Network Layer: Control Plane

Annalise Tarhan

December 17, 2020

1 Looking at figure 5.3, enumerate the paths from y to u that do not contain any loops.

y-x-u y-x-v-u y-x-w-u y-x-w-v-u y-x-v-w-u
y-w-u y-w-x-u y-w-v-u y-w-v-x-u y-w-x-v-u
y-z-w-u y-z-w-x-u y-z-w-v-u y-z-w-v-x-u y-z-w-x-v-u

2 Repeat the previous problem for paths for:

2.1 x to z

x-y-x x-w-z
x-y-w-z x-w-y-z x-v-w-z x-u-w-z
x-u-v-w-z x-v-w-y-z x-v-u-w-z x-u-v-w-z x-u-w-y-z
x-v-u-w-y-z x-u-v-w-y-z

2.2 z to u

z-w-u z-w-v-u z-w-x-u z-w-x-v-u z-w-v-x-u
z-y-x-u z-y-x-v-u z-y-x-w-u z-y-x-w-v-u z-y-x-w-v-u
z-y-w-u z-y-w-v-u z-y-w-x-u z-y-w-x-v-u z-y-w-v-x-u
z-w-y-x-u z-w-y-x-v-u z-y-x-v-w-u z-y-x-w-v-u

2.3 z to w

z-w z-y-w z-y-x-w z-y-x-v-w z-y-x-u-w z-y-x-v-u-w z-y-x-u-v-w

3 Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to table 5.1.

t: x-v-t   u: x-v-u   v: x-v   w: x-w   y: x-y   z: x-z

Step	N'	D(t), p(t)	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(y), p(y)	D(z), p(z)
0	x	∞	∞	3, x	6, x	6, x	8, x
1	xv	7, v	6, v		6, x	6, x	8, x
2	xvu	7, v			6, x	6, x	8, x
3	xvuw	7, v				6, x	8, x
4	xvuwy	7, v					8, x
5	xvuwyt						8, x
6	xvuwytz						

4 Consider the network in the previous problem. Using Dijkstra's algorithm, create a table to solve the following:

4.1 Compute the shortest path from t to all network nodes.

Step	N'	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	t	2, t	4, t	∞	∞	7, t	∞
1	tu		4, t	5, u	∞	7, t	∞
2	tuv			5, u	7, v	7, t	∞
3	tuvw				7, v	7, t	∞
4	tuvwx					7, t	15, x
5	tuvwxy						15, x
6	tuvwxyz						

4.2 Compute the shortest path from u to all network nodes.

Step	N'	D(t), p(t)	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	3, u	3, u	∞	∞	∞
1	ut		3, u	3, u	∞	9, t	∞
2	utv			3, u	6, v	9, t	∞
3	utvw				6, v	9, t	∞
4	utvwx					9, t	14, x
5	utvwxy						14, x
6	utvwxyz						

4.3 Compute the shortest path from v to all network nodes.

Step	N'	D(t), p(t)	D(u), p(u)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	v	4, v	3, v	4, v	3, v	8, v	∞
1	vt		3, v	4, v	3, v	8, v	∞
2	vtu			4, v	3, v	8, v	∞
3	vtux			4, v		8, v	11, x
4	vtuxw					8, v	11, x
5	vtuxwy						11, x
6	vtuxwyz						

4.4 Compute the shortest path from w to all network nodes.

Step	N'	D(t), p(t)	D(u), p(u)	D(v), p(v)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	w	∞	3, w	4, w	6, w	∞	∞
1	wu	5, u		4, w	6, w	∞	∞
2	wuv	5, u			6, w	12, v	∞
3	wuvt				6, w	12, v	∞
4	wuvtx					12, v	14, x
5	wuvtxy						14, x
6	wuvtxyz						

4.5 Compute the shortest path from y to all network nodes.

Step	N'	D(t), p(t)	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(z), p(z)
0	y	7, y	∞	8, y	∞	6, y	12, y
1	yx	7, y	∞	8, y	12, x		12, y
2	yxt		9, t	8, y	12, x		12, y
3	yxtv		9, t		12, x		12, y
4	yxtvu				12, x		12, y
5	yxtvuw						12, y
6	yxtvuwz						

4.6 Compute the shortest path from z to all network nodes.

Step	N'	D(t), p(t)	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)
0	z	∞	∞	∞	∞	8, z	12, z
1	zx	∞	∞	11, x	14, x		12, z
2	zxv	15, v	14, v		14, x		12, z
3	zxvy	15, v	14, v		14, x		
4	zxvyu	15, v			14, x		
5	zxvyuw	15, v					
6	zxvyuw						

5 Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.

	u	v	x	y	z
u	∞	∞	∞	∞	∞
v	∞	∞	∞	∞	∞
x	∞	∞	∞	∞	∞
y	∞	∞	∞	∞	∞
z	∞	6	2	∞	0

	u	v	x	y	z
u	0	1	∞	2	∞
v	1	0	3	∞	6
x	∞	3	0	3	2
y	2	∞	3	0	∞
z	7	5	2	5	0

	u	v	x	y	z
u	0	1	4	2	6
v	1	0	3	3	5
x	4	3	0	3	2
y	2	3	3	0	5
z	6	5	2	5	0

6 Consider a general topology and a synchronous version of the distance-vector algorithm. Suppose that at each iteration, a node exchanges its distance vectors with its neighbors and receives their distance vectors. Assuming that the algorithm begins with each node knowing only the costs to its immediate neighbors, what is the maximum number of iterations required before the distributed algorithm converges?

The worst case scenario is a chain, where each node has at most two neighbors. Information would spread slowly, one node at a time, albeit in both directions. For a chain of n nodes, it would take $n-2$ iterations for the cost of the final link to reach the first node, assuming the costs are symmetrical. (Otherwise, it would take $n-1$ iterations.) Consider a chain of five nodes, 0 through 4. The cost of the last link in the chain is known by node 3 and node 4. On the first iteration, that information would be sent to node 2. On the second, node 2 would tell node 1. On the third, the first node, node 0, would also be informed. At each iteration, each node updates its distance vector appropriately. Therefore, for a chain of $n = 5$ nodes, it would take $n - 2 = 5 - 2 = 3$ iterations for the distributed algorithm to converge.

7 Consider the network fragment shown below. x has only two attached neighbors, w and y. w has a minimum-cost path to destination u of 4, and y has a minimum-cost path to u of 6.

7.1 Give x's distance vector for destinations w, y, and u.

D(w)=2, D(y)=4, D(u)=7

7.2 Give a link-cost change for either $c(x, w)$ or $c(x, y)$ such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

c(x,y)=1

7.3 Give a link-cost change for either $c(x, w)$ or $c(x, y)$ such that x will not inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

c(x,y) = 10

8 Consider the three-node topology in figure 5.6. Rather than having the link costs shown, the link costs are $c(x, y) = 3$, $c(y, z) = 6$, $c(z, x) = 4$. Compute the distance tables after the initialization step and after each iteration of a synchronous version of the distance-vector algorithm.

Node x table

	x	y	z		x	y	z
x	0	3	4		0	3	4
y	∞	∞	∞		3	0	6
z	∞	∞	∞		4	6	0

Node y table

	x	y	z	x	y	z
x	∞	∞	∞	0	3	4
y	3	0	6	3	0	6
z	∞	∞	∞	4	6	0

Node z table

	x	y	z	x	y	z
x	∞	∞	∞	0	3	4
y	∞	∞	∞	3	0	6
z	4	6	0	4	6	0

9 Consider the count-to-infinity problem in distance-vector routing. Will the count-to-infinity problem occur if we decrease the cost of a link? Why? How about if we connect two nodes which do not have a link?

No, neither of these situations will result in a count-to-infinity problem, which only occurs when two nodes turn towards each other in search of a next-best option after their old path's weight increased, possibly infinitely. A decrease in weight, on the other hand, could only cause them to turn away from each other towards the now cheaper link. Turning away, as opposed to turning towards, cannot cause a loop.

10 Argue that for the distance-vector algorithm in figure 5.6, each value in the distance vector D(x) is non-increasing and will eventually stabilize in a finite number of steps.

From the initial state where x is only aware of the costs to its immediate neighbors through each iteration, the values in $D(x)$ cannot increase, since values only change when a path with a lower weight is found. Given a finite number of nodes and a finite number of links with static weights, after a finite number of iterations, all of the link information will have propagated through the entire network and each link will know which link to use to reach any other node in the network. Once every node has that information, the system is stable.

11 Consider figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows: $c(x, y) = 4$, $c(x, z) = 50$, $c(y, w) = 1$, $c(z, w) = 1$, $c(y, z) = 3$. Suppose that poisoned reverse is used in the distance-vector routing algorithm.

11.1 When the distance vector routing is stabilized, routers w, y, and z inform their distances to x to each other. What distance values do they tell each other?

To w: $D_y(x) = 4$ $D_z(x) = \infty$
To y: $D_w(x) = \infty$ $D_z(x) = 6$
To z: $D_w(x) = 5$ $D_y(x) = 4$

11.2 Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again?

Yes, it will cause a loop and a count-to-infinity problem. The poisoned reverse method prevents loops between two nodes, but this loop involves three nodes and three steps.
1. When the link cost between x and y increases, y's new (perceived) shortest path routes through z, since z advertises a distance of six. y believes its new weight is nine, z's advertised distance plus the three from the link between them.
2. y then advertises a distance of nine to w, but infinity to z, since it uses z as its next hop. w then increases its own distance to x to 10, y's advertised distance plus one.
3. Finally, w informs z of its new distance (telling y infinity instead), and z updates its own distance to 11.
This cycle continues until w advertises to z a distance to x that is higher than z's direct link to x: 50. Since each loop increases the calculated distance by five, it will take nine full loops, or 27 iterations, for z to switch its next-hop node to x. It will take another three loops for the entire system to update itself and reach a stable state.

11.3 How do you modify $c(y, z)$ such that there is no count-to-infinity problem at all if $c(y, x)$ changes from 4 to 60?

Any value of $c(y, x)$ greater than 54 would cause y to immediately re-route to x directly.

12 Describe how loops in paths can be detected in BGP.

BPG routes consist of the prefix as well as BGP attributes, including AS-PATH, a list of ASes the route passes through. If an AS sees itself on the AS-PATH, it knows that the route is looping.

13 Will a BGP router always choose the loop-free route with the shortest AS-PATH length?

No. Its first priority is to choose a route with the highest local preference, which is determined by the administrator based on policy. Only if two or more routes have equal local preference will the tie be broken by AS-PATH length.

14 Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.

14.1 Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?

eBGP, since it is an external router and directly connected to the AS containing x.

14.2 Router 3a learns about x from which routing protocol?

iBGP, since it is an internal router.

14.3 Router 1c learns about x from which routing protocol?

eBGP, since it is the only external router in its AS, so it must be the first router in its AS informed

14.4 Router 1d learns about x from which routing protocol?

iBGP, since it is an internal router

15 Referring to the previous problem, once router 1d learns about x it will put an entry (x, l) in its forwarding table.

15.1 Will l be equal to l_1 or l_2 for this entry?

l_1. There is only one possible AS-PATH to choose from, so hot-potato routing is used to choose the route to the external router, 1c.

15.2 Now suppose there is a physical link betweeen AS2 and AS4. Suppose router 1d learns that x is accessible via AS2 as well as AS3. Will l be set to l_1 or l_2?

Assuming there is no difference in local preference between AS2 and AS3, l will be set to l_2. Absent a difference in local preference or AS-PATH length, routers use hot potato routing to choose the fastest route out of its own AS, and l_2 allows to to exit the AS after only one hop through 1b.

15.3 Now suppose there is another AS, AS5, which lies on the path between AS2 and AS4. Suppose router 1d learns that x is accessible via AS2 AS5 AS4 as well as via AS3 AS4. Will l be set to l_1 or l_2?

Now, the AS-PATH through AS2 is longer than the one through AS3. Since choosing a shorter AS-PATH is a higher priority than shortest route out of the AS, l will be set to l_1.

16 Consider the following network. ISP B provides national backbone service to regional ISP A. ISP C provides national backbone service to regional ISP D. Each ISP consists of one AS. B and C peer with each other in two places using BGP. Consider traffic going from A to D. B would prefer to hand that traffic over to C on the West Coast, while C would prefer to get the traffic via its East Coast peering point with B. What BGP mechanism might C use, so that B would hand over A-to-D traffic at its East Coast peering point?

ISP C might choose not to advertise its cross-country service to ISP B at its West Coast connection.

17 In figure 5.13, consider the path information that reaches stub networks W, X, and Y. Based on the information available at W and X, what are their respective views of the network topology?

W's view and X's view are the same. They both show the complete topology, except for the link between B and C.

18 Consider figure 5.13. B would never forward traffic destined to Y via X based on BGP routing. But there are some very popular applications for which data packets go to X first and then flow to Y. Identify one such application, and describe how data packets follow a path not given by BGP routing.

A P2P file sharing application would route traffic in a different way, because packets flow through a cluster of peers as opposed to a single hub. A file received at X from W might then be sent from X to Y, causing those packets to take routes not used by the server-client model.

19 In figure 5.13, suppose there is another stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, and A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

In this case, A would advertise AW and AV to B, but only AV to C. From C's perspective, its route to V is CAV and its route to W is CBAW.

20 Suppose ASes X and Z are not directly connected but instead are connected by AS Y. Further suppose that X has a peering agreement with Y, and that Y has a peering agreement with Z. Finally, suppose that Z wants to transit all of Y's traffic but does not want to transit X's traffic. Does BGP allow Z to implement this policy?

Yes. There is a BGP attribute called NO-EXPORT that prevents the receiving AS, in this case Y, from advertising the AS that attached it, Z, to other ASes, X. Y will know about its route to Z and will be able to send its own traffic, but since it will not re-advertise the route to X, it will never be asked to transit X's traffic to Z.

21 Consider the two ways in which communication occurs between a managing entity and a managed device: request-response mode and trapping. What are the pros and cons of these two approaches, in terms of:

21.1 Overhead

Request-response mode requires two messages sent for every piece of information. Trapping only requires one. However, request-response is the only way for the managing entity to direct the behavior of the managed device, so the lower overhead is not especially relevant.

21.2 Notification time when exceptional events occur

Trapping has the advantage because messages are unsolicited. The managed device can send the message immediately, not waiting for the managing entity to inquire.

21.3 Robustness with respect to lost messages between the managing entity and the device

Request-response is better because the managing entity will be able to detect if a message has been lost, although it will not be able to differentiate between a lost request and a lost response. In either case, it will repeat the request.

Since trap requests have no required response, a lost message would not be detected by either side. Since PDU messages are typically carried via UDP, these distinctions are important.

22 We saw that it was prefereable to transport SNMP messages in unreliable UDP datagrams. Why do you think the designers of SNMP chose UDP rather than TCP as the transport protocol of choice for SNMP?

The overhead required to maintain TCP connections would outweigh the cost of re-sending lost request-response messages. Surprisingly, the potential for lost and non-recovered trap messages was considered acceptable. SNMP messages are also most needed while the network is stressed, and using TCP would cause the devices' sending rates to be throttled at exactly the wrong times.