CSEE W4119 Computer Networks

Homework 3

Yixing Chen, yc3094

1. The optimality principle of routing: This principle states that if a node J is on the optimal path from node I to node K, then the optimal path from J to K also falls along the same route. Prove it.

Answer:

Proof by contradiction:

Assume that we have the optimal path P from I to K. Node J is on the path, so we can split P into P1 (from I to J) and P2 (from J to K).

If there is another path P3 from J to K and this path is the optimal path from J to K. Then we can know that:

cost(P3) < cost(P2)

So cost(P1) + cost(P3) < cost(P1) + cost(P2)

And cost(P1) + cost(P2) = cost(P)

So cost(P1) + cost(P3) < cost(P)

Therefore, we can draw the conclusion that P isn't the optimal path from I to K because we can find another path which have less cost than P.

However, from the problem, we know that P is the optimal path from I to K, this is a contradiction. Thus, there is no another path P3 which has less cost than P2, the optimal path from J to K also falls along the same route.

2. Can you use the path vector in BGP as a distance metric? If not, why not?

Answer:

No. For the first reason, distance metric is a measure of the cost to reach a node, but the path vector just record the number of ASes in path, not the actual distance (cost) of the path, so the path vector cannot be used as a distance metric. And second, in BGP, path vectors are used to advertise the reachability of a network without loops, however, because BGP does not focus on the least-cost path between ASes, the path between two ASes in BGP is based on policy, not on the distance, thus there might be other smaller paths. Therefore, we cannot use the path vector in BGP as a distance metric.

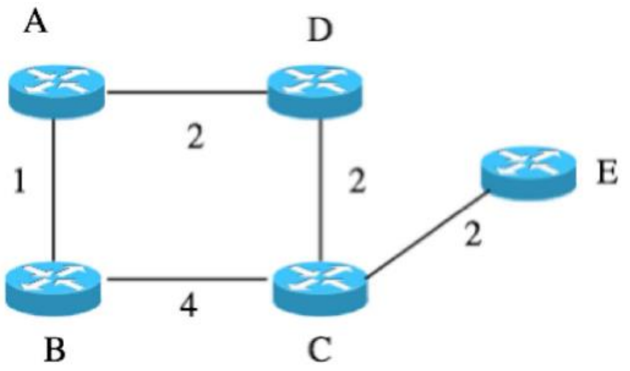
3. A set of IP addresses from 128.59.0.0 to 128.59.128.255 have been aggregated to 128.59.0.0/17 and assigned to nodes on campus. There is a gap of 1024 assigned addresses from 128.59.60.0 to 128.59.63.255. Now suddenly the block of addresses are assigned to an off-campus location, and thus nodes at that location have to be routed on a different outgoing line. Is it now necessary to split up the aggregate into its constituent blocks, add the new block into the table and then see if any re-aggregation is possible? If not, what can be done instead?

Answer:

No. It's not necessary.

Assume the on campus part is AS1, and we can encapsulate the off-campus block into a new autonomous system AS2. Then we can have two different prefixes to distinguish the two ASes because routers match the longest prefix. The first prefix is 128.59.0.0/17, which is corresponding to all nodes on campus (AS1), and another prefix is 128.59.60.0/22, which is corresponding to the nodes off-campus (AS2). According to the two prefixes, we can route correctly.

4. Consider the network shown in the figure. Assume a distance vector routing protocol is used. Give the state table for nodes A, B, C and D after the protocol has converged (i.e., a table displaying for every node X, the path cost to every other node in the network through every neighbor node of X). Give the state table under two scenarios, with and without poisoned reverse implemented. Now, the link CE breaks. Will the state tables converge at A, B, C and D? Consider both cases, with and without poisoned reverse being implemented.



Answer:

Without poisoned reverse being implemented, all nodes have the same table:

	A	B	C	D	E
A	0	1	4	2	6
B	1	0	4	3	6
C	4	4	0	2	2
D	2	3	2	0	4
E	6	6	2	4	0

Link CE breaks, state tables will not converge at A, B, C and D.

When CE breaks, there is no direct path from C to E, so

cost(CE) → cost(CD) + cost(DE) = 2+4 = 6

cost(DE) → cost(DA) + cost(AE) = 2+6 = 8

cost(AE) → cost(AB) + cost(BE) = 1+6 = 7

cost(BE) → cost(BA) + cost(AE) = 1+7 = 8

Then

cost(CE) → cost(CD) + cost(DE) = 2+8 = 10

cost(DE) → cost(DA) + cost(AE) = 2+7 = 9

cost(AE) → cost(AB) + cost(BE) = 1+8 = 9

cost(BE) → cost(BA) + cost(AE) = 1+9 = 10

Then

cost(CE) → cost(CD) + cost(DE) = 2+9 = 11

cost(DE) → cost(DA) + cost(AE) = 2+9 = 11

cost(AE) → cost(AB) + cost(BE) = 1+10 = 11

cost(BE) → cost(BA) + cost(AE) = 1+11 = 12

….

The cost of AE, BE, CE and DE are continually improving, so state tables will not converge.

With poisoned reverse being implemented, different nodes have different table

A:

	A	B	C	D	E
A	0	1	4	2	6
B	1	0	4	∞	6
C	4	4	0	2	2
D	2	∞	2	0	4
E	6	6	2	4	0

B:

	A	B	C	D	E
A	0	1	4	2	6
B	1	0	4	3	6
C	4	4	0	2	2
D	2	3	2	0	4
E	6	6	2	4	0

C:

	A	B	C	D	E
A	0	1	4	2	∞
B	1	0	4	3	∞
C	4	4	0	2	2
D	2	3	2	0	∞
E	6	∞	2	∞	0

D:

	A	B	C	D	E
A	0	1	∞	2	∞
B	1	0	4	3	6
C	∞	4	0	2	2
D	2	3	2	0	4
E	∞	6	2	4	0

Link CE breaks, state tables will not converge at A, B, C and D.

When CE breaks

cost(DE) → cost(DA) + cost(AE) = 2+∞ = ∞ (the value in D's table, then DE=∞ to A, DE=∞ to others)

cost(CE) → cost(CD) + cost(DE) = 2+∞ = ∞ (the value in C's table, then CE=∞ to D, E=∞ to others)

cost(BE) → cost(BC) + cost(CE) = 4+5 = 9 (the value in B's table, then BE=∞ to C, BE=9 to others)

cost(AE) → cost(AB) + cost(BE) = 1+9 =10 (the value in A's table, then AE=∞ to B, AE=10 to others)

Then

cost(DE) → cost(DA) + cost(AE) = 2+10 = 12

cost(CE) → cost(CD) + cost(DE) = 2+12 = 14

cost(BE) → cost(BC) + cost(CE) = 4+∞ = ∞

cost(AE) → cost(AB) + cost(BE) = 1+∞ = ∞

Then

cost(DE) → cost(DA) + cost(AE) = 2+∞ = ∞

cost(CE) → cost(CD) + cost(DE) = 2+∞ = ∞

cost(BE) → cost(BC) + cost(CE) = 4+14 = 18

cost(AE) → cost(AB) + cost(BE) = 1+18 = 19

…

Therefore, there are always 2 costs continually improving, and other 2 costs are ∞, so state tables will not converge.