Module 4

(Lecture – 6)

(Network Layer: Router architecture; Internet Protocol (IP) - Forwarding and Addressing in the Internet; Routing algorithms - Link-state routing, Distance vector routing, Hierarchical routing; Routing in the Internet - RIP, OSPF, BGP; Broadcast & multicast routing; ICMP; Next Generation IP - IPv6)

Dr. Nirnay Ghosh

Assistant Professor

Department of Computer Science & Technology

IIEST, Shibpur

Routing Algorithm – The Distance-Vector (DV) Routing Algorithm

- Properties:
  - Distributed: each node receives some information (distance vectors) from one or more of its directly attached neighbors - performs a calculation - distributes the results of its calculation back to its neighbors
  - Iterative: algorithm continues until no more information is exchanged between neighbors
  - Asynchronous: does not require all of the nodes to operate in lockstep with each other
- *$d_x(y)$*: least-cost path from node *x* to node *y*
- For all neighbors *v* of *x*, least cost given by the Bellman-Ford equation is as follows:

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\},$$

- Solution to the Bellman-Ford equation provides the entries for next hops in node *x's* forwarding table

```
Initialization:
   for all destinations y in N:
      Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
   for each neighbor w
      Dw(y) = ? for all destinations y in N
   for each neighbor w
      send distance vector Dx = [Dx(y): y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      Dx(y) = minv{c(x,v) + Dv(y)}

   if Dx(y) changed for any destination y
      send distance vector Dx = [Dx(y): y in N] to all neighbors

forever
```

Distance-Vector (DV) Algorithm at each Node *x*

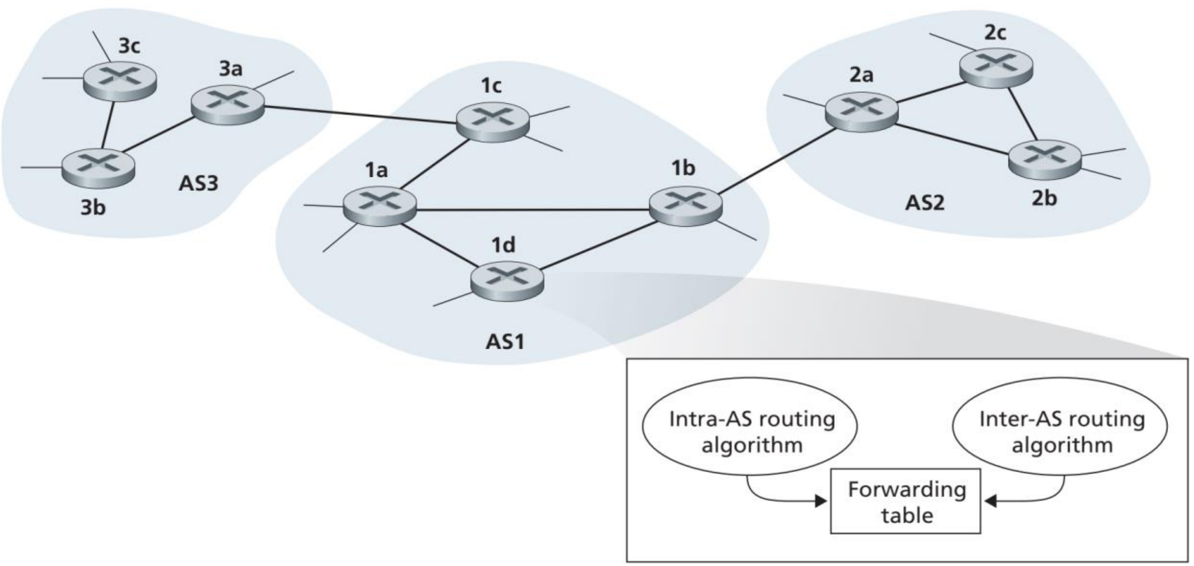
- Each node *x* begins with a vector of cost estimates from *itself* to *all* other nodes, *y,* in *N* => ***D_x*** = [*$D_x(y)$*: *y in N*]
- *x* shares ***D_x*** with its neighbors
- It receives and saves the ***D_v*** from its neighbors
- It updates ***D_x*** by using the distance vectors received from neighbors
- It sends the updated vector to neighbors, which in turn update their own distance vectors
- Limitation: routing loop leads to the count-to-infinity problem – occurs due to asynchronous update of DVs following a changes in link-costs

Comparison of LS and DV Routing Algorithms

Attribute	LS Routing Algorithm	DV Routing Algorithm
General Principle	Each node talks with *all* other nodes (via broadcast), but it tells them *only* the costs of its directly connected links.	Each node talks with *only* its directly connected neighbors and provides least-cost estimate from itself to *all* nodes in the network.
Message Complexity	It requires $O(\|N\|.\|E\|)$ messages to be sent, where N is the set of nodes (routers) and E is the set of links connecting the nodes. Whenever a link cost changes, the new link cost must be sent to all nodes.	When link costs change, the DV algorithm will propagate the result only if the new cost changes the least-cost path for one of the nodes attached to that link.
Speed of convergence	Implementation of LS is an $O(\|N\|^2)$ algorithm requiring $O(\|N\|.\|E\|)$ messages.	The DV algorithm can converge slowly and can have routing loops due to the *count-to-infinity* problem.
Robustness	Each node is computing only its own forwarding tables; other nodes are performing similar calculations for themselves. Route calculations are separated under LS, providing a degree of robustness in situations of link or node failures.	At each iteration, a node's calculation in DV is passed on to its neighbor and then indirectly to its neighbor's neighbor on the next iteration. As a result, an incorrect node calculation can be diffused through the entire network under DV. This makes DV routing less robust against random node/link failure.

Hierarchical Routing

- Limitations of LS and DV Routing Algorithms
  - Scale of the present day networks
    - Storage limitation for LS routing algorithm
    - Bandwidth bottleneck: broadcasting LS update
    - Presence of large number of routers: convergence problem for DV routing algorithm
  - Administrative autonomy
    - Autonomy in terms of administering organizational network - ensuring connection to the outside networks
- Organize routers into Autonomous Systems (ASs)
  - Routers under same administrative control
    - Operated by the same ISP or belong to the same company network
  - Intra-autonomous system routing protocol: all routers run the same routing algorithm (LS or DV algorithm)
  - Gateway routers – forwards packets to destinations outside the AS
  - Solves the scalability and administrative autonomy problems

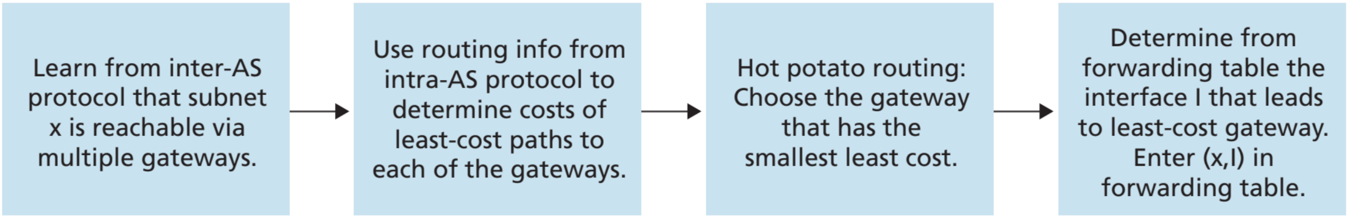


Interconnected Autonomous Systems

- Each AS usually contains multiple gateway routers – which router to choose for forwarding inter-AS packets?
- Inter-autonomous routing protocol
  - Communicating ASs must run the same routing (e.g., BGP4)
  - Each AS learns the destination reachability via other ASs
  - Propagates this to all routers within that AS

Hierarchical Routing

- Each router receives information from both intra-AS and inter-AS routing algorithms
  - Uses this information to configure the local forwarding table
- To send packets to any subnet *x* in another AS, the router needs to configure its forwarding table
- Suppose the local AS has multiple gateway routers via which *x* is reachable
- Router uses the *hot potato routing* to determine the gateway router for inter-AS routing

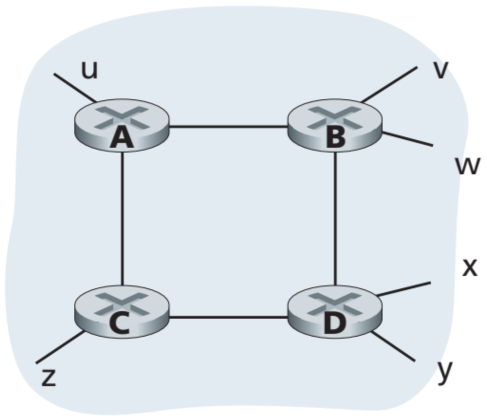


Steps in Adding an Outside-AS Destination in a Router's Forwarding Table

- Hot potato routing: choose the gateway with smallest router-to-gateway cost among all gateways with a path to the destination
- Router adds the path information in its forwarding table
- AS learns about a destination from a neighboring AS through advertisement – governed by policies based on economic and technical issues

Intra-AS Routing in the Internet: RIP

- RIP: Routing Information Protocol
- One of the earliest intra-AS Internet routing protocols
- Uses Distance-Vector (DV) protocol - cost metric: hop count
  - Hop: no. of subnets traversed along the shortest path from source router to the destination subnet
  - Maximum hop count: limited to 15
  - Used in ASs that are fewer than 15 hops in diameter
- Routers exchange two types of messages
  - RIP request: router's request about its neighbor's cost to a given destination (subnet)
  - RIP response message/RIP advertisement: messages for exchanging routing updates between neighbors (every 30 secs interval)
    - Returns a list of up to 25 destination subnets within the AS
    - Sender's distance to each of those subnets



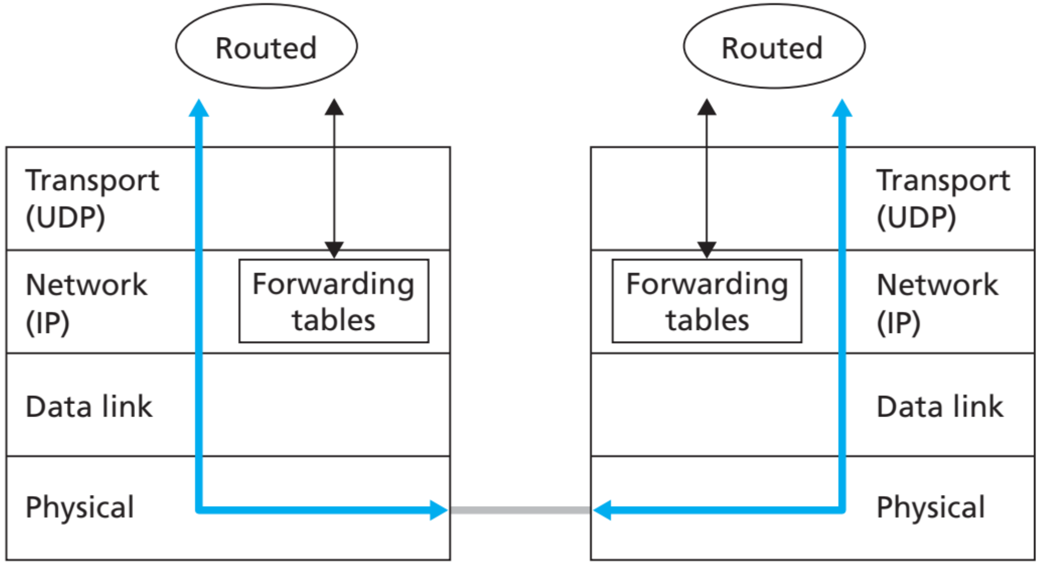
Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Number of hops from source router *A* to various subnets

- Each router maintains a RIP table (routing table)
  - RIP table content – router's distance vector and forwarding table
- Routing table: 3 columns
  - Destination subnet, next router along the shortest path, number of hops
- Neighbor not reachable: if any neighbor does not exchange its routing updates at least once in 180 seconds
  - Router incorporates this information (neighbor unreachable) into its routing table
  - Propagates this information to other neighboring routers by sending advertisements

Intra-AS Routing in the Internet: RIP

- RIP – included in the Berkeley Software Distribution (BSD) version of the UNIX supporting TCP/IP
- Typically implemented as an application layer process – *“routed”* (route-dee)
  - Runs in neighboring routers
  - Uses UDP Port 520
  - Capable of manipulating the routing tables within the UNIX kernel
  - Sends and receives messages over a standard socket
  - RIP request/response: implemented as UDP segment
  - IP datagram carries the UDP segment



Implementation of RIP as the *routed* (route dee) daemon