Routing

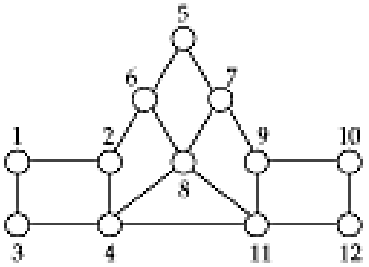
An Engineering Approach to Computer Networking

What is it?

- Process of finding a path from a source to every destination in the network
- Suppose you want to connect to Antarctica from your desktop
  - what route should you take?
  - does a shorter route exist?
  - what if a link along the route goes down?
  - what if you're on a mobile wireless link?
- Routing deals with these types of issues

Basics

- A *routing protocol* sets up a *routing table* in *routers* and *switch controllers*



ROUTING TABLE AT 1

Destination	Next hop	Destination	Next hop
1	—	7	2
2	2	8	2
3	3	9	2
4	3	10	2
5	2	11	3
6	2	12	3

- A node makes a *local* choice depending on *global* topology: this is the fundamental problem

Key problem

- How to make correct local decisions?
  - each router must know *something* about global state
- Global state
  - inherently large
  - dynamic
  - hard to collect
- *A routing protocol must intelligently summarize relevant information*

Outline

- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

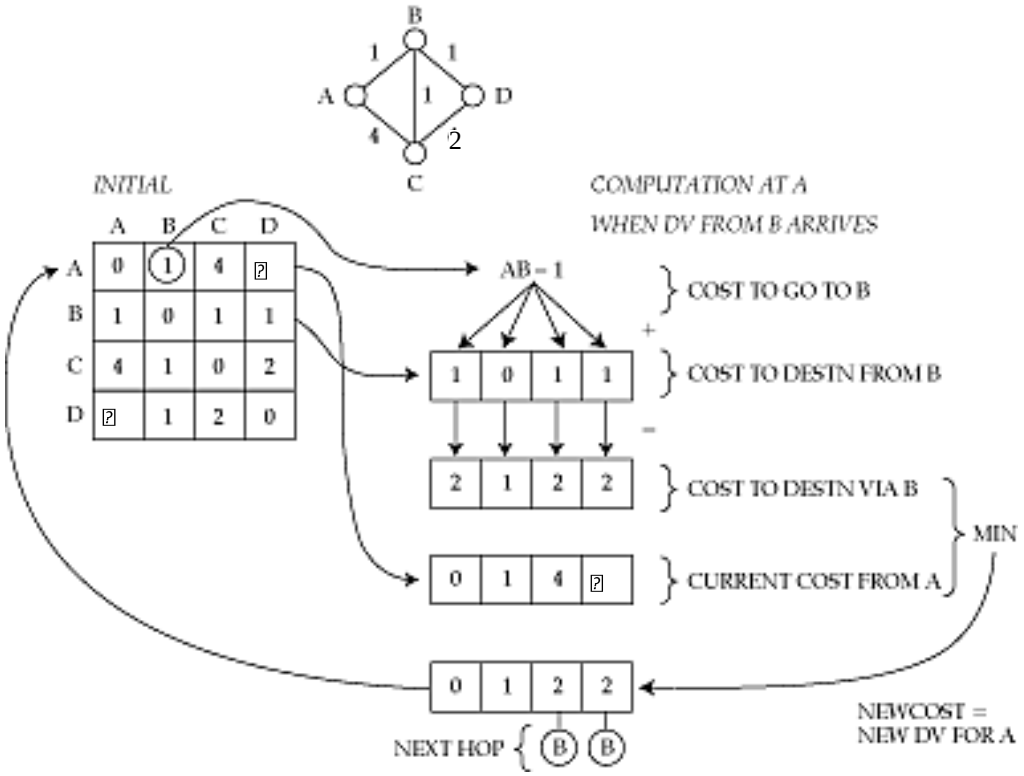
Distance vector routing

- Environment
  - links and routers unreliable
  - alternative paths scarce
  - traffic patterns can change rapidly
- Two key algorithms
  - distance vector
  - link-state
- Both assume router knows
  - address of each neighbor
  - cost of reaching each neighbor
- Both allow a router to determine global routing information by talking to its neighbors

Basic idea

- Node tells its neighbors its best idea of distance to *every* other node in the network
- Node receives these *distance vectors* from its neighbors
- Updates its notion of best path to each destination, and the next hop for this destination
- Features
  - distributed
  - adapts to traffic changes and link failures
  - suitable for networks with multiple administrative entities

Example

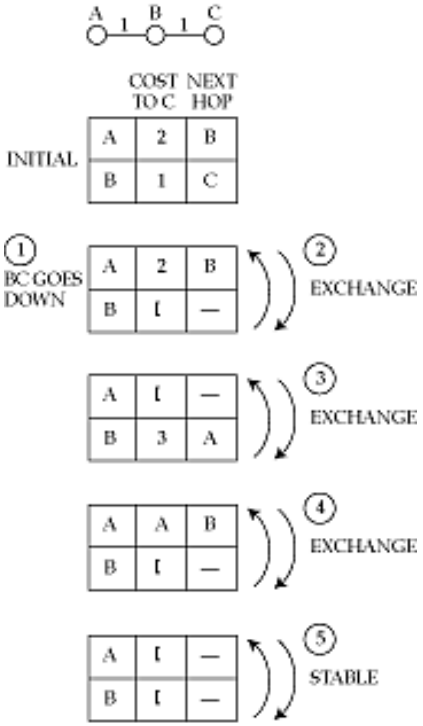

B
1
1
A
1
D
4
2
C
INITIAL
A B C D
A 0 1 4 ?
B 1 0 1 1
C 4 1 0 2
D ? 1 2 0
COMPUTATION AT A
WHEN DV FROM B ARRIVES
AB = 1
COST TO GO TO B
+
1 0 1 1
COST TO DESTN FROM B
=
2 1 2 2
COST TO DESTN VIA B
MIN
0 1 4 ?
CURRENT COST FROM A
0 1 2 2
NEXT HOP { B B
NEWCOST =
NEW DV FOR A

Why does it work

- Each node knows its true cost to its neighbors
- This information is spread to its neighbors the first time it sends out its distance vector
- Each subsequent dissemination spreads the truth one hop
- Eventually, it is incorporated into routing table everywhere in the network
- Proof: Bellman and Ford, 1957

Problems with distance vector

- Count to infinity



Dealing with the problem

- Path vector
  - DV carries path to reach each destination
- Split horizon
  - never tell neighbor cost to X if neighbor is next hop to X
  - doesn’t work for 3-way count to infinity (see exercise)
- Triggered updates
  - exchange routes on change, instead of on timer
  - faster count up to infinity
- More complicated
  - source tracing
  - DUAL