Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let \( d_x(y) := \text{cost of least-cost path from } x \text{ to } y \) then
\[
d_x(y) = \min_v \{ c(x,v) + d_v(y) \}
\]

B-F equation says:
\[
d_x(z) = \min \{ c(u,v) + d_v(z), c(u,x) + d_x(z), c(u,w) + d_w(z) \} = \min \{ 2 + 5, 1 + 3, 5 + 3 \} = 4
\]

node achieving minimum is next hop in shortest path, used in forwarding table

Bellman-Ford example

clearly, \( d_z(z) = 5, d_z(x) = 3, d_z(w) = 3 \)

Distance vector algorithm

key idea:
- from time-to-time, each node sends its own distance vector estimate to neighbors
- when \( x \) receives new DV estimate from neighbor, it updates its own DV using B-F equation:
  \[
  D_x(y) \leftarrow \min_v \{ c(x,v) + D_v(y) \} \text{ for each } y \in N
  \]
- under minor, natural conditions, the estimate \( D_x(y) \) converge to the actual least cost \( d_x(y) \)

Distance vector algorithm

iterative, asynchronous:
- each local iteration caused by:
  - local link cost change
  - DV update message from neighbor

distributed:
- each node notifies neighbors only when its DV changes
  - neighbors then notify their neighbors if necessary

Distance vector algorithm

each node:
- wait for (change in local link cost or msg from neighbor)
- recomputes estimates
- if DV to any dest has changed, notify neighbors
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

1. y detects link cost change, updates its DV, informs its neighbors.
2. z receives update from y, updates its table, computes new least cost to x, sends its neighbors its DV.
3. y detects z's update, updates its distance table. y's least costs do not change, so y does not send a message to z.

Comparison of LS and DV algorithms

**message complexity**
- LS: w n nodes, E links, O(nE) msgs sent
- DV: exchange between neighbors only
  - convergence time varies

**speed of convergence**
- LS: O(n+E) algorithm requires O(n+E) msgs
- may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

**robustness:** what happens if router malfunctions?
- LS:
  - node can advertise incorrect link cost
  - each node computes only its own table
- DV:
  - DV node can advertise incorrect path cost
  - each node's table used by others
  - error propagate thru network

Chapter 4: outline

4.1 introduction
4.2 virtual circuit and datagram networks
4.3 what's inside a router
4.4 IP: Internet Protocol
  - datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
4.5 routing algorithms
  - link state
  - distance vector
  - hierarchical routing
4.6 routing in the Internet
  - RIP
  - OSPF
  - BGP
4.7 broadcast and multicast routing
Hierarchical routing

- our routing study thus far - idealization
  - all routers identical
  - network "flat"
- ... not true in practice

**scale:** with 600 million destinations:
- can't store all dest's in routing tables!
- routing table exchange would swamp links!

**administrative autonomy**
- internet = network of networks
- each network admin may want to control routing in its own network

---

Interconnected ASes

- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
  - inter-AS & intra-AS sets entries for external dests

---

Example: setting forwarding table in router 1d

- suppose AS1 learns (via inter-AS protocol) that subnet \( x \) reachable via AS3 (gateway 1c), but not via AS2
  - inter-AS protocol propagates reachability info to all internal routers
  - router 1d determines from intra-AS routing info that its interface \( I \) is on the least cost path to 1c
  - installs forwarding table entry \((x,I)\)

---

Example: choosing among multiple ASes

- now suppose AS1 learns from inter-AS protocol that subnet \( x \) is reachable from AS3 and from AS2
  - to configure forwarding table, router 1d must determine which gateway it should forward packets towards for dest \( x \)
  - this is also job of inter-AS routing protocol!
Example: choosing among multiple ASes

- now suppose AS1 learns from inter-AS protocol that subnet x is reachable from AS3 and from AS2.
- to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x
- this is also job of inter-AS routing protocol!
- hot potato routing: send packet towards closest of two routers.

4.6 routing in the Internet
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Intra-AS Routing

- also known as interior gateway protocols (IGP)
- most common intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

RIP (Routing Information Protocol)

- included in BSD-UNIX distribution in 1982
- distance vector algorithm
  - distance metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination subnets (in IP addressing sense)

Routing table in router D

<table>
<thead>
<tr>
<th>Destination subnet</th>
<th>next router</th>
<th># hops to dest</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td>x</td>
<td>--</td>
<td>1</td>
</tr>
</tbody>
</table>

RIP: example

A-to-D advertisement

<table>
<thead>
<tr>
<th>Dest</th>
<th>next router</th>
<th># hops to dest</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>x</td>
<td>--</td>
<td>1</td>
</tr>
</tbody>
</table>
**RIP: link failure, recovery**

if no advertisement heard after 180 sec -->
neighbor/link declared dead
- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly (?) propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

**RIP table processing**

- RIP routing tables managed by *application-level* process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated

**OSPF (Open Shortest Path First)**

- "open": publicly available
- uses link state algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation using Dijkstra’s algorithm
- OSPF advertisement carries one entry per neighbor
- advertisements flooded to *entire* AS
  - carried in OSPF messages directly over IP (rather than TCP or UDP)
- IS-IS routing protocol: nearly identical to OSPF

**OSPF “advanced” features (not in RIP)**

- *security*: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)
  - for each link, multiple cost metrics for different TOS (e.g., satellite link cost set “low” for best effort ToS; high for real time ToS)
  - integrated uni- and multicast support:
    - Multicast OSPF (MOSPF) uses same topology data base as OSPF
  - hierarchical OSPF in large domains.

**Hierarchical OSPF**

- two-level hierarchy: local area, backbone.
  - link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- area border routers: "summarize" distances to nets in own area, advertise to other Area Border routers.
- backbone routers: run OSPF routing limited to backbone.
- boundary routers: connect to other AS’s.
**Internet inter-AS routing: BGP**

- **BGP (Border Gateway Protocol):** the de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASs.
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and policy.
- allows subnet to advertise its existence to rest of Internet: "I am here"

---

**BGP basics**

- **BGP session:** two BGP routers ("peers") exchange BGP messages:
  - advertising paths to different destination network prefixes ("path vector" protocol)
  - exchanged over semi-permanent TCP connections
- when AS3 advertises a prefix to AS1:
  - AS3 promises it will forward datagrams towards that prefix
  - AS3 can aggregate prefixes in its advertisement

---

**BGP basics: distributing path information**

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use iBGP to distribute new prefix info to all routers in AS1
  - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- when router learns of new prefix, it creates entry for prefix in its forwarding table.

---

**Path attributes and BGP routes**

- advertised prefix includes BGP attributes
  - prefix + attributes = "route"
- two important attributes:
  - AS-PATH: contains ASs through which prefix advertisement has passed: e.g., AS 67, AS 17
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS. (may be multiple links from current AS to next-hop-AS)
- gateway router receiving route advertisement uses import policy to accept/decline
  - e.g., never route through AS x
  - policy-based routing

---

**BGP route selection**

- router may learn about more than 1 route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria

---

**BGP messages**

- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection
Putting it Altogether: How Does an Entry Get Into a Router’s Forwarding Table?

- Answer is complicated!
- Ties together hierarchical routing (Section 4.5.3) with BGP (4.6.3) and OSPF (4.6.2).
- Provides nice overview of BGP!

How does entry get in forwarding table?

**High-level overview**
1. Router becomes aware of prefix
2. Router determines output port for prefix
3. Router enters prefix-port in forwarding table

Router becomes aware of prefix

- BGP message contains "routes"
- "route" is a prefix and attributes: AS-PATH, NEXT-HOP, ...
- Example: route:
  - Prefix: 138.16.64/22; AS-PATH: AS3 AS131; NEXT-HOP: 201.44.13.125

Router may receive multiple routes

- Router may receive multiple routes for same prefix
- Has to select one route

Select best BGP route to prefix

- Router selects route based on shortest AS-PATH

- Example:
  - AS2 AS17 to 138.16.64/22
  - AS3 AS131 AS201 to 138.16.64/22

- What if there is a tie? We’ll come back to that!
Find best intra-route to BGP route

- Use selected route’s NEXT-HOP attribute
  - Route’s NEXT-HOP attribute is the IP address of the router interface that begins the AS PATH.
- Example:
  - AS-PATH: AS2 AS17; NEXT-HOP: 111.99.86.55
  - Router uses OSPF to find shortest path from 1c to 111.99.86.55

Router identifies port for route

- Identifies port along the OSPF shortest path
- Adds prefix-port entry to its forwarding table:
  - (138.16.64/22, port 4)

Hot Potato Routing

- Suppose there are two or more best inter-routes.
- Then choose route with the closest NEXT-HOP
  - Use OSPF to determine which gateway is closest
- Q: From 1c, choose AS3 AS131 or AS2 AS17?
- A: route AS3 AS131 since it is closer

How does entry get in forwarding table?

Summary

1. Router becomes aware of prefix
   - via BGP route advertisements from other routers
2. Determine router output port for prefix
   - Use BGP route selection to find best inter-AS route
   - Use OSPF to find best intra-AS route leading to best inter-AS route
   - Router identifies router port for that best route
3. Enter prefix-port entry in forwarding table

BGP routing policy

- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - so X will not advertise a route to C

BGP routing policy (2)

- A advertises path AW to B
- B advertises path BAW to X
- Should B advertise path BAW to C?
  - No way! B gets no "revenue" for routing CBAW since neither W nor C are B’s customers
  - B wants to force C to route to W via A
  - B wants to route only to/from its customers!
**Why different Intra-, Inter-AS routing?**

**Policy:**
- Inter-AS: admins want control over how traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed.

**Scale:**
- Hierarchical routing saves table size, reduced update traffic.

**Performance:**
- Intra-AS: can focus on performance.
- Inter-AS: policy may dominate over performance.

**Chapter 4: outline**

4.1 Introduction
4.2 Virtual circuit and datagram networks
4.3 What’s inside a router
4.4 IP: Internet Protocol
   - Datagram format
   - IPv4 addressing
   - ICMP
   - IPv6
4.5 Routing algorithms
   - Link state
   - Distance vector
   - Hierarchical routing
4.6 Routing in the Internet
   - RIP
   - OSPF
   - BGP
4.7 Broadcast and multicast routing

**Broadcast routing**

- Deliver packets from source to all other nodes.
- Source duplication is inefficient.

**In-network duplication**

- Flooding: when node receives broadcast packet, sends copy to all neighbors.
  - Problems: cycles & broadcast storm.
- Controlled flooding: node only broadcasts pkt if it hasn’t broadcast same packet before.
  - Node keeps track of packet ids already broadcasted.
  - Or reverse path forwarding (RPF): only forward packet if it arrived on shortest path between node and source.

- Spanning tree:
  - No redundant packets received by any node.

**Spanning tree**

- First construct a spanning tree.
- Nodes then forward/make copies only along spanning tree.

(a) Broadcast initiated at A

(b) Broadcast initiated at D

(a) Stepwise construction of spanning tree (center: E)

(b) Constructed spanning tree
Multicast routing: problem statement

**goal:** find a tree (or trees) connecting routers having local mcast group members
- **tree:** not all paths between routers used
- **shared-tree:** same tree used by all group members
- **source-based:** different tree from each sender to rcvs

Approaches for building mcast trees

approaches:
- **source-based tree:** one tree per source
  - shortest path trees
  - reverse path forwarding
- **group-shared tree:** group uses one tree
  - minimal spanning (Steiner)
  - center-based trees

…we first look at basic approaches, then specific protocols adopting these approaches

Shortest path tree

- mcast forwarding tree: tree of shortest path routes from source to all receivers
  - Dijkstra’s algorithm

Reverse path forwarding

- rely on router’s knowledge of unicast shortest path from it to sender
- each router has simple forwarding behavior:

Reverse path forwarding: example

- result is a source-specific reverse SPT
  - may be a bad choice with asymmetric links

Reverse path forwarding: pruning
**Shared-tree: steiner tree**

- **steiner tree**: minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/leave

**Center-based trees**

- single delivery tree shared by all
- one router identified as “center” of tree
- to join:
  - edge router sends unicast join-msg addressed to center router
  - join-msg “processed” by intermediate routers and forwarded towards center
  - join-msg either hits existing tree branch for this center, or arrives at center
  - path taken by join-msg becomes new branch of tree for this router

**Center-based trees: example**

Suppose R6 chosen as center: