What is “Routing”?

Routing algorithm that part of the network layer responsible for deciding on which output line to transmit an incoming packet

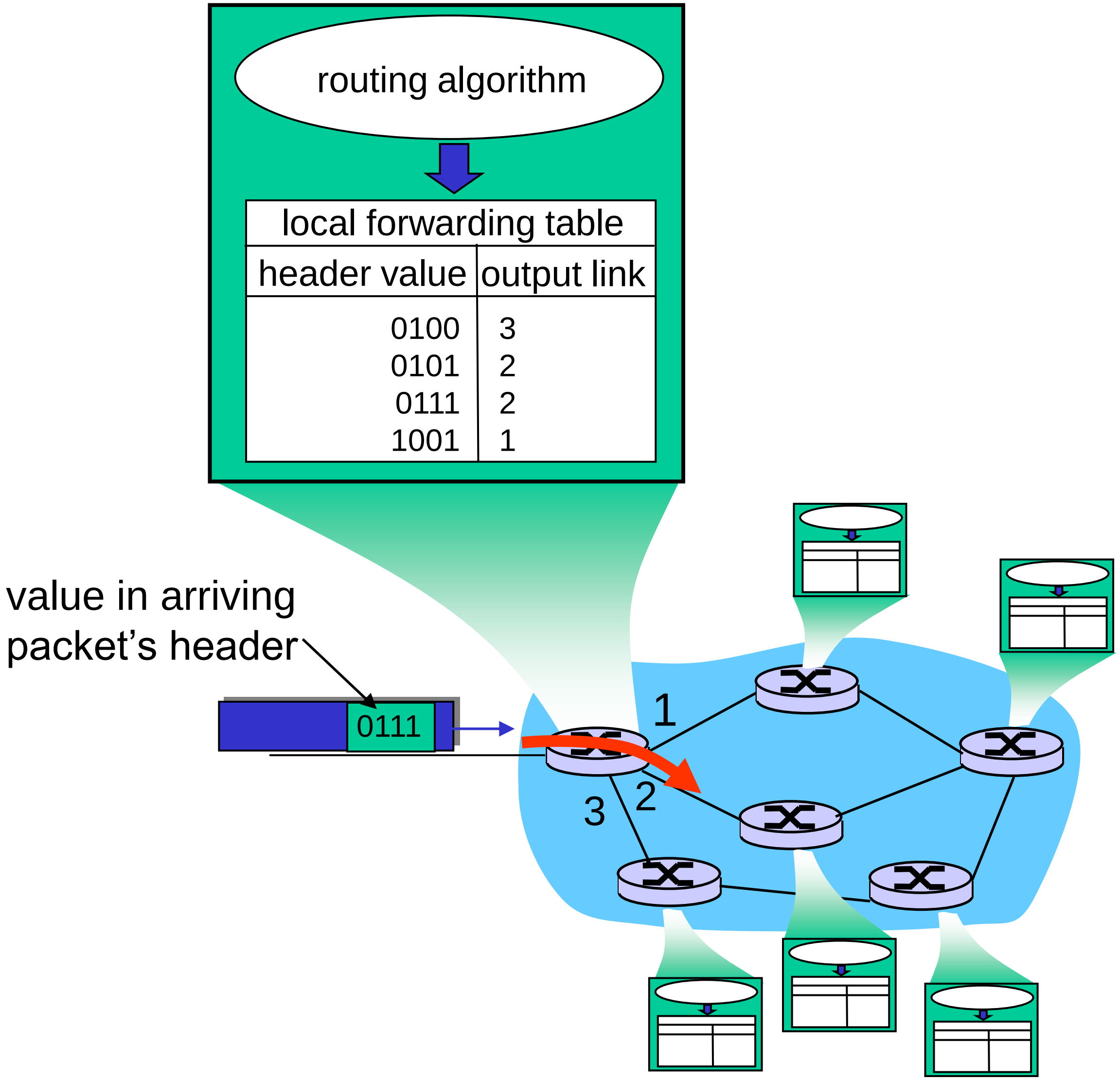
Adaptive Routing based on current measurements of traffic and/or topology

- centralized, isolated, distributed

Non-adaptive Routing

- flooding
- static routing {shortest path}

Interplay between routing and forwarding

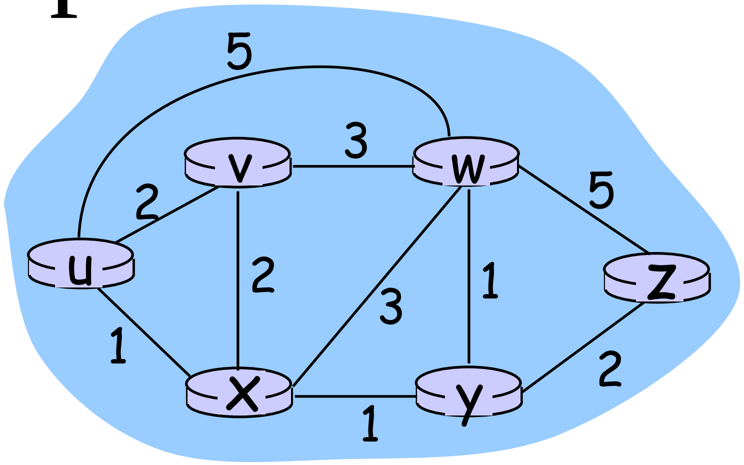


local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

value in arriving packet's header

Graph abstraction

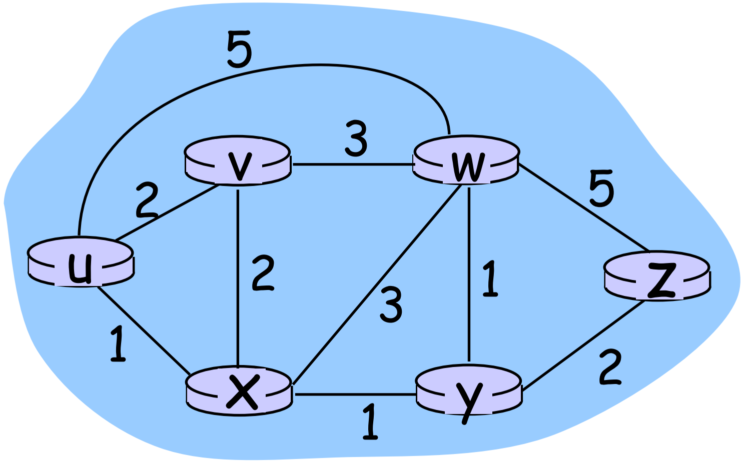


Graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Graph abstraction: costs



- $c(x,x')$ = cost of link (x,x')

  - e.g., $c(w,z) = 5$

- cost could always be 1, or inversely related to bandwidth, or related to congestion

Cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds least-cost path

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors

Static or dynamic?

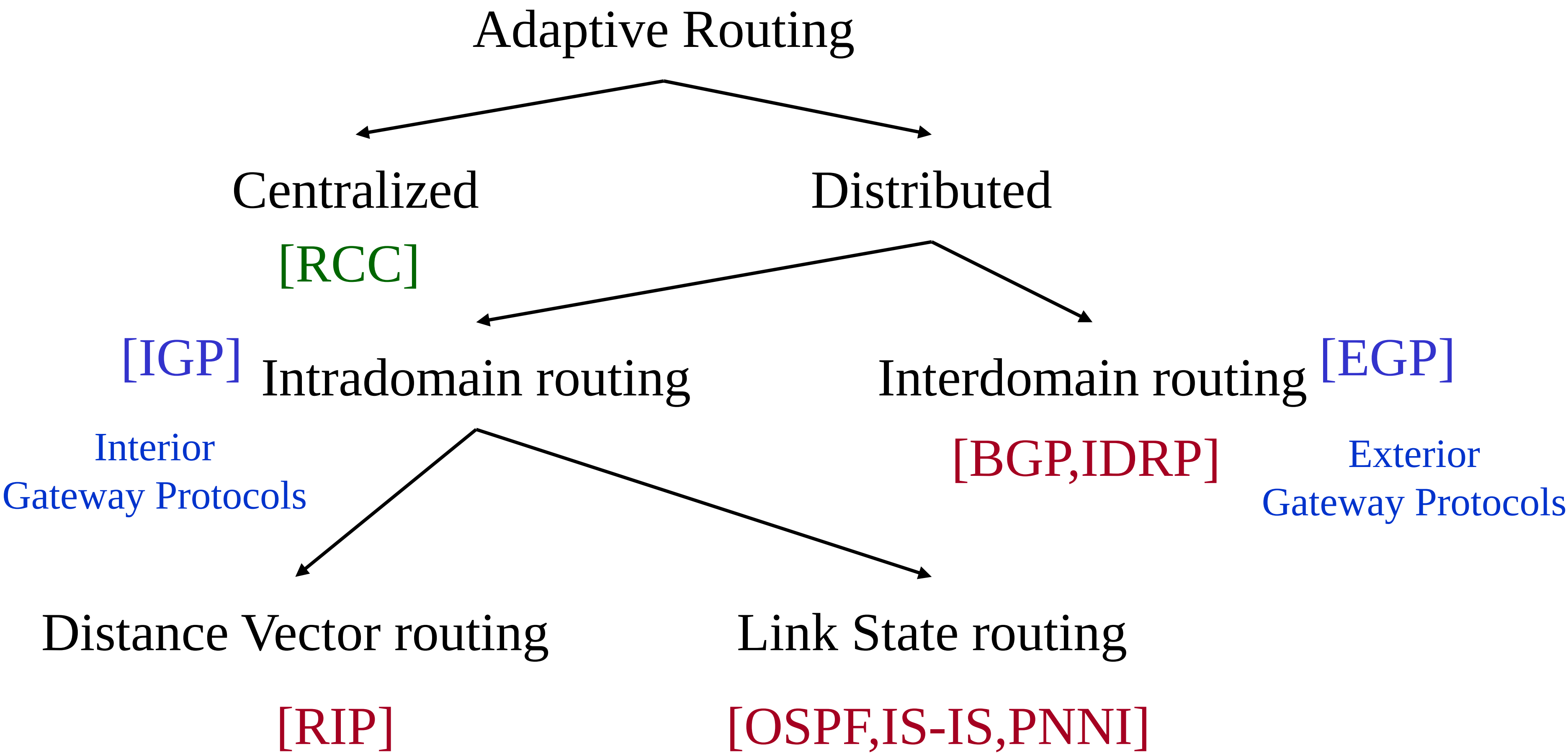
Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

Internetwork Routing



Adaptive Routing

Centralized

[RCC]

Distributed

[IGP] Intradomain routing

Interior
Gateway Protocols

Interdomain routing [EGP]

[BGP,IDRP]

Exterior
Gateway Protocols

Distance Vector routing

[RIP]

Link State routing

[OSPF,IS-IS,PNNI]

Distance Vector Routing

- Historically known as the *old* ARPANET routing algorithm {also known as *Bellman-Ford algorithm*}.

Basic idea: each network node maintains a table containing the ***distance*** between itself and **ALL** possible destination nodes.

- Distance are based on a chosen metric and are computed using information from the **neighbors’** distance vectors.

Metric: *usually hops or delay*

Distance Vector

Information needed by node :

each router has an ID

associated with each link connected to a router there is a link cost (static or dynamic) *the metric issue!*

Each router starts with:

DV = 0 {distance measure to itself}

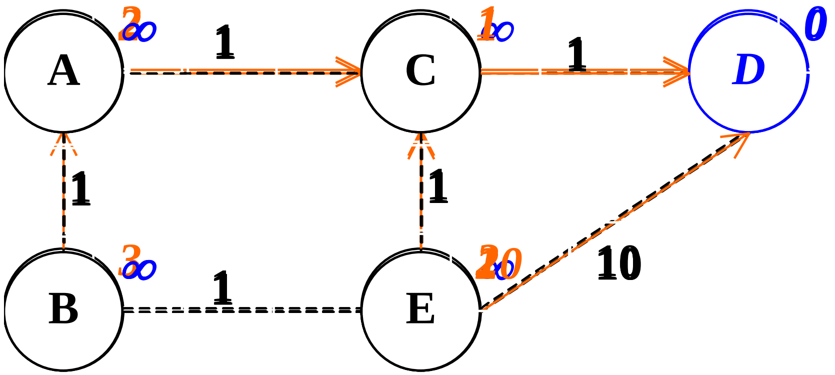
DV = infinity number {for ALL other destinations}

Distance Vector Algorithm [Perlman]

1. Router transmits its *distance vector* to each of its neighbors.
2. Each router receives and saves the most recently received *distance vector* from each of its neighbors.
3. A router *recalculates* its distance vector when:
   a. It receives a *distance vector* from a neighbor containing different information than before.
   b. It discovers that a link to a neighbor has gone down.

Distance-vector paradigm

- Based on distributed Bellman-Ford algorithm
- Each router maintains its distance and next-hop to the destination
- Method of choosing next hop: shortest path



Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node

Distance Vector Routing: overview

Iterative, asynchronous:

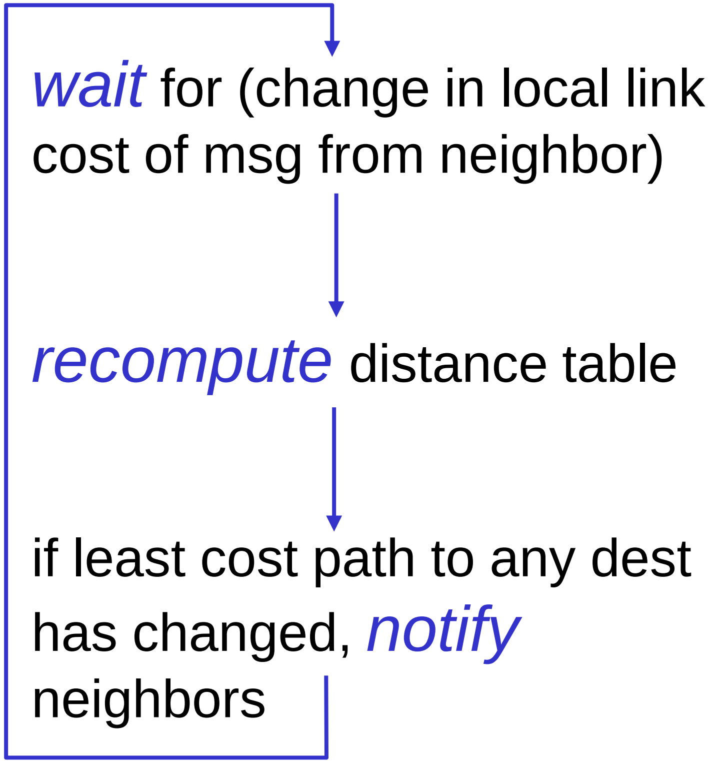
each local iteration caused by:

- local link cost change
- message from neighbor: its least cost path change from neighbor

Distributed:

- each node notifies neighbors *only* when its least cost path to any destination changes
  - neighbors then notify their neighbors if necessary

Each node:



wait for (change in local link cost of msg from neighbor)

recompute distance table

if least cost path to any dest has changed, *notify* neighbors