6.263/16.37 Data Communication Networks October 4, 2005
MIT, Fall 2005

Solutions to Problem Set 2

Problem 2.1

(a) Refer to the algorithm statement in the text. Let B be the lower bound on the arc lengths. Then in step 1, each node $i \notin P$ with

$$D_i \leq \min_{j \notin P}\{D_j\} + B$$

can be added to P. To see why this is correct, recall that, at the start of each iteration, D_i for $i \notin P$ is the shortest distance from 1 to i for which all nodes on the path except i lie in P. The inductive argument which proved Dijkstra's algorithm required that each node added to P must have a shortest path for which all but the final node lie in P. This must be true for each node i which meets the above condition, since any path which included another node not in P would have a length of at least D_i.

(b) Assume that the shortest paths from node 1 to all other nodes have been found and have lengths D_j^*, $j \neq 1$. If link (i,k) increases in length, paths which do not traverse link (i,k) are not affected by this change. Therefore, we can initialize Dijkstra's algorithm as

$$P = \{j \text{ such that a shortest path from 1 to } j \text{ does } \textit{not} \text{ traverse arc } (i,k)\}$$

$$D_j = \begin{cases} D_j^* & \text{for all } j \in P \\ \min_{l \in P}[D_l + d_{lj}] & \text{for all } j \notin P \end{cases}$$

and continue with the ordinary algorithm.

Problem 2.2

You will get the answers simply by counting the number of rows in the results of traceroute. (You may not be able to get a valid result for www.cnn.com.)

Destination	Number of Hops	Avg. Delay
www.yahoo.com	9 hops	about 12ms
www.cnn.com	30 hops	No answer (routers/machines do not answer ICMP msg)
www.berkeley.edu	17 hops	about 93ms

Note the above are round trip delays, i.e., the time form when you send a packet from no-knife until you receive a reply from the destination.

Problem 2.3

(a) Yes. I-BGP is used to advertise external routes inside the domain. OSPF is the intra-domain routing protocol, and is needed to discover internal routes in the AS.

(b) TCP is used to provide reliability. Remember that BGP announces changes to routes. Thus, it needs to make sure that the previous announcements were received by the neighbor BGP router.

(c) Note that *split horizon* is the idea that when a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. However, we will show that even this can fail in the given situation.

1) When link breaks, C marks D as unreachable and reports that to A and B.
2) Suppose A learns it first. A now thinks best path to D is through B.
3) A reports D unreachable to B and a route of cost 3 to C.
4) C thinks D is reachable through A at cost 4 and reports that to B.
5) B reports a cost 5 to A who reports new cost to C.
6) The process keeps going on indefinitely.

Problem 2.4

(a) B and C are peers. Y is a customer of C. From the perspective of C, there are no interesting routes that it can learn from B. B will advertise to C its route to X. But C will not advertise this route to Y because it prefers to advertise to Y its direct route to X (which does not traverse B). C could have advertised to Y the route to B's internal network, but (given the figure) it decided not to do so. There might be many reasons for that. First, B is a provider network, and thus B's network does not contain end hosts. Customers are usually interested in accessing end hosts, and thus advertising B to Y is not important. Further, Y has only one ISP which is C, thus Y will have a default route to C, i.e., any packet addressed for a destination for which Y has no route will go to C.

(b) C advertises to X its route to Y and its route to W, which traverses A. Similarly, B advertises to X its route to Y, which traverses C, and its route to W, which traverses A. It is unlikely that C advertises B's internal network to X, or that B advertises C's internal network to X, because these are provider networks with no end hosts. (It is also correct to say that C and B advertise their peer network to X.)

(c) X will see the whole network.

Problem 2.5

(a) Let us look at an example. Say at $t = 0$, every AS is using his direct route to $AS0$. The ASes advertise the route they are using to their neighbors. Now each AS will want to move to the indirect route because the indirect route has higher preference. But moving from the direct route to the indirect route will require the AS to withdraw the direct path. If the ASes are synchronized they will keep switching back and forth between the direct and indirect route. On the other hand, likely one of the ASes will manage to move first, say it is AS1. Now, AS1 moves to the route $1, 3, 0$. But AS3 will also move to the route $3, 2, 0$ because it is preferred to the direct route. But because AS3 have changed its route, AS1 has to move back to the direct route, which will cause AS2 to move to $2, 1, 0$, which will cause AS3 to go back to the direct route, etc . It is relatively easy to prove that no path stable route assignment exists, either by contradiction or by listing all of the possible route combinations (a maximum of 6) and proving that none of them is stable.

(b) In contrast to persistent loops, transient loops will be resolved once the protocol converges. Further, in transient loops the route itself loops, while in persistent loops, the route is unstable but does not loop.

(c) BGP can show transient loops before converging (assume the policy is to follow the shortest path then BGP is the same as Distance Vector). No, OSPF and RIP do not show persistent loops because there is always a stable route assignment that once we converge to it, the routes stop changing (as long as there is no failure).

(d) Use shortest path policies. Or use the customer provider policies and valley free routes (as long as there is no loop in the customer-provider graph, the routes will be stable)

Problem 2.6

In the algorithm, for $j \notin P$, D_j is the minimum 1 hop distance from j to a member of P, and a_j is the member of P for which this minimum is obtained.

To show that this implements the Prim-Dijkstra algorithm, we must show that the graph defined by $G(P, T)$ is a fragment of an MST, and that this fragment is enlarged at each iteration by the addition of a minimum weight outgoing arc. Then, by Proposition 1 in section 5.2.2, G will eventually be an MST.

Assume that P contains k nodes, that a_j is the closest member of P to j, and that $D_j = w_{ja_j}$ for $j \notin P$. Then step 1 chooses the minimum weight outgoing arc, and step 2 reestablishes the above assumptions about a_j and D_j for the new set P. The hypothesis is clearly true for $k = 1$ and by induction is true for all k.

Each iteration of steps 1 and 2 requires a number of operations proportional to the number of nodes i: $i \notin P$. The algorithm terminates in $N - 1$ iterations. Hence, $O(N^2)$ operations are required.

Problem 2.7

Any graph with equal edge weights will have such property.

Problem 2.8

MSTs are identical if $a > 0$ (proof by contradiction), otherwise they may be different. Shortest paths are identical if $a > 0$ and $b = 0$.