Intradomain Routing

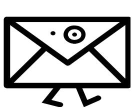
Routing Overview

- Routing
  - Routing vs. forwarding
  - Properties of routing protocols
- Internet routing architecture
  - Separation of intradomain and interdomain
  - Intra: metric-based and router level
  - Inter: policy-based at Autonomous System level

What is a Route?

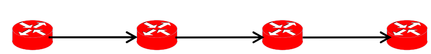
- A famous quotation from RFC 791

"A *name* indicates what we seek.
An *address* indicates where it is.
A *route* indicates how we get there."
-- Jon Postel

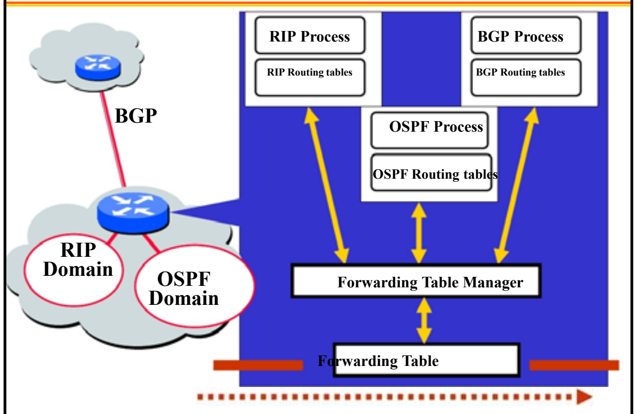


Forwarding vs. Routing

- **Forwarding:** data plane
  - Directing a data packet to an outgoing link
  - Individual router *using* a forwarding table
- **Routing:** control plane
  - Computing paths the packets will follow
  - Routers talking amongst themselves
  - Individual router *creating* a forwarding table



Where Does Forwarding Table Come From?



Why Does Routing Matter?

- End-to-end performance
  - Quality of the path affects user performance
  - Propagation delay, throughput, and packet loss
- Use of network resources
  - Balance of the traffic over the routers and links
  - Avoiding congestion by balancing load
- Transient disruptions during changes
  - Failures, maintenance, and load balancing
  - Limiting packet loss and delay during changes
- Realizing business objectives
  - Maximizing revenue or minimizing cost
  - Avoiding paths going through untrusted parties

Choosing a Routing Protocol

- Who is in charge of selecting the path?
  - The network or the end host?
- How complex are the path-selection goals?
  - Shortest-path vs. policy-based routing
- Are participants willing to cooperate?
  - Willing to share information?
  - Have a common goal in selecting paths?
- Is large-scale behavior a concern?
  - Stability of the network topology
  - State and message overhead
  - Disruptions during routing convergence

Many Kinds of Routing Protocols

- Link-state routing (Dijkstra)
  - Routers flood topology information
  - And compute (shortest) paths
- Distance-vector routing (Bellman-Ford)
  - Routers learn path costs from their neighbors
  - And select the neighbor along shortest path
- Policy-based path-vector routing
  - Routers learn full path from their neighbors
  - And select the most desirable path

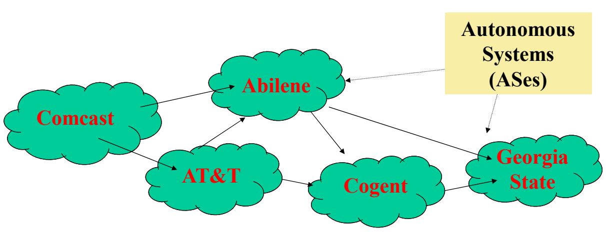
Many Kinds of Routing Protocols (Continued)

- Source routing
  - End host or edge router learn the topology
  - And select the end-to-end path
- Route servers
  - Set of servers learn topology and compute routes
  - And tell all the routers how to forward packets
- Ad hoc routing
  - Routers keep track of a small neighborhood
  - And forward packets in (hopefully) right direction

Two-Tiered Internet Routing Architecture

- Goal: distributed management of resources
  - Internetworking of multiple networks
  - Networks under separate administrative control
- Solution: two-tiered routing architecture
  - Intradomain: inside a region of control
    - Okay for routers to share topology information
    - Routers configured to achieve a common goal
  - Interdomain: between regions of control
    - Not okay to share complete information
    - Networks may have different/conflicting goals

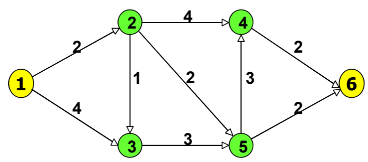
Internet Routing Overview



- Intradomain (*i.e.,* "intra-AS") routing
- Interdomain routing

Modelling Routing with ILP

Network modelling Techniques: the shortest path problem

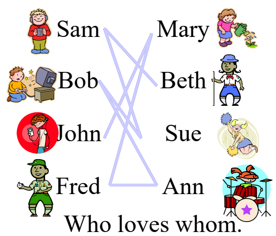


Consider a network G = (N, A) in which there is an origin node s and a destination node t.
standard notation: n = |N|, m = |A|

What is the shortest path from s to t?

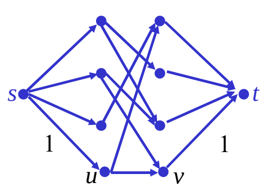
- ILP: Integer Linear Programming

A Matching Example



Who loves whom.
Who should be matched with whom
so as many as possible matched
and nobody matched twice?

Model the Matching with Flow

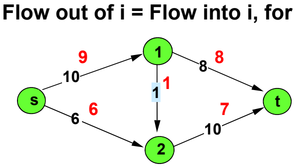


$c_{<s,u>} = 1$
•Total flow out of u = flow into $u \leq 1$
•Boy u matched to at most one girl.
$c_{<v,t>} = 1$
•Total flow into v = flow out of $v \leq 1$
•Girl v matched to at most one boy.

Ex2: the Maximum Flow Problem

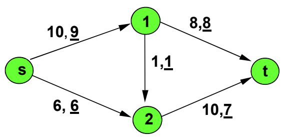
Directed Graph G = (N, A).
- **Source s**
- **Sink t**
- **Capacities u_{ij} on arc (i,j)**
- **Maximize the flow out of s, subject to**

Flow out of i = Flow into i, for i ≠ s or t.



A Network with Arc Capacities (and the maximum flow)

Representing the Max Flow as an LP

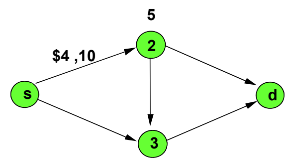


Flow out of i - Flow into i = 0 for i ≠ s or t.

$$\max \; v$$
$$\text{s.t} \quad x_{s1} + x_{s2} = v$$
$$-x_{s1} + x_{12} + x_{1t} = 0$$
$$-x_{s2} - x_{12} + x_{2t} = 0$$
$$-x_{1t} - x_{2t} = -v$$
$$0 \le x_{ij} \le u_{ij} \text{ for all } (i,j)$$

$$\max \; v$$
$$\text{s.t.} \quad \sum_j x_{sj} = v$$
$$\sum_j x_{ij} - \sum_j x_{ji} = 0 \quad \text{for each } i \ne s \text{ or } t$$
$$\text{s.t.} \quad -\sum_i x_{it} = -v$$
$$0 \le x_{ij} \le u_{ij} \text{ for all } (i,j)$$

Ex3: Min Cost Flows

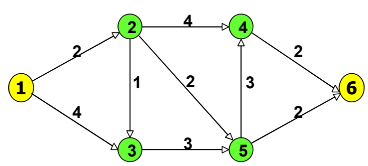


Flow out of s: b

Each arc has a linear cost and a capacity

$$\min \; \sum_{i,j} c_{ij} x_{ij}$$
$$\text{s.t} \quad \sum_j x_{ij} - \sum_j x_{ji} = 0 \text{ for each } i \text{ not } s, d$$
$$0 \le x_{ij} \le u_{ij} \text{ for all } (i,j)$$
$$\sum_j x_{sj} = b$$

Ex4: the shortest path problem



Consider a network G = (N, A) in which there is an origin node s and a destination node t.
standard notation: n = |N|, m = |A|

What is the shortest path from s to t?

Intradomain Topology and Routing

Routing Inside an AS

- Intra-AS **topology**
  - Nodes and edges
- Intradomain **routing** protocols
  - Distance Vector
    - Split-horizon/Poison-reverse
    - *Example:* RIP
  - Link State
    - *Example:* OSPF

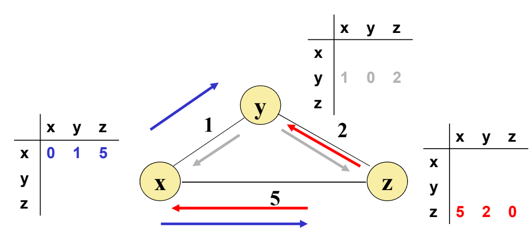
Key Questions

- **Where to place "nodes"?**
  - Typically in dense population centers
    - Close to other providers (easier interconnection)
    - Close to other customers (cheaper backhaul)
  - *Note:* A "node" may in fact be a group of routers, located in a single city. Called a "Point-of-Presence" (PoP)
- **Where to place "edges"?**
  - Often constrained by location of fiber

Problem: Routing

- **Routing:** the process by which nodes discover where to forward traffic so that it reaches a certain node
- **Within an AS:** there are two "styles"
  - Distance vector
  - Link State

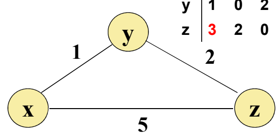
Distance-Vector Routing



- Routers send routing table copies to neighbors
- Routers compute costs to destination based on shortest available path
- Based on Bellman-Ford Algorithm
  - $d_x(y) = \min_v\{ c(x,v) + d_v(y) \}$
  - Solution to this equation *is* x's forwarding table

Good News Travels Quickly

	x	y	z
x	0	1	3
y	1	0	2
z	3	2	0



	x	y	z
x	0	1	3
y	1	0	2
z	3	2	0

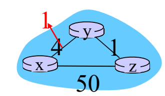
	x	y	z
x	0	1	3
y	1	0	2
z	3	2	0

- When costs decrease, network converges quickly

Distance Vector: link cost changes

Link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- if DV changes, notify neighbors



"good news travels fast"

At time t_0, y detects the link-cost change, updates its DV, and informs its neighbors.

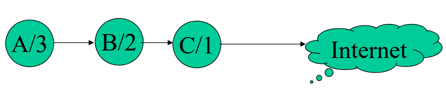
At time t_1, z receives the update from y and updates its table. It computes a new least cost to x and sends its neighbors its DV.

At time t_2, y receives z's update and updates its distance table. y's least costs do not change and hence y does *not* send any message to z.

Distance Vector: link cost changes

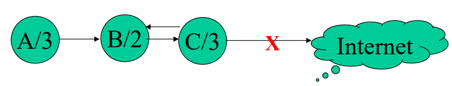
Link cost changes:

- good news travels fast
- bad news travels slow - "count to infinity" problem!



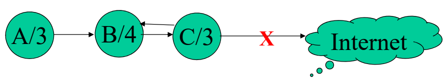
•Assume we use hop count as cost,
•A uses B and B uses C to reach Internet with costs 3, 2, 1, respectively

The Count to Infinity Problem



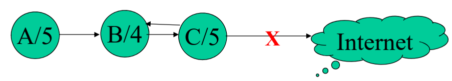
- •C's Internet link breaks
- • C erroneously switches to B, increases its cost to B's +1 = 3

The Count to Infinity Problem



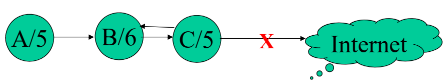
- B's path cost is now C's plus 1 = 4
- A hasn't realized what has happened yet

The Count to Infinity Problem



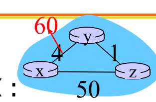
- •B's path cost is still 4
- •A's & C's cost are now B's + 1 = 5

The Count to Infinity Problem



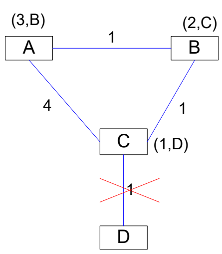
- •B's path cost is now C's + 1 = 6
- • Cycle repeats while "counting to infinity"
- • Packets caught between B & C loop

What are some solutions?



- Poisoned reverse:
  - If Z routes through Y to get to X :
    - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
  - will this completely solve count to infinity problem?
- Split horizon
- Triggered Updates

DV: Poisoned Reverse Fails Too



A	B	C
(3,B)	(2,C)	(1,D)
(3,B)	∞	(7,A)
∞	(8,C)	(7,A)
(9,B)	(8,C)	∞
(9,B)	∞	(13,A)
∞	(14,C)	(13,A)
…	…	…

Example: Routing Information Protocol

- Earliest IP routing protocol (1982 BSD)
  - Version 1: RFC 1058
  - Version 2: RFC 2453
- Features
  - Edges have unit cost
  - "Infinity" = 16
- Sending Updates
  - Router listens for updates on UDP port 520
  - Message can contain up to 25 table entries

RIP Updates

- Initial
  - When router first starts, asks for copy of table for every neighbor
  - Uses it to iteratively generate own table
- Periodic
  - Table refresh every 30 seconds
- Triggered
  - When every entry changes, send copy of entry to neighbors
    - Except for one causing update (*split horizon* rule)
  - Neighbors use to update their tables

RIP: Staleness and Oscillation Control

- Small value for Infinity
  - Count to infinity doesn't take very long
- Route Timer
  - Every route has timeout limit of 180 seconds
    - Reached when haven't received update from next hop for 6 periods
  - If not updated, set to infinity
  - *Soft-state*
- Behavior
  - When router or link fails, can take minutes to stabilize

Link-State Routing

- ***Idea:*** distribute a network map
- Each node performs shortest path (SPF) computation between itself and all other nodes
- Initialization step
  - Add costs of immediate neighbors, *D(v)*, else infinite
  - Flood costs *c(u,v)* to neighbors, *N*
- For some *D(w)* that is not in *N*
  - *D(v) = min(c(u,w) + D(w), D(v))*

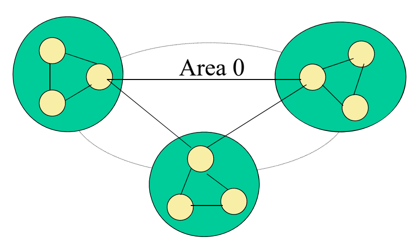
Link-State vs. Distance-Vector

- **Convergence**
  - DV has count-to-infinity
  - DV often converges slowly (minutes)
  - Odd timing dependencies in DV
- **Robustness**
  - Route calculations a bit more robust under link-state.
  - DV algorithms can advertise incorrect least-cost paths
- **Bandwidth Consumption** for Messages
- **Computation**
- **Security**

OSPF: Salient Features

- Dijkstra, plus some additional features
- Equal-cost multipath
- Support for hierarchy: Inter-Area Routing

Example: Open Shortest Paths First (OSPF)

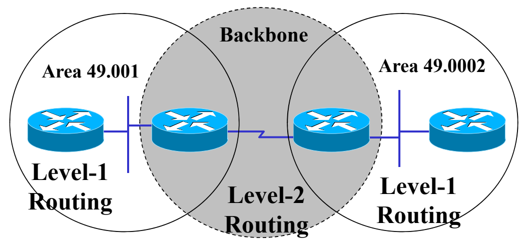


- Key Feature: hierarchy
- Network's routers divided into *areas*
- *Backbone area* is area 0
- Area 0 routers perform SPF computation
  - All inter-area traffic travles through Area 0 routers ("border routers")

Example: IS-IS

- Originally: ISO Connectionless Network Protocol (CLNP) .
  - CLNP: ISO equivalent to IP for datagram delivery services
  - ISO 10589 or RFC 1142
- Later: Integrated or Dual IS-IS (RFC 1195)
  - IS-IS adapted for IP
  - Doesn't use IP to carry routing messages
- OSPF more widely used in enterprise, IS-IS in large service providers

Hierarchical Routing in IS-IS



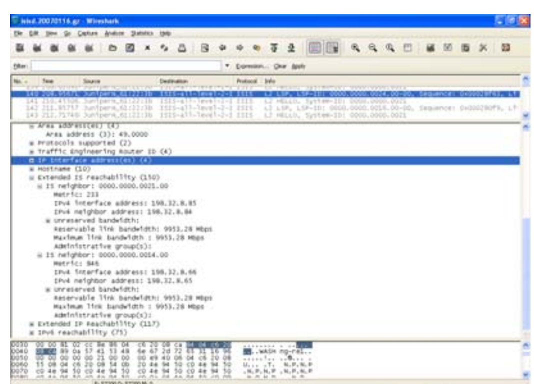
- Like OSPF, 2-level routing hierarchy
  - Within an area: level-1
  - Between areas: level-2
  - Level 1-2 Routers: Level-2 routers may also participate in L1 routing

Level-1 vs. Level-2 Routing

Level 1 routing
- Routing *within* an area
- Level 1 routers track links, routers, and end systems within L1 area
- L1 routers do not know the identity of destinations outside their area.
- A L 1 router forwards all traffic for destinations outside its area to the nearest L2 router within its area.

Level 2 routing
- Routing *between* areas
- Level 2 routers know the level 2 topology and know which addresses are reachable via each level 2 router.
- Level 2 routers track the location of each level 1 area.
- Level 2 routers are not concerned with the topology within any level 1 area (for example, the details internal to each level 1 area).
- Level 2 routers can identify when a level 2 router is also a level 1 router within the same area.
- Only a level 2 router can exchange packets with external routers located outside its routing domain.

ISIS on the Wire…



IS-IS vs. OSPF

- Cisco ships OSPF in 1991
- Cisco ships dual IS-IS in 1992
- *Circa 1995:* ISPs need to run IGPs, IS-IS is recommended due to the recent rewrite
- IS-IS became very popular in late 1990s
  - Deployed in most large ISPs (also Abilene)
  - Some ISPs (*e.g.,* AOL backbone) even switched

Measuring Internet Routing

Motivations for Measuring the Routing System

- Characterizing the Internet
  - Internet path properties
  - Demands on Internet routers
  - Routing convergence

- Improving Internet health
  - Protocol design problems
  - Protocol implementation problems
  - Configuration errors or attacks

- Operating a network
  - Detecting and diagnosing routing problems
  - Traffic shifts, routing attacks, flaky equipment, …

Techniques for Measuring Internet Routing

- Active probing
  - Inject probes along path through the data plane
  - E.g., using traceroute
- Passive route monitoring
  - Capture control-plane messages between routers
  - E.g., using tcpdump or a software router
  - E.g., dumping the routing table on a router
- Injecting network events
  - Cause failure/recovery at planned time and place
  - E.g., BGP route beacon, or planned maintenance

Internet Routing is Hard to Measure

- Nobody knows the Internet topology
  - No central registry of the AS-level graph
  - Little public information about intra-AS topologies
- Deploying monitoring infrastructure is hard
  - Forwarding: active probes of end-to-end paths
  - Routing: passive monitoring of routing messages
- Many measurement challenges
  - Network conditions vary by location
  - Network conditions change over time
  - One-way measurements are hard to collect
  - Controlled experiments are hard to do

Conclusions

- Internet routing architecture
  - Two-tiered system
  - Intradomain is metric-based, with common goal
  - Interdomain is policy-based, reconciling different goals across ASes
- Behavior of complete systems is mysterious
  - Challenging to measure
  - Challenging to characterize, and diagnose

Interdomain Routing: Border Gateway Protocol

- ASes exchange info about who they can reach
  - IP prefix: block of destination IP addresses
  - AS path: sequence of ASes along the path
- Policies configured by the AS's operator
  - Path selection: which of the paths to use?
  - Path export: which neighbors to tell?