ECE/COMPSCI 356 Computer Network Architecture

Lecture 12: Dynamic Routing: Routing Information Protocol

Neil Gong

neil.gong@duke.edu

Slides credit: Xiaowei Yang, PD

Today

- Dynamic Routing
  - Routing Information Protocol

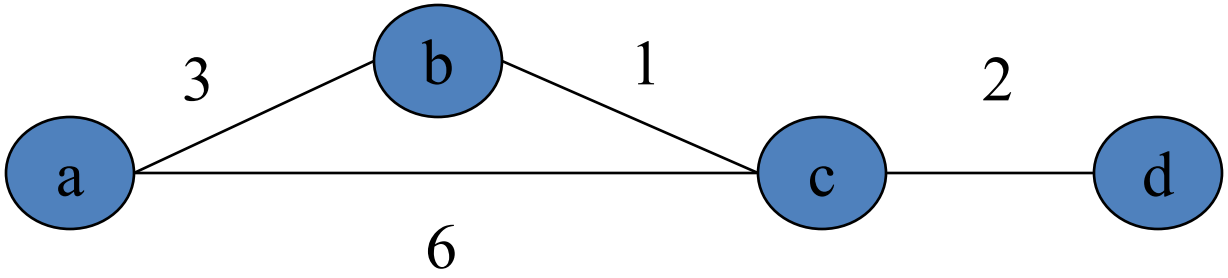
Forwarding vs. Routing

- Forwarding
  - Forward packets given a forwarding table

- Routing
  - Obtain forwarding table
  - Routing protocol maintains its own table
    - Called routing table
  - Forwarding table is built based on routing table

Routing

- Network as a Graph



- The basic problem of routing is to find the lowest-cost path between any two nodes
  - Where the cost of a path equals the sum of the costs of all the edges that make up the path

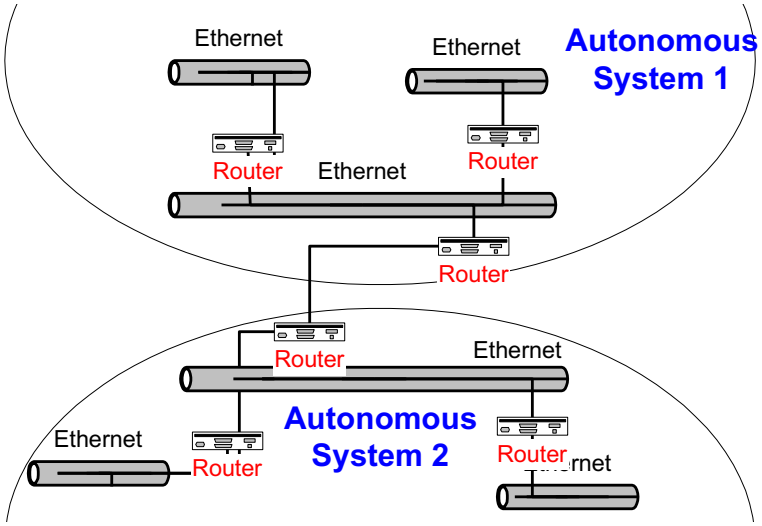
Static versus dynamic routing

- Static routing (Lab 2)
  - For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.
  - Such a static approach has several shortcomings
    - It does not deal with node or link failures
    - It does not consider the addition of new nodes or links
    - It implies that edge costs cannot change
- Dynamic routing
  - A distributed and dynamic protocol

Diverse routing choices

- The Internet is a network of networks
  - each network admin may want to control routing in its own network

Autonomous Systems



- An **autonomous system (AS)** is a region of the Internet that is administered by a single entity
  - Also called domain

- Examples of AS are:
  - Duke's campus network
  - at&t's backbone network
  - Regional Internet Service Provider (NC regional)

Intradomain vs. interdomain routing

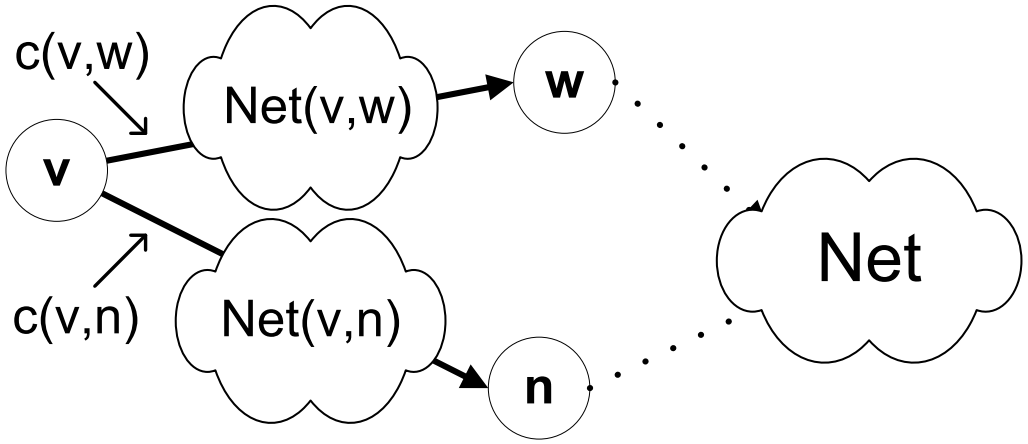
- intradomain routing
  - Routers in the same AS run the same routing protocol
  - routers in different AS can run different intradomain routing protocols
  - Examples: RIP, OSPF, IGRP, and IS-IS

- interdomain routing
  - Routing across AS
  - BGP is the only interdomain routing protocol

Routing Information Protocol (RIP)

- Representing an AS as a graph
- Use distance vector algorithm

AS/Network as a graph



- Nodes
  - Routers, identified by IP addresses
  - E.g., v, w, n
- Edges
  - A directed link or a multiple-access link (Ethernet) connecting two routers
  - E.g., Net(v,w), Net(v,n)
- Costs
  - Cost of transmitting a packet between two routers.
  - Delay, bandwidth, queueing time, etc.
  - Could be bidirectional

Distance vector algorithm

- A distributed algorithm
  - Each node has a partial view
    - Neighbors
    - Link costs to neighbors
- Each node maintains a distance vector
  - Distance to every other node
    - Distance is the sum of the costs of the shortest path
  - And the NextHop to reach every other node along the shortest path
- Distance vectors are iteratively computed
  1. A router sends its distance vector to its neighbors
  2. A router updates its distance vector based on its neighbors' distance vectors
  3. The process repeats until all routers' distance vectors do not change (this condition is called convergence).

A router updates its distance vector using Bellman-Ford equation

Bellman-Ford Equation

Define

$d_x(y)$:= cost of the least-cost path from x to y

Then

- $d_x(y) = \min_v\{c(x,v) + d_v(y)\}$, where min is taken over all neighbors of node x

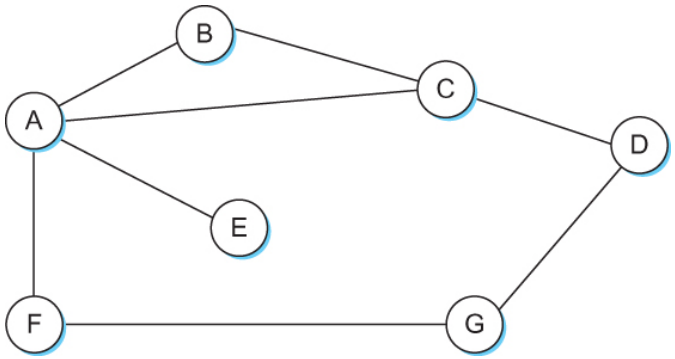
Distance vector algorithm: initialization

- Let $d_x(y)$ be the least cost from x to y

- Initialization:
  - Each node x knows the cost to each neighbor: c(x,v).
  - For each neighbor v of x, $d_x(v) = c(x,v)$
  - $d_x(x) = 0$
  - $d_x(y)$ to other nodes are initialized as infinity

- Each node x maintains a distance vector:
  - An entry is a tuple: (u, $d_x(u)$, NextHop)

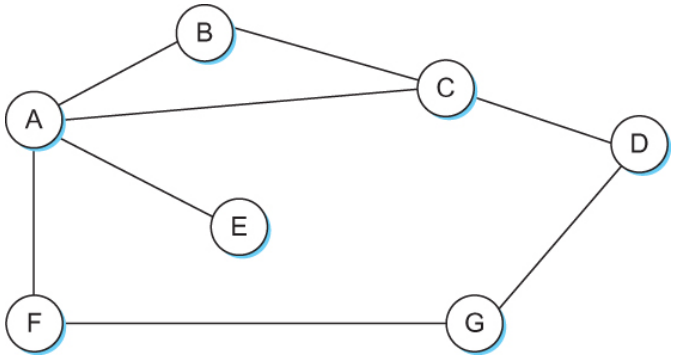
Distance Vector



Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Initial distances stored at each node (global view)

Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

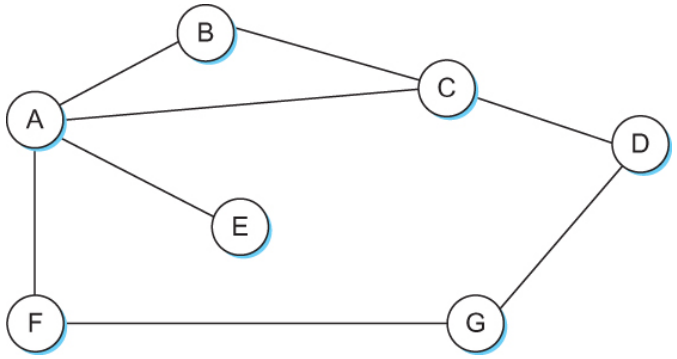
local view at node A (called routing table)

Distance vector algorithm: updates

- Each node x sends its distance vector to its neighbors
  - periodically or triggered by a change in its distance vector

- When a node x receives a distance vector from a neighbor v, it updates its own distance vector using the Bellman-Ford Equation
  - Considering entry (u, $d_x(u)$, NextHop)
  - If $c(x,v) + d_v(u) < d_x(u)$ then
    - $d_x(u) = c(x,v) + d_v(u)$
    - NextHop=v

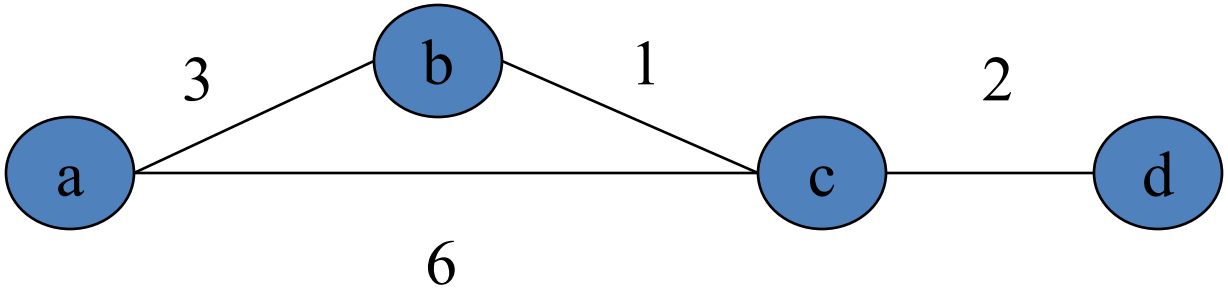
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final routing table at node A

Distance vector algorithm: an example



- t = 0
- a = ((a, 0, a), (b, 3, b), (c, 6, c))
- b = ((a, 3, a), (b, 0, b), (c, 1, c))
- c = ((a, 6, a), (b, 1, b), (c, 0, c) (d, 2, d))
- d = ((c, 2, c), (d, 0, d))

- t = 1
- a = ((a, 0, a), (b, 3, b), (c, 4, b), (d, 8, c))
- b = ((a, 3, a), (b, 0, b), (c, 1, c), (d, 3, c))
- c = ((a, 4, b), (b, 1, b), (c, 0, c), (d, 2, d))
- d = ((a, 8, c), (b, 3, c), (c, 2, c), (d, 0, d))

- t = 2
- a = ((a, 0, a), (b, 3, b), (c, 4, b), (d, 6, b))
- b = ((a, 3, a), (b, 0, b), (c, 1, c), (d, 3, c))
- c = ((a, 4, b), (b, 1, b), (c, 0, c), (d, 2, d))
- d = ((a, 6, c), (b, 3, c), (c, 2, c), (d, 0, d))

Protocols versus algorithms

- Routing protocols establish forwarding tables at routers

- A routing protocol implements a distributed algorithm
  - What messages are sent
  - When are they sent
  - How are they handled

- At the heart of any routing protocol is a distributed algorithm that determines the path from a source to a destination

What distributed algorithms common routing protocols use

Routing protocol	Distributed algorithm
Routing information protocol (RIP)	Distance vector
Interior Gateway routing protocol (IGRP, cisco proprietary)	Distance vector
Open shortest path first (OSPF)	Link state
Intermediate System-to-Intermediate System (IS-IS)	Link state
Border gateway protocol (BGP)	Path vector

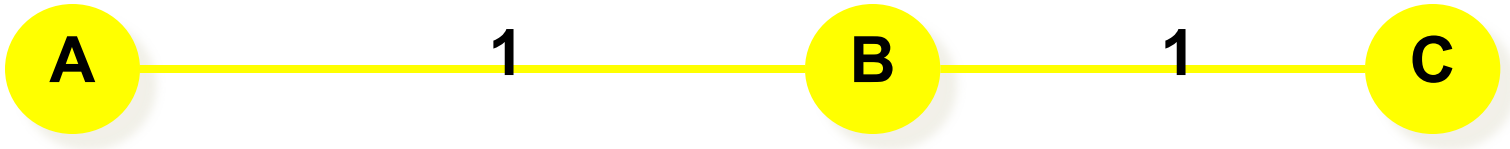
RIP - Routing Information Protocol

- A simple intradomain protocol
- Straightforward implementation of distance vector algorithm
- RIP uses 1 as link cost for every link
- Each router maintains a routing table
  - Entries are the distance vector entries: (node, distance to node, NextHop)
- **Periodic Updates:**
  - Advertise its distance vector/routing table to its neighbors every 30 seconds
- **Triggered Updates:**
  - If a node's distance vector changes, e.g., link/neighbor failure, a router immediately sends out its routing table without waiting for the end of the update period

RIP Problems

- RIP takes a long time to stabilize
  - Even for a small network, it takes several minutes until the routing tables have settled after a change
- Count-to-Infinity problem

The Count-to-Infinity Problem



A's Routing Table

to	via (next hop)	cost
C	B	2

B's Routing Table

to	via (next hop)	cost
C	C	1

now link B-C goes down

C	B	2

C	-	∞

→ ←

C	-	∞

C	A	3

→ ←

C	B	4

C	-	∞

→ ←

Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a “next-hop-view”
- How can the Count-to-Infinity problem be solved?

Solutions to Count-to-Infinity

- Solution 1: Split Horizon
  - Never advertise the cost to a neighbor if this neighbor is the next hop on the current path
    - Example: A would not send the first routing update to B, since B is the next hop on A's current route to C
    - Used by RIP
  - Stronger variant: split horizon with poison reverse
    - Send to the next hop neighbor an invalid route (C, B, ∞)
  - Only solve the problem if routing loops involve only two nodes

- Solution 2: Has a small infinity so that routing messages will not bounce forever
  - RIP uses 16 as infinity
  - Drawbacks
    - Maximum hop count is 15
    - Only supporting small networks

Summary

- Dynamic Routing
  - Distant vector algorithm
  - RIP