The Network Layer: Control Plane

Michael Brodskiy
Professor: E. Bernal Mor

November 13, 2023

- Network-Layer Functions
  - Forwarding (data plane)
  - Routing: determine route taken by packets from source to destination (control plane)
    - Two approaches to structuring a network control plane:
      - Per-router plane (traditional)
      - Software-defined
- Per-Router Control Plane
  - Individual routing algorithm components in each and every router interact in the control plane
- Logically Centralized Control Plane (SDN)
  - Remote controller computers, installs forwarding tables (aka flow tables) in routers)
- Routing Protocols
  - Routing protocol goal: determine "good" paths (equivalently, routes) from sending hosts to receiving hosts, through network of routers
    - Path: sequence of routers that packets traverse from given initial source host to destination host
    - "Good": least "cost", "fastest", "least congested"
    - Routing is a top networking challenge
- Graph Abstraction: Link Costs
  - $c_{a,b}$ is the cost of a direct link connecting a and b

        * Cost is defined by network operator: could always be 1, or inversely related to link capacity, or proportional to length, etc.
    - The overall cost is a sum of all the costs from link to link
    - The goal of a routing algorithm is to identify the least-cost path (aka shortest path) from sources to destination
    - If all links have the same cost, the least-cost path is the path with the minimal number of links

- Routing Algorithm Classification
    - Centralized or global: all routers have complete topology, link cost info ("link state" algorithms)
    - Decentralized: iterative process of computation, exchange of info with neighbors ("distance vector" algorithms)
    - Static: routes change slowly over time
    - Dynamic: routes change more quickly (periodic updates or in response to link cost changes)

- Djikstra's Link-State Routing Algorithm
    - Centralized: network topology and link costs known to all nodes
        * Accomplished vie "link state broadcast"
        * All nodes have same info
    - Computes least cost paths from one node ("source") to all other nodes
        * Gives forwarding table for that node
    - Iterative: after k iterations, know least cost path to k destinations
    - Notation:
        * $c_{x,y}$: direct link cost from node x to y; $c_{x,y} = \infty$ if not direct neighbors
        * $D(v)$: current estimate of cost of least-cost-path from source to destination v
        * $p(v)$: predecessor node along path from source to v
        * N': set of nodes whose least-cost-path is definitively known
        * Ties can exist, and are broken arbitrarily
        * Construct least-cost-path tree by tracing predecessor nodes
    - Djikstra's Algorithm Complexity: n nodes
        * Each of n iterations: need to check all the nodes, w, not in N'
        * $n(n+1)/2$ comparisons: $O(n^2)$ complexity
        * More efficient implementations possible $O(n\log(n))$
    - Oscillations possible: when link costs depend on traffic volume

- Distance Vector Algorithm
  - Based on Bellman-Ford (BF) Equation [dynamic programming]
  - Let $d_x(y)$: cost of least-cost path from x to y, then:

$$d_x(y) = \min_v(c_{x,v} + d_v(y))$$

  - $D_x(y)$ is the estimate of the least cost from x to y
    * Then x maintains distance vector $D_x = [D_x(y) : y \in N]$
  - Iterative, asynchronous: each local iteration caused by:
    * Local link cost change
    * DV update message from neighbor
  - Distributed, self-stopping: each node notifies neighbors only when its DV changes
    * Neighbors then notify their neighbors, only if necessary
    * No notification received; no action taken