pairs of values are stored in an ARP cache? What values from a received frame need to be examined to validate an ARP cache entry?

ARP cache





we can use src MAC & IP addresses to update ARP cache

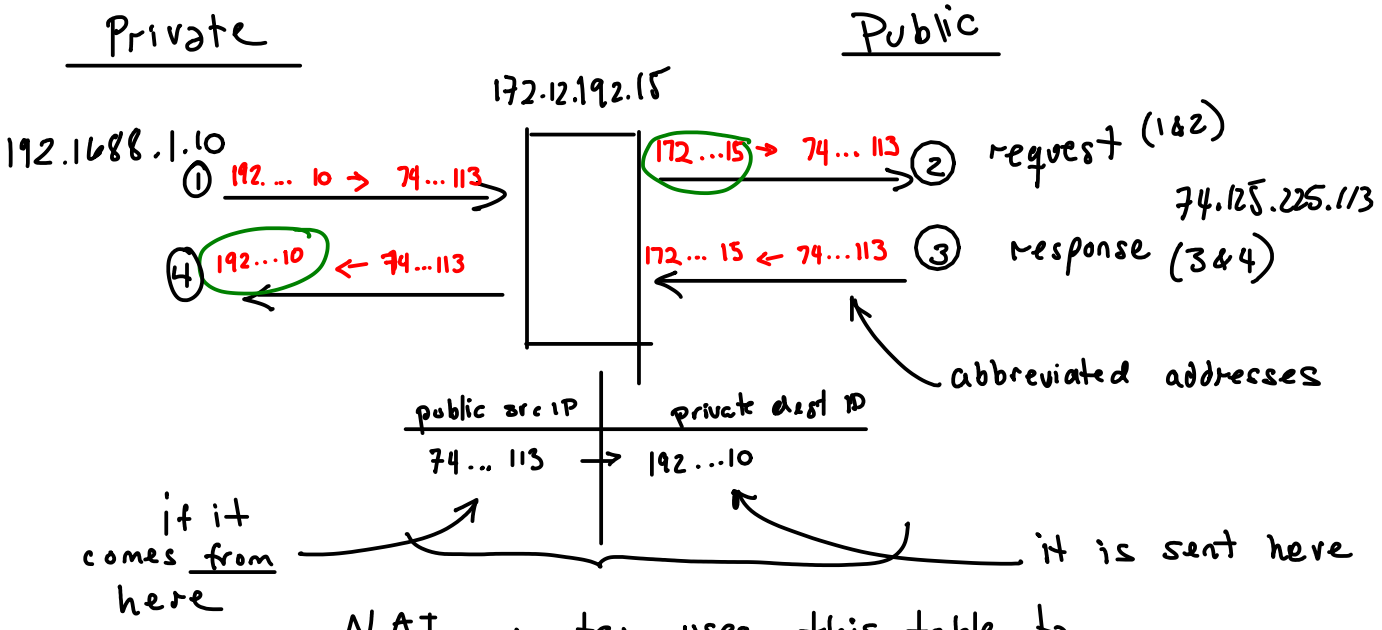
Exercise 7: When a host boots up, what must it send out first, an ARP request or a DHCP request?

ARP request is to get MAC address for an destination IP address and is sent as broadcast Ethernet frame to destination = ff:ff:ff:ff:ff:ff

DHCP frames are sent to Ethernet & IP broadcast addresses, so we don't need to get any IP addresses before the DHCP request.

∴ neither protocol requires the other to run first.

Exercise 8: A host with a (private) address 192.168.1.10 is behind a NAT router with an (public) address of 172.12.192.15. The host sends a frame to a host at address 74.125.225.113 requesting a web page. Show the source/destination address pairs of the request and response frames on the private and public sides of the router.

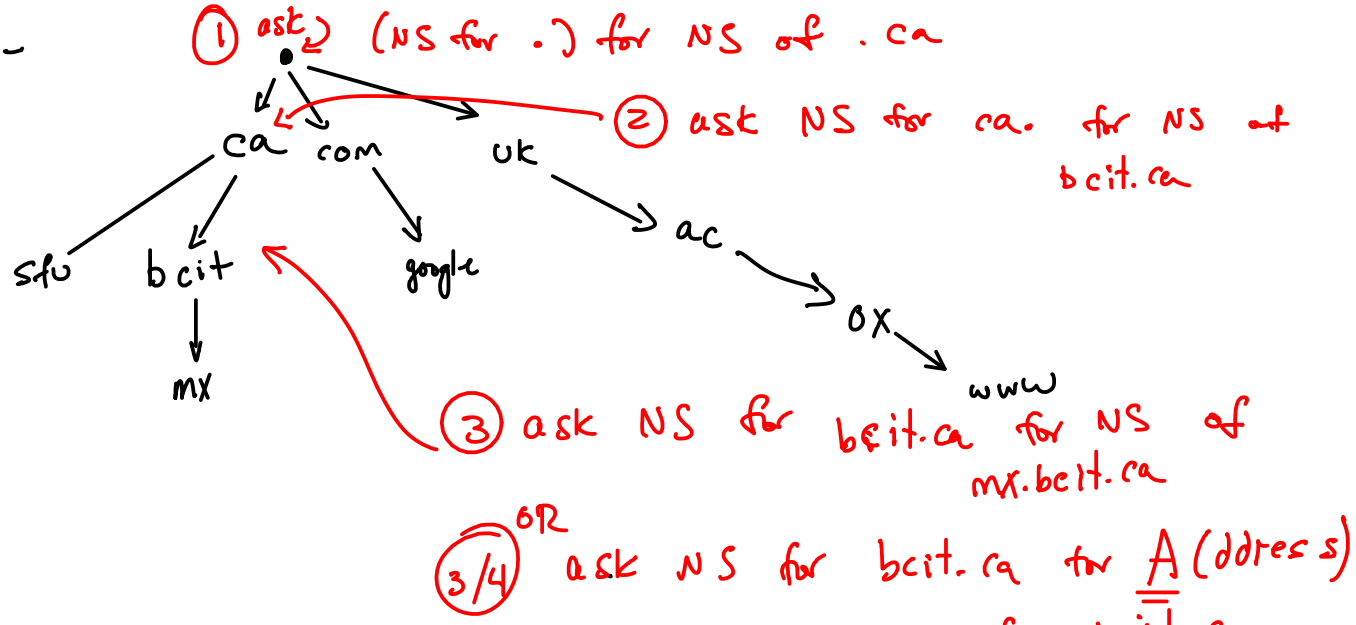


NAT router uses this table to decide which host on the priate side incoming frames should be sent to

recursive name server

Exercise 9: Can a host's DNS server be configured using a host name? Why or why not? Assuming a ~~host~~ has an empty DNS cache, what queries would it generate to look up the IP address of the host mx.bcit.ca?

- DNS server must be on IP address because no other way to get its IP address.



lookup("mx.bcit.ca")
↳ lookup("bcit.ca")
↳ lookup("ca")
↳ (".")