Computer Networks - CS 204

Rishit Saiya - 180010027, Assignment - 3

April 15, 2020

1 IP Address and CIDR

1. For class C address, size of network field is 24 bits, the total number of networks possible is 2^{21}.

2. The network 198.78.41.0 is a Class C network.

3. In a classful addressing, first three bits in Class C IP address are 110.

4. Given n as the number of bits in the host ID, the number of addresses usable for addressing specific host in each network is $2^n - 2$. Those 2 are reserved hosts, 0.0.0.0 & 255.255.255.255 and aren't used in the network.
   After converting the subnet mask to binary, i.e. 11111111.11111111.11111000.00000000, we see that here the n is 11, hence the maximum number of hosts for a subnet is 2046.

5. Given the subnet, a set of IP addresses belong to same network if and only if their bit wise AND operation with the given subnet computes to same Subnet ID.
   Since the subnet is 255.255.31.0 and its binary is 11111111.11111111.00011111.00000000, so AND operation with first 12 bits of Subnet and any IP address would obviously yield the same bits of the IP address. So option A and C are eliminated. When we check for the remaining 2 options by AND operation, we obtain the following result:
   (10.35.28.2) AND (255.255.31.0) = 10.35.28.0
   (10.35.29.4) AND (255.255.31.0) = 10.35.29.0
   This concludes that they don't belong to same network. We check for last remaining option.
   (128.8.129.43) AND (255.255.31.0) = 128.8.1.0
   (128.8.161.55) AND (255.255.31.0) = 128.8.1.0
   Hence they belong to the same network.

6. The given IP address is 125.134.112.66 and corresponding Subnet is 255.255.224.0, the Network ID is
   (125.134.112.66) AND (255.255.224.0) = 125.134.96.0
   The network address is 125.134.96.0.

Destination	Distance Value (which is the link cost value)	Next Hop
A	0	A
B	4	B
C	∞	-
D	∞	-
E	∞	-

Table 1: Routing Table A (Initial)

Destination	Distance Value (which is the link cost value)	Next Hop
A	4	A
B	0	B
C	1	C
D	∞	-
E	5	E

Table 2: Routing Table B (Initial)

7. The network ID for the IP address (172.16.0.0) and Subnet (255.255.0.0) is (172.16.0.0) AND (255.255.0.0) = 172.16.0.0
   Putting 1 in all host ID, will give us Broadcast address of the given IP address. The Network ID is 10101100.00010000.00000000.00000000, so the broadcast address is 10101100.00010000.11111111.11111111 which is equivalent to 172.16.255.255.

2 Distance Vector Routing and Link State Routing

8. Initial Routing Tables are as follows:

   Routing Table A: Mentioned as in Table 1

   Routing Table B: Mentioned as in Table 2

   Routing Table C: Mentioned as in Table 3

   Routing Table D: Mentioned as in Table 4

Destination	Distance Value (which is the link cost value)	Next Hop
A	∞	-
B	1	B
C	0	C
D	7	D
E	∞	-

Table 3: Routing Table C (Initial)

Destination	Distance Value (which is the link cost value)	Next Hop
A	∞	-
B	∞	-
C	7	C
D	0	D
E	2	E

Table 4: Routing Table D (Initial)

Routing Table E: Mentioned as in Table 5

After 1^{st} Iteration the following are the routing tables of the routers:

Routing Table A: Mentioned as in Table 6

Routing Table B: Mentioned as in Table 7

Routing Table C: Mentioned as in Table 8

Destination	Distance Value (which is the link cost value)	Next Hop
A	∞	-
B	5	B
C	∞	-
D	2	D
E	0	E

Table 5: Routing Table E (Initial)

Destination	Distance Value (which is the link cost value)	Next Hop
A	0	A
B	4	B
C	∞	-
D	∞	-
E	∞	-

Table 6: Routing Table A (1^{st} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	4	A
B	0	B
C	1	C
D	7	E
E	5	E

Table 7: Routing Table B (1^{st} Iteration)

Routing Table D: Mentioned as in Table 9

Routing Table E: Mentioned as in Table 10

After 2^{nd} Iteration the following are the routing tables of the routers:

Routing Table A: Mentioned as in Table 11

Destination	Distance Value (which is the link cost value)	Next Hop
A	5	B
B	1	B
C	0	C
D	7	D
E	6	B

Table 8: Routing Table C (1^{st} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	∞	-
B	7	E
C	7	C
D	0	D
E	2	E

Table 9: Routing Table D (1^{st} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	9	B
B	5	B
C	6	B
D	2	D
E	0	E

Table 10: Routing Table E (1^{st} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	0	A
B	4	B
C	5	B
D	11	B
E	9	B

Table 11: Routing Table A (2^{nd} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	4	A
B	0	B
C	1	C
D	7	E
E	5	E

Table 12: Routing Table B (2^{nd} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	5	B
B	1	B
C	0	C
D	7	D
E	6	B

Table 13: Routing Table C (2^{nd} Iteration)

Routing Table B: Mentioned as in Table 12

Routing Table C: Mentioned as in Table 13

Routing Table D: Mentioned as in Table 14

Routing Table E: Mentioned as in Table 15

Destination	Distance Value (which is the link cost value)	Next Hop
A	11	E
B	7	E
C	7	C
D	0	D
E	2	E

Table 14: Routing Table D (2^{nd} Iteration)

Destination	Distance Value (which is the link cost value)	Next Hop
A	9	B
B	5	B
C	6	B
D	2	D
E	0	E

Table 15: Routing Table E (2^{nd} Iteration)

Destination	Shortest Path
B	A $\rightarrow$ B
C	A $\rightarrow$ B $\rightarrow$ C
D	A $\rightarrow$ B $\rightarrow$ E $\rightarrow$ D
E	A $\rightarrow$ B $\rightarrow$ E

Table 16: Router A (Shortest Path)

For the rest of the iterations, we observe that the iteration tables remain same and hence the convergence is achieved.

Shortest Path between every pair of nodes is as follows:

Router A: Mentioned as in Table 16

Router B: Mentioned as in Table 17

Router C: Mentioned as in Table 18

Router D: Mentioned as in Table 19

Destination	Shortest Path
A	B $\rightarrow$ A
C	B $\rightarrow$ C
D	B $\rightarrow$ E $\rightarrow$ D
E	B $\rightarrow$ E

Table 17: Router B (Shortest Path)

Destination	Shortest Path
A	C → B → A
B	C → B
D	C → D
E	C → B → E

Table 18: Router C (Shortest Path)

Destination	Shortest Path
A	D → E → B → A
B	D → E → B
C	D → C
E	D → E

Table 19: Router D (Shortest Path)

Router E: Mentioned as in Table 20

9. (1) (a) A → B → C → E
   (b) A → B → D

   (2) Vertex A is considered as the source vertex throughout the whole process of finding the shortest path.

   The implementation of Dijkstra's Algorithm is given in **Table 21**. The explanation is as follows:
   Each row in the table is an iteration of the vertex. As mentioned, A is the source vertex. Direct edges to B & C exist in the graph. So correspondingly, 5 & 10 are marked. Rest are marked ∞. For each iteration, we have to choose the vertex possessing the shortest distance from the previous selected vertex. In here, for the 1^{st} Iteration, out of {5,10,∞,∞}, 5 is minimum, so it underlined and corresponding B is selected as the selected vertex for the next iteration.

Destination	Shortest Path
A	E → B → A
B	E → B
C	E → B → C
E	E → D

Table 20: Router E (Shortest Path)

Selected Vertex	B	C	D	E
A (Source)	$\underline{\mathbf{5}}$	10	∞	∞
B	$\underline{\mathbf{5}}$	8	$\underline{\mathbf{7}}$	14
D	$\underline{\mathbf{5}}$	$\underline{\mathbf{8}}$	$\underline{\mathbf{7}}$	13
C	$\underline{\mathbf{5}}$	$\underline{\mathbf{8}}$	$\underline{\mathbf{7}}$	$\underline{\mathbf{9}}$
E	$\underline{\mathbf{5}}$	$\underline{\mathbf{8}}$	$\underline{\mathbf{7}}$	$\underline{\mathbf{9}}$

Table 21: Implementation of Dijkstra's Algorithm

In the next iteration, we iteratively use dist[v] $\leftarrow$ dist[u] + w(u,v) for every neighbour and if dist[v] $>$ dist[u] + w(u,v), all notations being standard. So in here, B to D, we replace ∞ by (5 + 2). Similarly E is filled by (9 + 5).

These iterations carried on till we get the whole row underlined, in which case we understand that we have done the iterations on all the vertices.

The final thing obtained is shortest distances from A to other vertices. Just tracing back the path, on where the shortest distances were changing will give us the Shortest Path.

Time Complexity:
Here V & E are number of vertices and edges respectively. The time complexity of Dijkstra's Algorithm is O(V^2). But when we implement it using the minimum priority queue, we reduce the complexity to O((E + V)$\times$log(V)). Also if we implement using Fibonacci Heap for Priority Queue, we can further decrease it to O(E + ($V \times logV$)).