

The Network Layer: Control Plane

Michael Brodskiy

Professor: E. Bernal Mor

November 15, 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)
- Dijkstra’s Link-State Routing Algorithm
 - Centralized: network topology and link costs known to all nodes
 - * Accomplished via “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
 - Dijkstra’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

- Distance Vector: Link Cost Changes

- Node detects local link cost change
- Updates routing info, recalculates local DV
- If DV changes, notify neighbors
- “Good news travels fast” and “Bad news travels slow” (count to infinity problem)

- Comparison of LS and DV Algorithms

- Message Complexity
 - * LS: n nodes, E links, $O(nE)$ messages sent
 - * DV: exchange between neighbors only during convergence time
- Speed of Convergence
 - * LS: $O(n^2)$ algorithm requires $O(nE)$ messages
 - May have oscillations
 - * DV: convergence time varies
 - May be routing loops
 - Count-to-infinity problem
- Robustness: what happens if router malfunctions or is compromised
 - * LS:
 - Router can advertise incorrect link cost
 - Each router computes only its own table
 - * DV:

- DV router can advertise incorrect path cost
 - “I have a really low-cost path to everywhere” is called black-holing
 - Each router’s table used by others
 - Errors propagate through network
- Internet Approach to Scalable Routing
 - Aggregate routers into regions known as “Autonomous Systems” (AS, aka “domains”)
 - Intra-AS routing (aka “intra-domain”): routing among routers in the same AS (“network”)
 - * All routers in AS must run the same intra-domain protocol
 - * Routers in different AS can run different intra-domain routing
 - * Gateway router: at “edge” of its own AS, it has link(s) to router(s) in other ASs
- Intra-AS Routing: Routing Within an AS
 - Most common intra-AS routing protocols
 - * RIP: Routing Information Protocol
 - Classic DV: DVs exchanged every 30 seconds
 - No longer widely used
 - * EIGRP: Enhanced Interior Gateway Routing Protocol
 - Formerly Cisco-proprietary for decades (became open in 2013)
 - DV-based
 - * OSPF: Open Shortest Path First
 - Link-state routing
 - * IS-IS Protocol (Intermediate System to Intermediate System) is an ISO standard (not RFC standard) that is essentially same as OSPF
- OSPF Routing
 - “Open”: publicly available
 - Classic Link-State
 - * Each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS

- * Multiple link costs metrics possible: link capacity, delay
 - * Each router has full topology and uses Dijkstra's algorithm to compute forwarding table
- Security: all OSPF messages authenticated (to prevent malicious intrusion)
- Hierarchical OSPF
 - Two-level hierarchy: local area, backbone
 - * Link-state advertisements flooded only in area, or backbone
 - * Each node has detailed area topology; only knows direction to reach other destinations
- Internet Inter-AS Routing: BGP
 - BGP (Border Gateway Protocol): the de facto inter-domain routing protocol
 - * “Glue that holds the Internet control plane together”
 - * Uses an algorithm in the vein of distance vector routing
 - Allows network to advertise its existence, and the destinations it can reach, to rest of internet: “I am here, here is who I can reach, and how”
 - * Advertise prefixes (subnet or collection of subnets)
 - BGP provides each AS a means to:
 - * eBGP (external BGP): obtain subnet reachability information from neighboring ASes
 - * iBGP (internal BGP): propagate reachability information to all AS-internal routers
 - * Determine “good” routes to other networks based on reachability information and policy
- BGP Basics
 - BGP session: two BGP routers (“peers”) exchange BGP messages over semi-permanent TCP connection
 - BGP is a “path vector” protocol: advertises paths to different destination network prefixes

- Path Attributes and BGP Routes
 - BGP advertised route: prefix + attributes
 - * Prefix: destination advertised
 - * Two important attributes:
 1. AS-PATH: list of ASs through which prefix advertisement has passed
 2. NEXT-HOP: indicates specific internal-AS router to next-hop AS
 - Policy-based Routing
 - * Gateway receiving route advertisement uses import policy to accept/decline a path