CIDR, Route Summarisation and Routing

1 Examples

1. Aggregate the following set of 4 24-bit network addresses to the highest degree possible.

172.47.30.0/24

172.47.31.0/24

172.47.32.0/24

172.47.33.0/24

Here is how to do it:

List each address in binary format and determine the common prefix for all of the addresses:

	|← . . . . . . first prefix . . . . . . →	
172.47.30.0/24	10101100.00101111.0001111	**0**.00000000
172.47.31.0/24	10101100.00101111.0001111	**1**.00000000
172.47.32.0/24	10101100.00101111.0010000	**0**.00000000
172.47.33.0/24	10101100.00101111.0010000	**1**.00000000
	|← . . . . second prefix . . . . →	

Note that this set of 4 24-bit blocks cannot be summarised as a single 22-bit block.

172.47.30.0/23 10101100.00101111.0001111**0**.00000000

172.47.32.0/23 10101100.00101111.0010000**0**.00000000

So the two 23-bit blocks are:

172.47.30.0/23

172.47.32.0/23

Note: it looks as if there could be an 18-bit prefix in common; is it possible to choose 172.47.30.0/18? No, because this includes $2^{32-18=14} = 16384$, while there are only $2^8 \times 4 = 1024$ addresses in the original four blocks. The aim is to include only our addresses, not those that belong to others.

General Approach:

(a) Determine which octet the prefix will end in. Here, we have $2^8 \times 4 = 1024$ addresses, so we have the prefix ending in the third octet.

(b) Convert that octet only from the first and last address, to binary. So here, we convert $30_{10} \to 00011110_2$ and $33_{10} \to 00100001_2$.

(c) Do these binary numbers have a common prefix, to the right of which all bits count from 000...000 to 111...111? Well, in this case, no, so...

(d) Find the power of two over which the third octet counts. Here, the power of 2 is $32 = 2^5$. Convert the value before and after the power of 2 to binary: $2^5 - 1 = 00011111_2$, and $2^5 = 00100000_2$.

(e) Now compare the first $30_{10} \rightarrow 00011110_2$ with $2^5 - 1 = 00011111_2$, and see if we have a common prefix, to the right of which all bits count from 000...000 to 111...111. Well, yes we do! It is $0001111|x$.

(f) Now compare $2^5 = 00100000_2$ with $33_{10} \rightarrow 00100001_2$. Can we see a common prefix, with bits to the right counting from all 0's to all 1's? Yes! It is $0010000|x$.

2. Aggregate the following set of (64) 24-bit network addresses to the highest degree possible.

202.1.96.0/24
202.1.97.0/24
202.1.98.0/24
⋮
202.1.126.0/24
202.1.127.0/24
202.1.128.0/24
202.1.129.0/24
⋮
202.1.158.0/24
202.1.159.0/24

Here is how to do it:

List each address in binary format and determine the common prefix for all of the addresses:

	|←...first prefix...→|
202.1.96.0/24	11001010.00000001.011**00000**.00000000
202.1.97.0/24	11001010.00000001.011**00001**.00000000
202.1.98.0/24	11001010.00000001.011**00010**.00000000
⋮	⋮
202.1.126.0/24	11001010.00000001.011**11110**.00000000
202.1.127.0/24	11001010.00000001.011**11111**.00000000
202.1.128.0/24	11001010.00000001.100**00000**.00000000
202.1.129.0/24	11001010.00000001.100**00001**.00000000
⋮	⋮
202.1.158.0/24	11001010.00000001.100**11110**.00000000
202.1.159.0/24	11001010.00000001.100**11111**.00000000
	|←..second prefix..→|

Note that this set of 64 24-bit blocks cannot be summarised as a single 18-bit block

202.1.96.0/19 11001010.00000001.**011**00000.00000000

202.1.128.0/19 11001010.00000001.**100**00000.00000000

So the two 19-bit blocks are:

202.1.96.0/19

202.1.128.0/19

Could the answer be 202.1.96.0/16? No, because that includes $2^{16} = 65536$ different addresses, not just the $2^8 \times 64 = 16384$ addresses that we are taking care of.

General Approach Applied to This Problem:

1. Determine which octet the prefix will end in. Here, we have $2^8 \times 64 = 16{,}384$ addresses, so we have the prefix ending in the third octet.

2. Convert that octet only from the first and last address, to binary. So here, we convert $96_{10} \rightarrow 0110\,0000_2$ and $159_{10} \rightarrow 1001\,1111_2$.

3. Do these binary numbers have a common prefix, to the right of which all bits count from 000...000 to 111...111? Well, in this case, no, so...

4. Find the power of two over which the third octet counts. Here, the power of 2 is $128 = 7^5$. Convert the value before and after the power of 2 to binary: $2^7 - 1 = 0111\,1111_2$, and $2^7 = 1000\,0000_2$.

5. Now compare the third octet from the first address block, $96_{10} \rightarrow 0110\,0000_2$ with $2^7 - 1 = 0111\,1111_2$, and see if we have a common prefix, to the right of which all bits count from 000...000 to 111...111. Well, yes we do! It is $011|xxxxx$.

6. Now compare $2^7 = 1000\,0000_2$ with $159_{10} \rightarrow 1001\,1111_2$. Can we see a common prefix, with bits to the right counting from all 0's to all 1's? Yes! It is $100|xxxxx$.

Some Other Points:

- The prefixes do not all have to be the same size.
- In the two examples given here, we only needed to convert four eight-bit numbers to binary, *not* sixty-four 32-bit numbers.
- You may have to continue to divide these groups of addresses until you find a single address block. In other words, you may need to apply the above steps recursively until you find all the address blocks.
- If you want to see a computer algorithm for doing this, see the `compact()` method in the Perl module NetAddr::IP, from CPAN.
- You can always make a simple sanity check by calculating the number of host addresses in the input, and making sure that it matches the number in the summarised output.

2 Questions

1. (a) How many 24-bit network blocks are available within the CIDR block 200.56.168.0/21?
(Hint: how many times does 2^{32-24} divide into 2^{32-21}? Hmm, $2^{32-21-(32-24)} = 2^{24-21}$) . . . .

(b) List them.

2. Aggregate the following 24-bit blocks into as few blocks as possible:

212.56.132.0/24
212.56.133.0/24
212.56.134.0/24
212.56.135.0/24

(Hint: determine the prefix common to them all).

3. Aggregate the following 24-bit blocks into as few blocks as possible:

212.56.146.0/24
212.56.147.0/24
212.56.148.0/24
212.56.149.0/24

4. Here is a quote from an email:

> I'm thinking if we allocate, say 48 groups of 8-bit address space to you, let's say, from 172.19.16.x – 172.19.63.x, would it solve your problem ? The point is, if you agree on such an arrangement, we don't have to ask for outside help than CC/IVE(TY) as 172.x.x.x are solely allocated to us. What's your opinion ?

Aggregate the following 24-bit blocks into as few blocks as possible:

172.19.16.0/24

172.19.17.0/24

172.19.18.0/24

⋮

172.19.62.0/24

172.19.63.0/24

✐

2.1 Routers and Address Allocation

1. In the example problem given in the lecture (see figure 1 on the following page), the addresses were allocated, but the routes advertised by each router were not

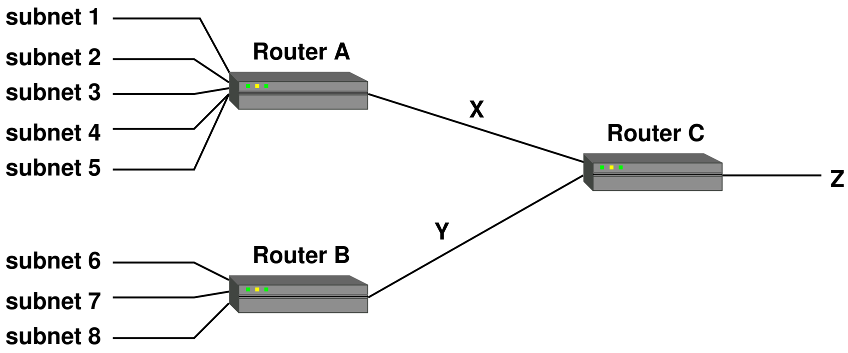

Figure 1: The routing problem from the lecture.

determined. Using the addresses given in the lecture, what routes does Router A advertise at X, and Router B advertise at Y, and Router C advertise at Z?

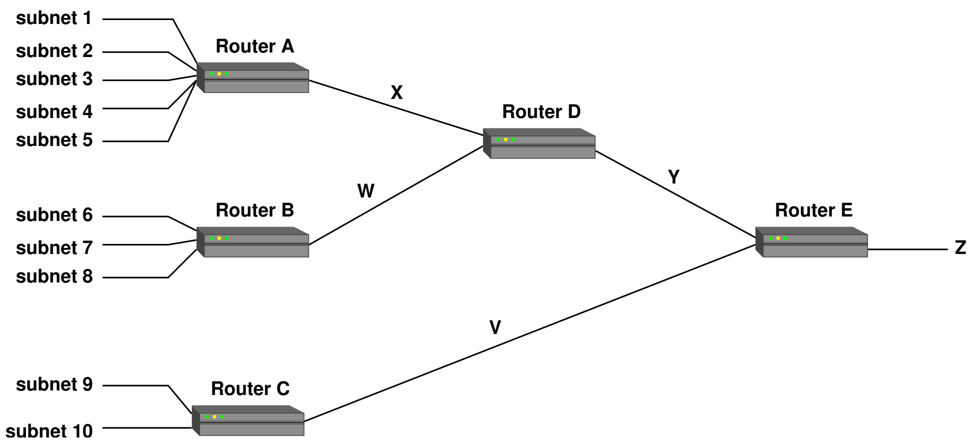

Figure 2: A network with five routers and fourteen subnets.

2. Figure 2 shows a network with 5 routers and 14 subnets. You may select IP addresses from the two blocks of addresses 172.12.0.0/19 and 192.168.0.0/27. You must leave at least one quarter of these addresses available for other purposes. The requirements are that each of subnets 1, 2,. . . , 8 must support up to 128 computers, while subnets 9 and 10 must each support up to 520 computers.

(a) Allocate a suitable block of addresses to each of the fourteen subnets that will allow maximum route aggregation. (Do not include link Z).

✐

(b) Given your selection in the previous part, with route summarisation *disabled* on all the routers, list the routes that would be advertised by router A at X, by router B at W, by router C at V, by router D at Y, and by router E at Z.

✐

(c) What would be a necessary requirement for the routers to support route summarisation?

✐

(d) Repeat part 2b, but for the case where route summarisation is enabled on all routers.

✐