# CSE 461: Distance Vector Routing

## Next Topic

- Focus
  - How do we calculate routes for packets?
  - Routing is a network layer function
- Routing Algorithms
  - Distance Vector routing (RIP)

| |
|---|
| Application |
| Presentation |
| Session |
| Transport |
| Network |
| Data Link |
| Physical |

## IP Addresses and IP Datagram Forwarding

- How the source gets the packet to the destination:
  - if source is on same network (LAN) as destination, source sends packet directly to destination host
  - else source sends data to a router on the same network as the source
  - router will forward packet to a router on the next network over
  - and so on…
  - until packet arrives at router on same network as destination; then, router sends packet directly to destination host
- Requirements
  - every host needs to know IP address of the router on its LAN
  - every router needs a routing table to tell it which neighboring network to forward a given packet on

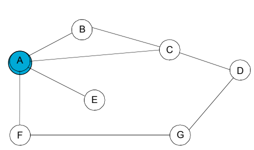
## Forwarding and Routing

- Forwarding is the process that each router goes through for every packet to send it on its way
  - Involves local decisions
- Routing is the process that all routers go through to calculate the routing tables
  - Involves global decisions

## What's in a Routing Table?

- The routing table at A, for example, lists at a minimum the next hops for the different destinations

| Dest | Next Hop |
|---|---|
| B | B |
| C | C |
| D | C |
| E | E |
| F | E |
| G | F |




## Kinds of Routing Schemes

- Many routing schemes have been proposed/explored!
- Distributed or centralized
- Hop-by-hop or source-based
- Deterministic or stochastic
- Single or multi-path
- Static or dynamic route selection
- Internet is to the left ☺

## Routing Questions/Challenges

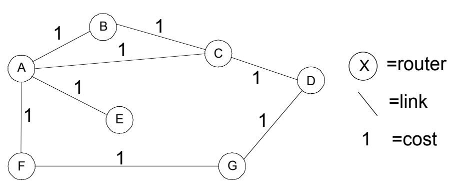
- How to choose best path?  What is best path?
- How to scale to millions of users?
- How to adapt to failures or changes?
  - Node and link failures, plus message loss
  - We will use distributed algorithms

## Some Pitfalls

- Using global knowledge is challenging
  - Hard to collect
  - Can be out-of-date
  - Needs to summarize in a locally-relevant way
- Inconsistencies in local /global knowledge can cause:
  - Loops (black holes)
  - Oscillations, esp. when adapting to load

## Network as a Graph

- Routing is essentially a problem in graph theory



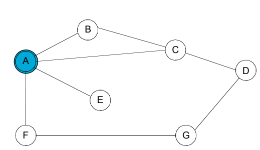

## Distance Vector Routing

- Assume:
  - Each router knows only address/cost of neighbors
- Goal:
  - Calculate routing table of next hop information for each destination at each router
- Idea:
  - Tell neighbors about learned distances to all destinations

## DV Algorithm

- Each router maintains a vector of costs to all destinations as well as routing table
  - Initialize neighbors with known cost, others with infinity
- Periodically send copy of distance vector to neighbors
  - On reception of a vector, if neighbors path to a destination plus neighbor cost is better, then switch to better path
    - update cost in vector and next hop in routing table
- Assuming no changes, will converge to shortest paths
  - But what happens if there are changes?

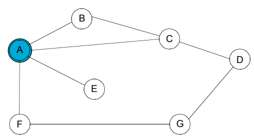
## DV Example – Initial Table at A




| Dest | Cost | Next |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | ∞ | - |
| E | 1 | E |
| F | 1 | F |
| G | ∞ | - |

## DV Example – Final Table at A

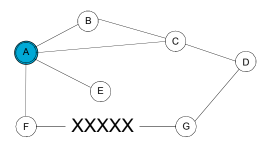
- Reached in a single iteration … simple example




| Dest | Cost | Next |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | 2 | C |
| E | 1 | E |
| F | 1 | F |
| G | 2 | F |

## What if there are changes?

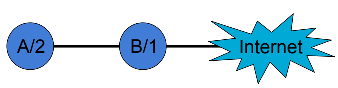
- One scenario: Suppose link between F and G fails
  1. F notices failure, sets its cost to G to infinity and tells A
  2. A sets its cost to G to infinity too, since it learned it from F
  3. A learns route from C with cost 2 and adopts it




| Dest | Cost | Next |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | 2 | C |
| E | 1 | E |
| F | 1 | F |
| G | 3 | C |

## Count To Infinity Problem

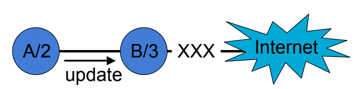
- Simple example
  - Costs in nodes are to reach Internet




- Now link between B and Internet fails …

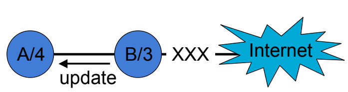
## Count To Infinity Problem

- B hears of a route to the Internet via A with cost 2
- So B switches to the "better" (but wrong!) route



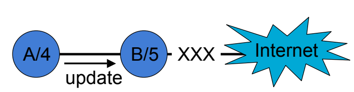

## Count To Infinity Problem

- A hears from B and increases its cost




## Count To Infinity Problem

- B hears from A and (surprise) increases its cost
- Cycle continues and we "count to infinity"




- Packets caught in the crossfire loop between A and B

## Split Horizon

- Solves trivial count-to-infinity problem
- Router never advertises the cost of a destination back to its next hop – that's where it learned it from!
- Poison reverse: go even further – advertise back infinity
- However, DV protocols still subject to the same problem with more complicated topologies
  - Many enhancements suggested

## Routing Information Protocol (RIP)

- DV protocol with hop count as metric
  - Infinity value is 16 hops; limits network size
  - Includes split horizon with poison reverse
- Routers send vectors every 30 seconds
  - With triggered updates for link failures
  - Time-out in 180 seconds to detect failures
- RIPv1 specified in RFC1058
  - www.ietf.org/rfc/rfc1058.txt
- RIPv2 (adds authentication etc.) in RFC1388
  - www.ietf.org/rfc/rfc1388.txt

## RIP is an "Interior Gateway Protocol"

- Suitable for small- to medium-sized networks
  - such as within a campus, business, or ISP
- Unsuitable for Internet-scale routing
  - hop count metric poor for heterogeneous links
  - 16-hop limit places max diameter on network
- Later, we'll talk about "Exterior Gateway Protocols"
  - used between organizations to route across Internet

## Key Concepts

- Routing is a global process, forwarding is local one
- The Distance Vector algorithm and RIP
  - Simple and distributed exchange of shortest paths.
  - Weak at adapting to changes (loops, count to infinity)