Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.02 Fall 2011

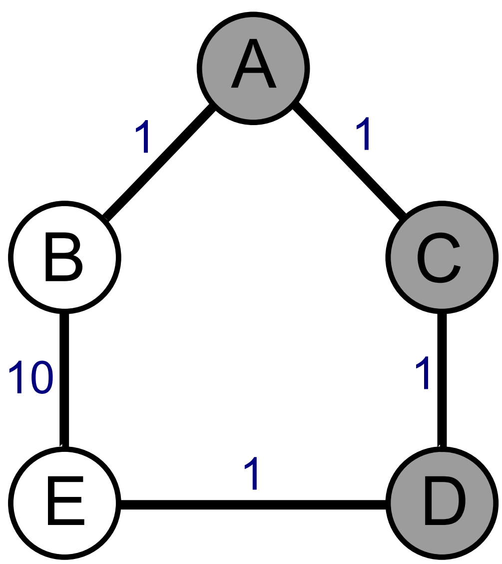
Solutions to Chapter 18

Updated Nov. 2011

---

1. Dijkstra's algorithm is a greedy algorithm. We add nodes in order of non-decreasing cost of the minimum-path to the nodes. We get:
   A: Order 1, Cost 2.
   E: Order 2, Cost 3.
   B: Order 3, Cost 4.
   C: Order 4, Cost 5.
   D: Order 5, Cost 6.

2. In the picture below, the grey nodes (A in particular) run Bob's algorithm (shortest number of hops), while the white nodes (B in particular) run Alice's (minimum-cost).

   Suppose the destination is E. A will pick B as its next hop because ABE is the shortest path. B will pick A as its next hop because BACDE is the minimum-cost path (cost of 4, compared to 11 for the ABE path). The result is a routing loop ABABAB...

3. (a) Yes. Any shortest path in G is also a shortest path in G' because if the cost of a path P in G is c, then the cost of P in G' is kc, for all P. Also, there's a one-to-one correspondence between paths in G and G'.

   (b) No. A counter-example is easy to construct. For example, suppose G is a triangle, A, B, C, where cost$(AB) = 1$, cost$(BC) = 1$, and cost$(CA) = 3$. The shortest path between A and C in G is A-B-C, with cost 2. But now suppose $k = 1$ and $h = 2$. The link costs become cost$(AB) = 4$, cost$(BC) = 4$, and cost$(AC = 5)$. Now, the shortest path between A and C is the direct link, AC.

Hari Balakrishnan

4. *Statements (a) and (b) are false, statement (c) is true, and statement (d) is false.* Statement (a) is false because u could propagate an incorrect cost to its neighbors causing the neighbor to have an incorrect route. In fact, u's neighbors could do the same. Statement (c) is correct; a simple example is when the network is a tree, where there is exactly one path between any two nodes.

   Statement (d) is false; no routing loops can occur under the stated condition. We can demonstrate this property by contradiction. Consider the shortest path from any node s to any other node t running the flawed routing protocol. If the path does not traverse u, no node on that path can have a loop because distance vector routing without any packet loss or failures is loop-free. Now consider the nodes for which the computed paths go through u; all these nodes are correctly implemented except for u, which means the paths between u and each of them is loop-free. Moreover, the path to u is itself loop-free because u picks one of its neighbors with *smaller* cost, and there is no possibility of a loop.

5. Reverse-engineering routing trees: See PSet #9.

6. This question asks for the update rule in the Bellman-Ford integration step. The cost in S's routing table for D should be set to $\min_i\{c_i + p_i\}$.

7. FishNet: See PSet #9.

8. (a) **B,C,E,A,F,D** and **B,C,E,F,A,D**.
   (b) **No effect.** The edge AC is not in any shortest path.
   (c) **Can affect route to A**. If $cost_{AC} \leq 3$, then we can start using this edge to go to A instead of the edge BA.
   (d) **Can affect route to C,F,E**. If $cost_{BC} \geq 7$, then we can use BE-EC to go to C instead of BC. If $cost_{BC} \geq 5$, then we can use BE-EF to go to F. If $cost_{BC} \geq 3$, can use BE to go to B.
   (e) **Can affect route to D**. If $cost_{BC} \leq 1$, then we can use BC-CE-ED to go to D instead of BD.