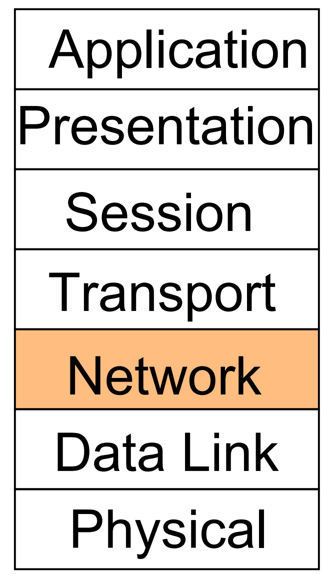
CSE/EE 461
Distance Vector Routing

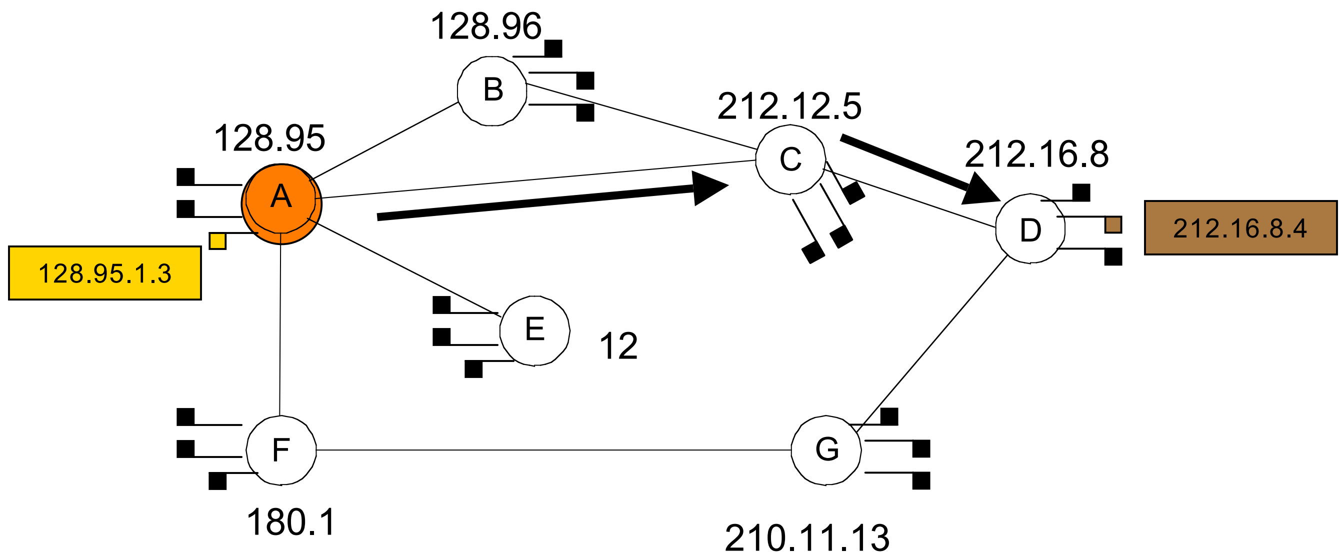
This Time

- Focus
  - How do we calculate routes for packets
  - Routing is a network layer function
    - Each packet is routed on the basis of its destination
      - In contrast to MST where we are creating an extended LAN
        - » Broadcast, but to all the right places
- Routing Algorithms
  - Distance Vector routing (RIP)



Routing and Forwarding

source	128.95.1.3
dest	212.16.8.4



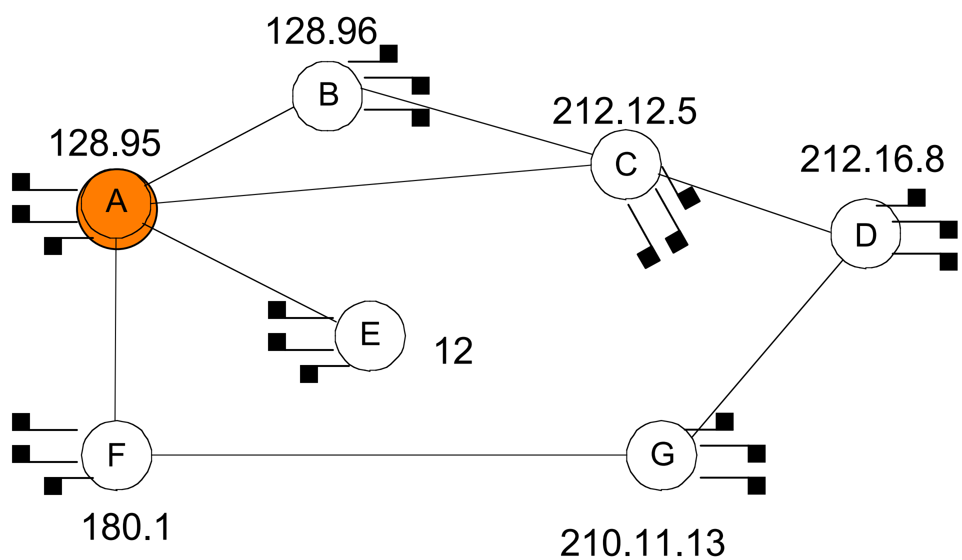
Routing and Forwarding

- Routing is the global process that all routers go through to calculate routes
  - Involves global decisions
  - "For this link, which networks can I get to?"
- Forwarding is local the process that each router goes through for every packet to send it on its way
  - Involves local decisions
  - "For this packet, which link should I use to send it out?"
    - Compare to MST

Routing Table: Gives Next Hop

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

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

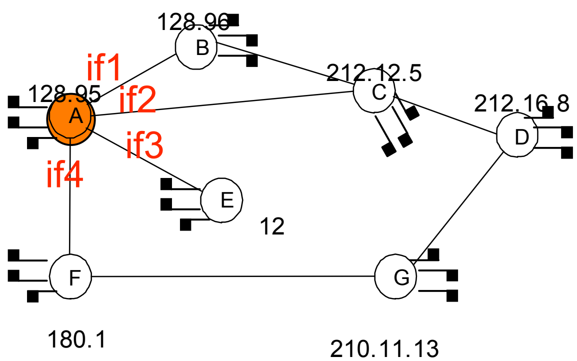


"B" == "128.96", etc

Forwarding Table: Maps IP address to Link

Dest	Next Hop	Other.
B	B	
C	C	
D	C	
E	E	
F	F	
G	F	

Routing Table @ A (128.95)



Speedy Lookup!

Dest	Link
B	if1
C	if2
D	if2
E	if3
F	if4
G	if4

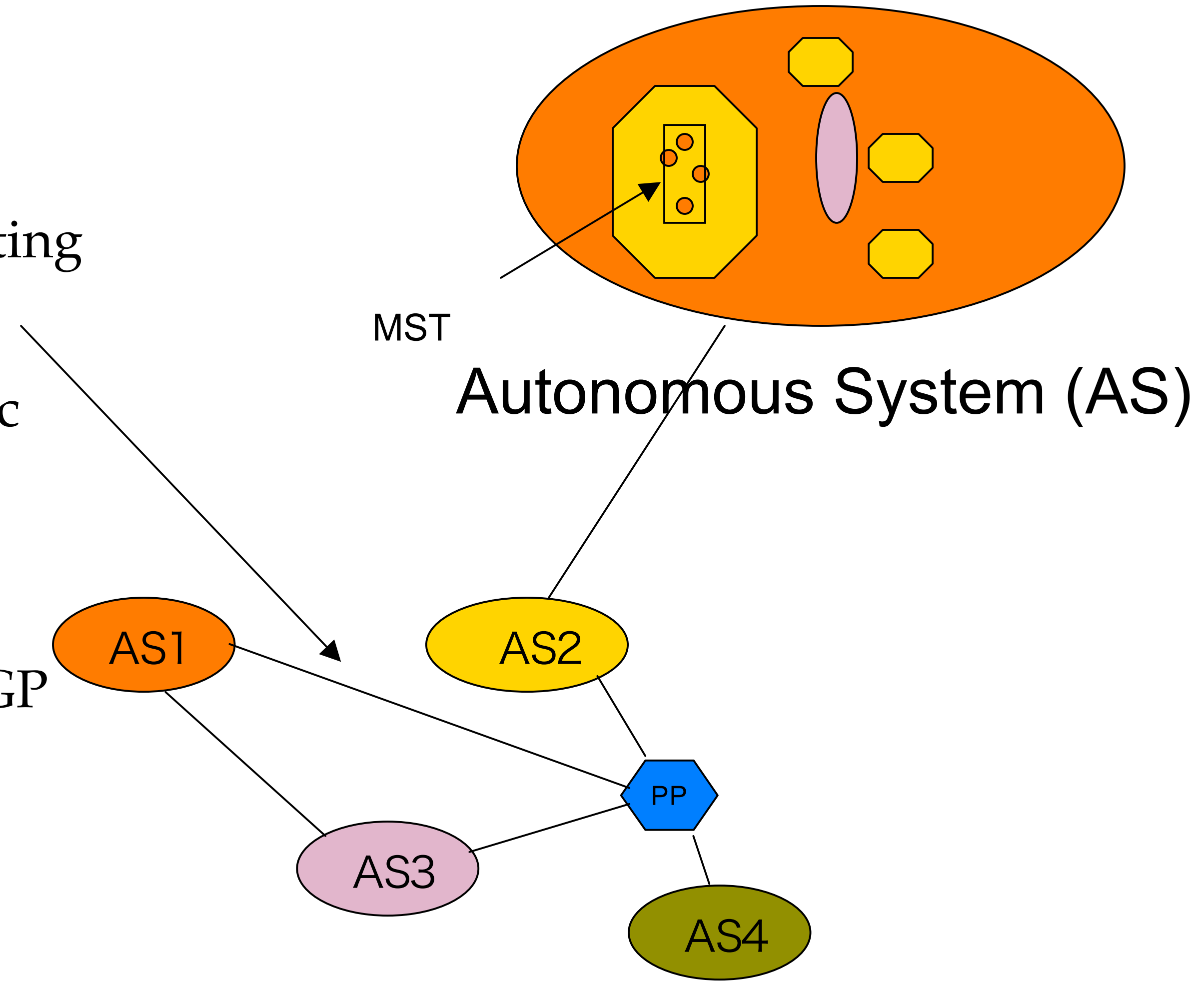
Forwarding Table @ A (128.95)

Routing Issues

- Automatic
- Adaptive
  - Failures or changes?
  - Node and link failures, plus message loss
  - We will use distributed algorithms
- Best path
  - Defining "best" is slippery
    - Especially as a universal
- Scale
  - Minimize # control messages and routing table size per router
- Policy
    - "A should not route to B unless there is an emergency?"
    - "A should route to B iff B routes to A."
  - Manual switching
- Scope
  - Intradomain and Interdomain

Intra vs. Inter

- Intradomain
  - Local traffic
  - Interior Routing
- Interdomain
  - Transit traffic
  - Manual and Automatic routing
    - OSPF, BGP



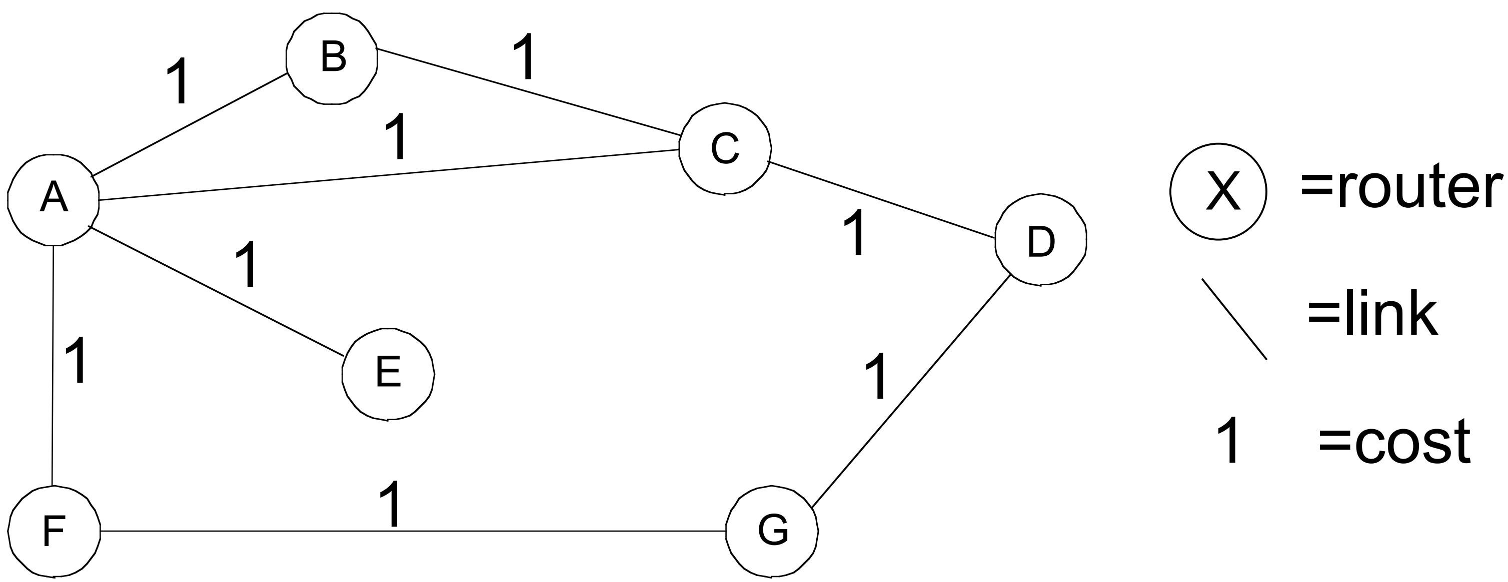
Some Pitfalls

- 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

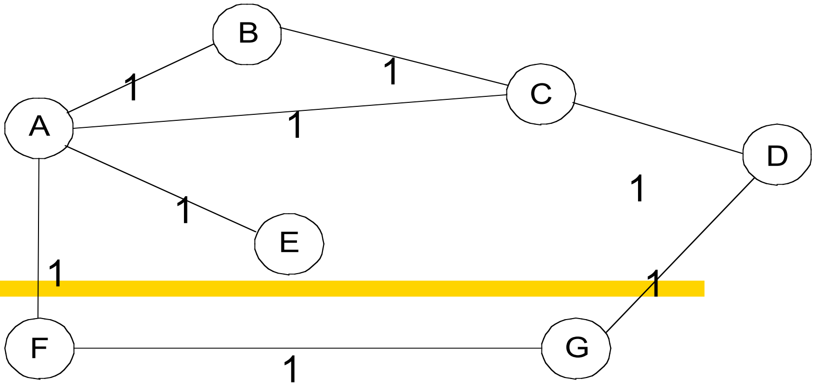
- Routing is essentially a problem in graph theory



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
  - Neighbors update internal routing tables
  - Do it again

Global View



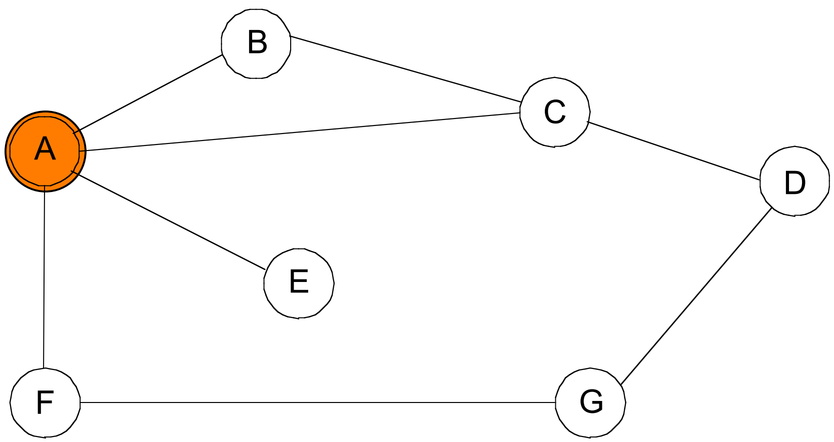
Initial
View From

	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

DV Algorithm

- Each router maintains a vector of costs to all destinations (networks) as well as a routing table
  - Initialize neighbor entries 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

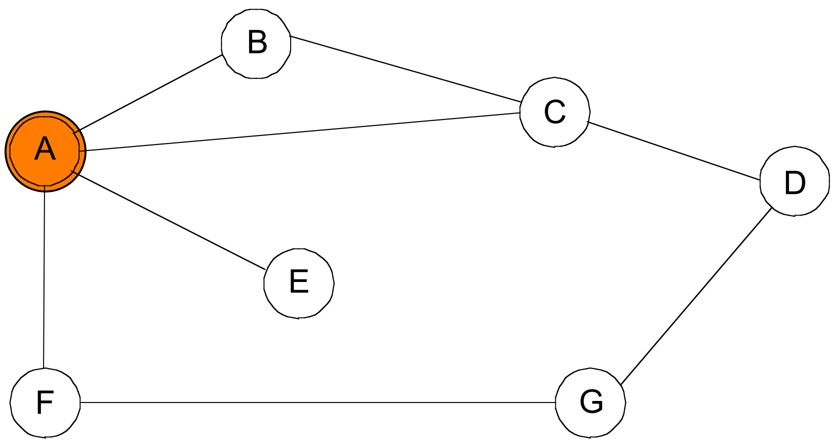


Initial table at A

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

DV Example – Final Table at A

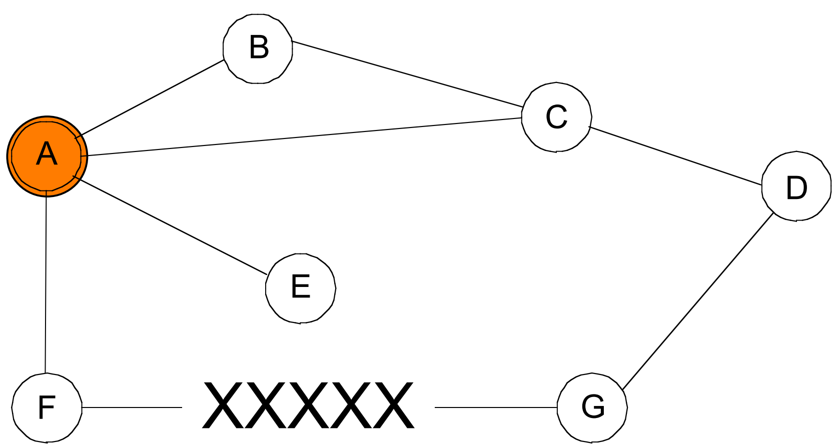
- Reached in a single round of updates from C and F



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

Adapting to Change

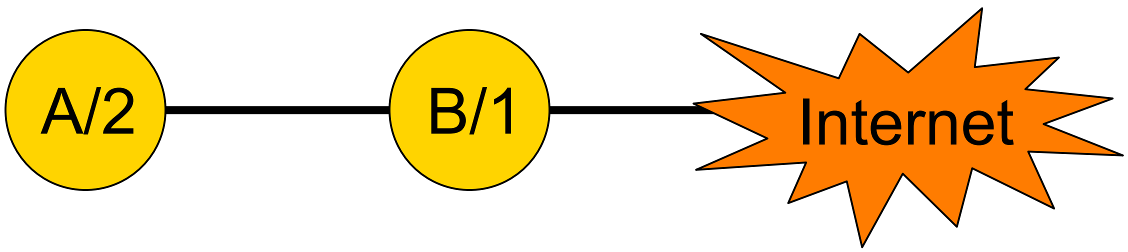
- One scenario: Suppose link between F and G fails
  1. F notices failure, sets its cost to G to infinity and tells A
  2. A sets its cost to G to infinity too, since it learned it from F
  3. A learns route from C with cost 2 and adopts it



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

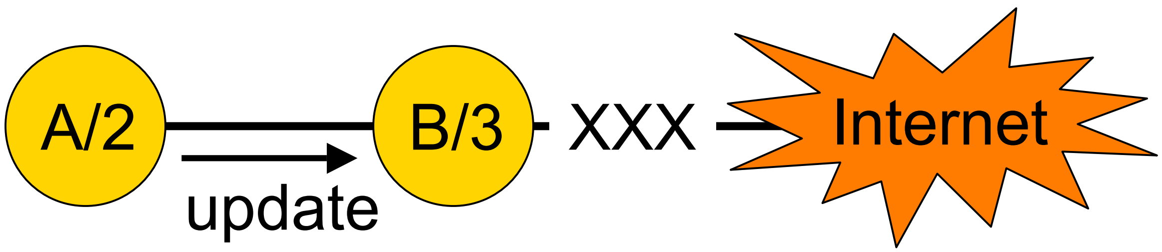
- Costs in nodes are to reach Internet



- Now link between B and Internet fails …

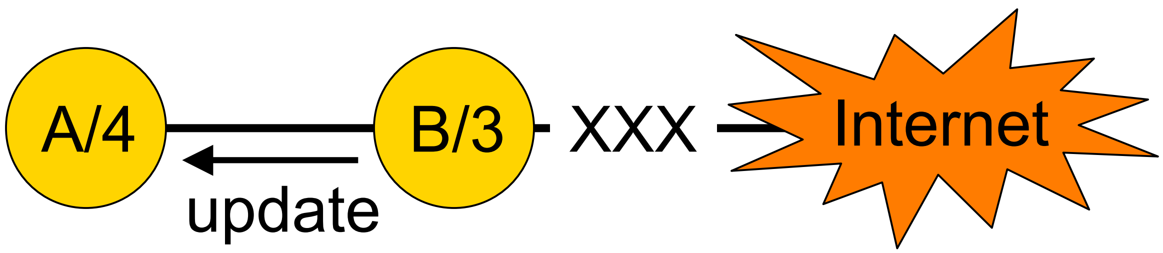
Count To Infinity Problem

- B hears of a route to the Internet via A with cost 2
  - Maybe B's update to A was lost, or wasn't sent yet
- So B switches to the "better" (but wrong!) route



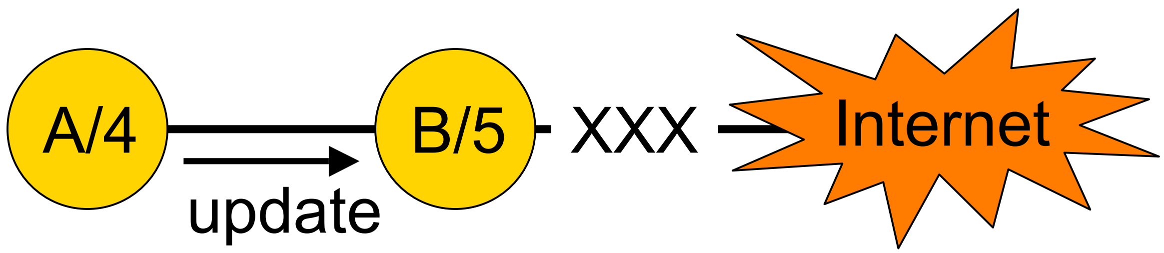
Count To Infinity Problem

- 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
  - Data and routing packets get lost

Split Horizon

- Solves simple count-to-infinity problem

- Router never advertises the cost of a destination back to to the destination's next hop – that's where it learned it from!
- Poison reverse: go even further – advertise back infinity to keep the receiver from routing "right" when they should go "left"
  - In other words, A's DV message to B sets distance(Y)=infinity if A's next hop for Y is B.

- However, DV protocols still subject to the same problem with more complicated topologies (eg, 3 way loops with lost messages)
  - 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

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)