CN Sheet 8

Network Layer in the Internet

1. Show the classful address ranges, number of hosts and their relative network masks.

Class	Indicator		Mask	#Networks	#Hosts
	Binary (Initial Bits)	Decimal (Range)			
A	0	$1-127$	$/8$	2^{8-1}	$2^{24}-2$
B	10	$128-191$	$/16$	2^{16-2}	$2^{16}-2$
C	110	$192-223$	$/24$	2^{24-3}	$2^{8}-2$
D	1110	$224-239$	Multicasting		
E	1111	$240-255$	Future Use		

2. Briefly describe one disadvantage of the classful addressing and how subnetting can solve it.

The network address for each of the classes has a fixed number of hosts which may not suit the needs of an organization. A is too big and C is too small which caused most organizations to go for B which still offers too many hosts than the organization needs which leads to a waste in the address space although we are running out of IPv4 addresses anyway.

Subnetting solves the problem by allowing multiple organizations (e.g., through an ISP) to share the same internet-wide network address and then use a portion of the host bits as a continuation to the network address to identify the organization as a subnetwork within the network.

3. Given the following addresses, choose their relative class) and their default subnet mask (network mask)

IP	10.250.1.1	192.14.2.0	150.10.15.0	230.230.45.58	219.21.56.0

Recall,

Class	A	B	C	D	E
Start	1	128	192	224	249
Mask	8	16	24	-	-

4. Using the IP address and subnet mask shown write out the network address and the host address of the following:

IP Address	Network Mask	Network Address	Host Address
188.10.18.2	255.255.0.0	188.10.0.0	0.0.18.2
10.10.48.80	255.255.255.0	10.10.48.0	0.0.0.80
223.169.23.20	255.255.0.0	223.169.0.0	0.0.23.20

All we need to do is AND the IP address and the network mask to get the network address then the rest of the IP address is the host address. Try solving the last row again with the subnet mask 255.255.192.00, you will need to convert 192 and 023 to binary first.

5. Consider a company provided with the address range from 193.62.83.0 to 193.62.83.255 calculate the maximum number of hosts if it uses classful addressing and compare it to using classless addressing with 255.128.0.0 as network mask.

If it uses classful addressing then its sufficient to know that 193.62.83.x is one of the host IP addresses in the system to conclude from the first three digits that this is a class C network and that the network address by ANDing with /24 is 193.62.83.0 which has hosts with IPs ranging from 193.62.83.1 to 193.62.83.254 and equivalently as known from Q1, 256-2 hosts.

On the other hand, if classless addressing is used with 255.128.0.0 as the network mask, then without even knowing the network IP address we can conclude from the mask that its 9 bits (as $255.128.0.0 = 11111111\ 10000000\ 00000000\ 00000000$) which means that we have $32 - 9 = 23$ bits left for the host which makes $2^{23} - 2$ of them possible. Additionally, if we knew that the network address is 211.128.0.0 then the hosts have IPs starting from 211.128.0.1 to 211.255.255.254 (convert both to decimal to see this).

6. A company using classful addressing is provided by network address 192.10.10.0, the company needed at least 14 subnets each of them has at least 14 hosts, show the address class, default and custom subnet masks and total number of subnets and bits borrowed for subnetting then find the total number of hosts and compare it with no subnetting.

1. Address Class

   First three digits are 192 so its class C

2. Default Subnet Mask

   Since its class C then default subnet mask is /24 (255.255.255.0)

3. Custom Subnet Mask

   The custom subnet mask is the subnet mask after subnetting. Since class C has 8 host bits, we need to decide how many bits will let any router in the system decide one of the 14 subnets. The least number of bits we can give for this is clearly 4 (i.e., $ceil(log_2 14)$) so custom subnet mask is /(24+4)=/28 (i.e., 255.255.255.240). Any router in the system must use this one instead of /24 which will be used by routers outside the system.

4. Total Number of Possible Subnets

   Since we took 4 bits for the subnet (couldn't take less) this allows up to $2^4 = 16$ subnets (two will be unused in this case).

5. Total Number of Hosts

   Since we took 4 bits for the subnet it must be the case that 4 bits are left for the host so we have $2^4 - 2 = 16 - 2 = 14$ hosts for each subnet. This is a consequence of the requirement "need 14 subnets" regardless of the "need 14 hosts for each" requirement. If this client wanted any more hosts, it would be impossible given their poor class C network address.

6. Number of Bits Borrowed for Subnetting

   4 bits were borrowed from the host portion for subnetting.

7. Compare the Number of Hosts if we didn't use Subnetting at all.

   If we didn't use subnetting then class C offers $2^8 - 2 = 254$ hosts, after subnetting we have $N_{subnets} * N_{hosts} = 16 * 14 = 224$ total possible hosts. The difference in 30 hosts is because each subnet of the 16 must reserve the 2 special addresses since each is considered a network compared to reserving them once for the whole class C network which implies $(\quad - 1) * 2 = 30$ less hosts.

7. Convert the IP address whose hexadecimal representation is C22F1582 to dotted decimal notation.

C2 2F 15 82=1100 0010.0010 1111.0001 0101.1000 0010= 194.47.21.130

8. A large number of consecutive IP address (classless) are available starting at 198.16.0.0. Suppose that four organizations, A, B, C, and D, request 4000, 2000, 4000, and 8000 addresses, respectively, and in that order. For each of these, give the first IP address assigned, the last IP address assigned, and the mask in the w.x.y.z/s (CIDER) notation.

   Notice that the IPs we have start from 11000110 00010000 00000000 00000000 in binary and that we can't apply ordinary subnetting over this network address (decide a fixed number of subnet bits for all) because they don't require the same number of hosts.

   Since each organization will have its own network address, any IP assignments we do must be in powers of 2. So, we can only assign, by taking ceil log of each, $2^{12} = 4096, 2^{11} = 2048, 2^{12} = 4096, 2^{13} = 8192$ to A, B, C, D respectively.

   A needs 12 bits for the host so its start and end IPs must take the form
   From: 11000110 00010000 AAAA0000 00000000
   To: 11000110 00010000 AAAA1111 11111111

   B needs 11 bits for the host so its start and end IPs must take the form
   From: 11000110 00010000 BBBBB000 00000000
   To: 11000110 00010000 BBBBB111 11111111

   C needs 12 bits for the host so its start and end IPs must take the form
   From: 11000110 00010000 CCCC0000 00000000
   To: 11000110 00010000 CCCC1111 11111111

   D needs 13 bits for the host so its start and end IPs must take the form
   From: 11000110 00010000 DDD00000 00000000

To: 11000110 00010000 DDD11111 11111111

Now all we need to decide are the subnet field for each which any router should be able to use to figure in which of the four networks a given packet belongs, this implies any subnet field assigned to one of them should never occur as a prefix in another. (e.g., if we let DDD=100 then CCCC can't be 1001 as in this case a message to D could be confused as being to C)

Assuming that it's required to assign the IP blocks in the order A, B, C, D then we must choose them as well such that that AAAA0<BBBBB<CCCC0<DDD00

Let DDD = 111 // Start from the smallest to watch out for the prefix
Let CCCC = 1101
Let AAAA = 0001
Let BBBBB = 00100

Observe that they satisfy the two conditions mentioned above. Now if we decide to go for these, we get the following IPs:

A's IP Block:
From: 198.16.16.0/20
To: 198.16.31.255/20

B's IP Block:
From: 198.16.32.0/21
To: 198.16.39.255/21

C's IP Block:
From: 198.16.208.0/20
To: 198.16.223.255/20

D's IP Block:
From: 198.16. 224.0/19
To: 198.16.255.255/19

Notice that our assignment assumes that we have the full 192.16.x.x range, this is somewhat rational (compared to going out of 192.16.x.x because no upper bound is given) but we can also try to make the assignment in the most compact way (spanning the least number of IPs in 192.16.x.x) as follows

<Bonus>

Let AAAA = 0000

Now A lasts until

To: 11000110 00010000 00001111 11111111

If we add 1 to this, we get the first next unused IP address, 11000110 00010000 00010000 00000000 we can let this be the start for B by letting BBBBB=00010

Now B lasts until

To: 11000110 00010000 00010111 11111111

If we add 1 to this, we get the first next unused IP address, 11000110 00010000 00011000 00000000 we can't let this be the start for C as it has 4 bits for the subnet field and thus if we let CCCC=0001 and start at 00011000 00000000 we don't give C the number of hosts it wants. In this case, to get the next first unused IP that C can use skip the entire current 0001 block by doing CCCC = 0001+1= 0010.

Now C lasts until

To: 11000110 00010000 00101111 11111111

If we add 1 to this, we get the first next unused IP address, 11000110 00010000 00110000 00000000 we can't let this be the start for D as it has 3 bits for the subnet field and thus if we let DDD=001 and start at 00110000 00000000 we don't give D the number of hosts it wants. In this case, to get the next first unused IP that D can use we skip the entire current 001 block by doing DDD = 001+1= 010

Now D lasts until

To: 11000110 00010000 01011111 11111111

So far, we have AAAA=0000, BBBBB=00010, CCCC=0010, DDD=010. Notice that the order constraint was implicitly taken care of.

The IP addresses become:

A's IP Block:
From: 198.16.0.0/20
To: 198.16.15.255/20

B's IP Block:
From: 198.16.16.0/21
To: 198.16.23.255/21

C's IP Block:
From: 198.16.32.0/20
To: 198.16.47.255/20

D's IP Block:
From: 198.16.64.0/19
To: 198.16.95.255/19

</Bonus>

9. A router has the following (CIDR) entries in its routing table:

Address/Mask	Next Hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
default	Router2

What hop does the router choose if the destination IP of the packet is each of the following:

135.46.63.10	135.46.57.14	135.46.52.2	192.53.40.7	192.53.56.7

1. Make sure the givens are network addresses* and write them in binary

Network Address	Binary Network Addresses	Next Hop
135.46.56.0/22	10000111. 00101110. 00111000. 00000000	Interface 0
135.46.60.0/22	10000111. 00101110. 00111100. 00000000	Interface 1
192.53.40.0/23	11000000. 00110101. 00101000. 00000000	Router 1
default	default	Router2

 # For 135.46.63.10 we know if this will match it must be one of the first two (obviously doesn't match the third) and they all have in common 135.46.x.x so we can start checking from 63, considering only the first $22 - 8 - 8 = 6$ bits for both in the process (prefix length is 22):
63 = 00111111 which doesn't match the first (001110≠001111) but matches the second so we forward on Interface 1.

 # For 135.46.57.14 we know if this will match it must be one of the first two and they all have in common 135.46.x.x so we can start checking from 57:
57 = 00111001 which doesn't match the second (001111≠001110) but matches the first so we forward on Interface 0.

 # For 135.46.52.2 we know if this will match it must be one of the first two and they all have in common 135.46.x.x so we can start checking from 52:
52 = 00110100 which doesn't match either so we forward towards router 2 (default)

 # For 192.53.40.7 we know if this will match it must be the third (obviously doesn't match first or second) and they have in common 192.53.x.x so we can start checking from 40, considering only the first $23 - 8 - 8 = 7$ bits in the process (prefix length is 23):
40 = 00101000 which matches the third so we forward towards router 1.

 # For 192.53.56.7 we know if this will match it must be the third and they have in common 192.53.x.x so we can start checking from 56:
56 = 00111000 which doesn't match third so we forward towards router 2 (default)

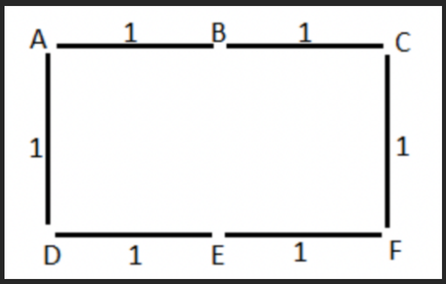
Some Notes:

- If the two routers had different masks, then when matching for each, we use the router's specific match
- We wrote the whole binary just because it feels safer that way but we don't really have to.

CN Sheet 7

// All problems of sheet 6 and 7 were solved in the written notes except for one that we'll solve here.

2. Consider the following network topology where initially each node has a table with distances to immediate neighbors measured in hops. Perform two distance vector exchanges.



Initial Tables

//In each cost column, put 1 if it's a neighbor and write it in 'Line', 0 if its you and ∞ otherwise

TO	A		B		C		D		E		F	
	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line
A	0		1	A	∞		1	A	∞		∞	
B	1	B	0		1	B	∞		∞		∞	
C	∞		1	C	0		∞		∞		1	C
D	1	D	∞		∞		0		1	D	∞	
E	∞		∞		∞		1	E	0		1	E
F	∞		∞		1	F	∞		1	F	0	

First Exchange

// Consider how we'd update A only; A will add 1 to the costs assigned by B and D (its neighbors) then make the update by taking the min (cost) over the neighbors only for each node in TO except A (or more wisely, only distant neighbors since 1 for neighbors can't improve) update the cost with min and the line with the neighbor that caused it. Same applies to all other nodes and always use the old table above (simultaneous updates).

// neighbors: n(A)={B,D}, n(B)={A,C), n(C)={B,F}, n(F)={C,E}, n(E)={D,F}, n(D)={E,A}

TO	A		B		C		D		E		F	
	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line
A	0		1	A	2	B	1	A	2	D	∞	
B	1	B	0		1	B	2	A	∞		2	C
C	2	B	1	C	0		∞		2	F	1	C
D	1	D	2	A	∞		0		1	D	2	E
E	2	D	∞		2	F	1	E	0		1	E
F	∞		2	C	1	F	2	E	1	F	0	

Second Exchange

// Repeat, use the table above (result from first exchange). Observe that after the 1st exchange all entries for each node have converged (they won't get updated even if you try) so you only need to update the ∞ distant neighbor for each node and that's it.

// neighbors: n(A)={B,D}, n(B)={A,C), n(C)={B,F}, n(F)={C,E}, n(E)={D,F}, n(D)={E,A}

TO	A		B		C		D		E		F	
	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line	Cost	Line
A	0		1	A	2	B	1	A	2	D	3	C
B	1	B	0		1	B	2	A	3	D	2	C
C	2	B	1	C	0		3	A	2	F	1	C
D	1	D	2	A	3	B	0		1	D	2	E
E	2	D	3	A	2	F	1	E	0		1	E
F	3	B	2	C	1	F	2	E	1	F	0	

Thank you <3.