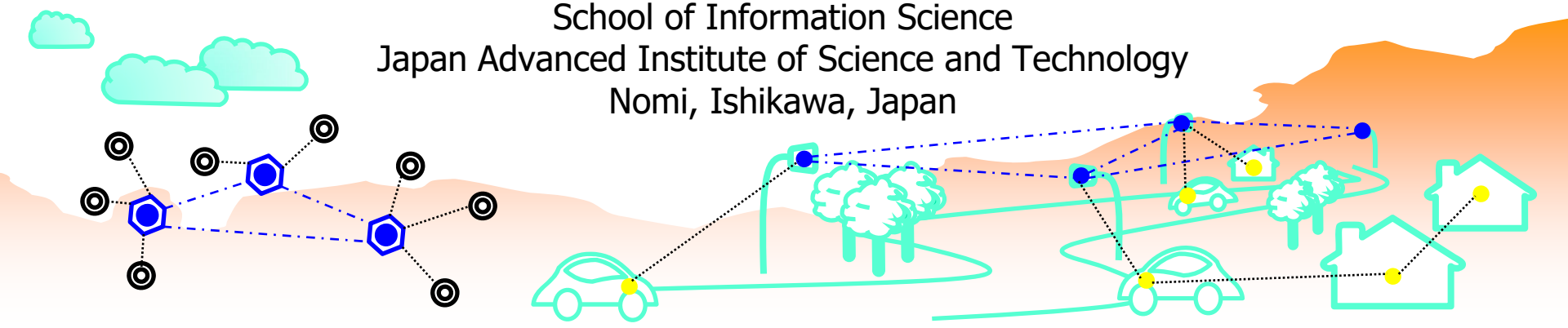
I226
Computer Networks

Chapter 8

Name Service and Internetworking

Assoc. Prof. Yuto Lim

School of Information Science
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan



Objectives of this Chapter

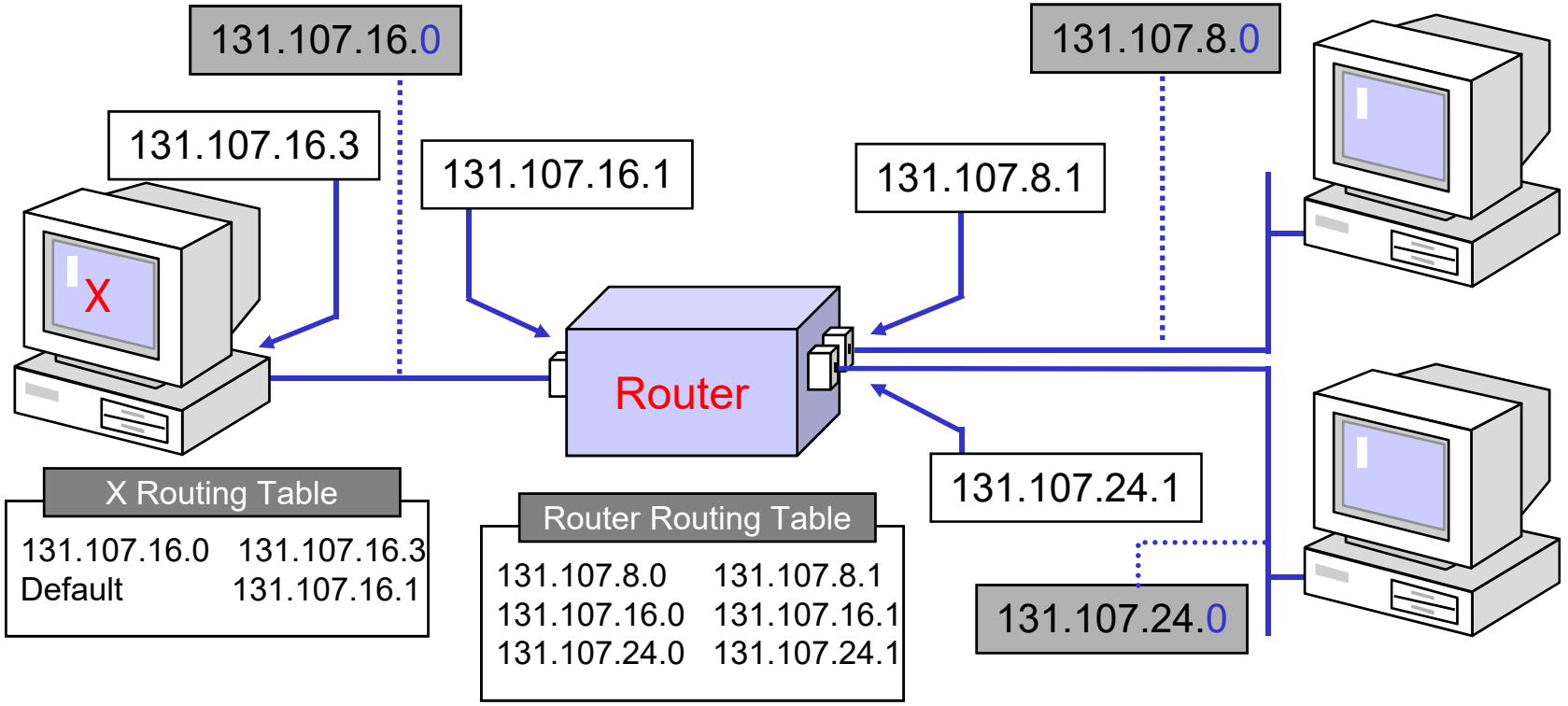
- Explain two differences routing methods: distance vector and link state
- Give the knowledge of Internet routing protocols: RIP, OSPF, and BGP
- Provide the knowledge of directory service, network information service, LDAP and DNS
- Explain the DHCP and how it works

Outline

- Introduction to Routing
- Graph Model
- Shortest Path Method
  - Distance Vector (DV) and Link State (LS)
- Internet Routing
  - Intra-AS Routing: Interior Gateway Protocol (IGP)
    - Routing Information Protocol (RIP)
    - Open Shortest Path First (OSPF)
  - Inter-AS Routing: Exterior Gateway Protocol (EGP)
    - Border Gateway Protocol (BGP)
- Directory Service and Naming
- Network Information Service (NIS)
- Lightweight Directory Access Protocol (LDAP)
- Domain Name System (DNS)
- Dynamic Host Configuration Protocol (DHCP)

What Is Routing?

- Process of sending packets via routers to other networks
- Routing table defines paths to other networks



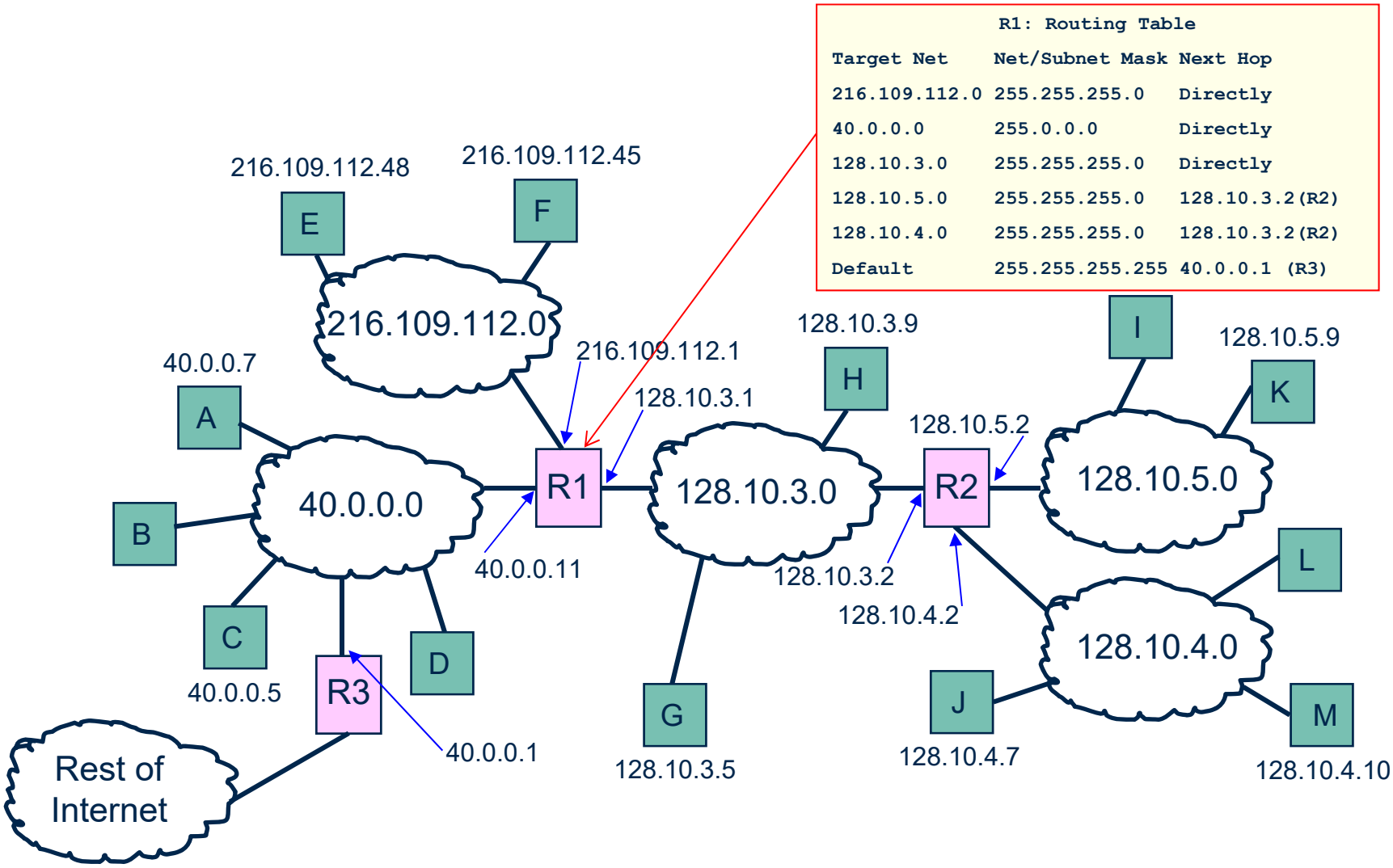
Routing Table

- Every router contains a routing table with the following entries
  - Target Network – destination network
  - Net/Subnet Mask – a word that contains 1's for the bits that correspond to the network number of subnet number
  - Next Hop – the next router to use for this destination or "directly" if the packet can be delivered directly
- Special entry called "default" that is used when the destination does not match any of the other entries in the table
- Example of a routing table:

R1: Routing Table

Target Net	Net/Subnet Mask	Next Hop
216.109.112.0	255.255.255.0	Directly
40.0.0.0	255.0.0.0	Directly
128.10.3.0	255.255.255.0	Directly
128.10.5.0	255.255.255.0	128.10.3.2(R2)
128.10.4.0	255.255.255.0	128.10.3.2(R2)
Default	255.255.255.255	40.0.0.1 (R3)

IP Networks



Routing Table (cont.)

- Entries are sorted from more specific to more general
- First entry that matches is the one used
- Routing entry "`i`" matches if the following is true:
  - `(target-ip & net_mask[i]) == target_net[i]`
  - where `target-ip` is the target IP address of the packet. So we use `next_hop[i]`
- Matching is done as follows:

```
IPAddr next_hop(IPAddr target_ip) {
  for (int i = 0; i < nentries - 1; i++) {
    if ( (target-ip & net_mask[i]) == target_net[i]) {
      return next_hop[i];
    }
  }
  // No match. Return default
  return next_hop[nentries - 1];
}
```

Examples

R1: Routing Table

Target Net	Net/Subnet Mask	Next Hop
216.109.112.0	255.255.255.0	Directly
40.0.0.0	255.0.0.0	Directly
128.10.3.0	255.255.255.0	Directly
128.10.5.0	255.255.255.0	128.10.3.2(R2)
128.10.4.0	255.255.255.0	128.10.3.2(R2)
Default	255.255.255.255	40.0.0.1(R3)

- Example1: Router R1 receives a packet with destination 128.10.4.3
- Example2: Router R1 receives a packet with destination 128.10.3.9
- Example3: Router R1 receives a packet with destination 43.0.0.5

Answer 1:

① It tries to match (128.10.4.3 & 255.255.255.0)==128.10.4.0 that is different than 216.109.112.0 and so on

② Until it finds the entry 128.10.4.0 that matches the target net 128.10.4.3 so the next hop is 128.10.3.2(R2)

Answer 2:

① It tries to match (128.10.3.9 & 255.255.255.0)==128.10.3.0 that is different than 216.109.112.0 and so on

② Until it finds the entry 128.10.3.0 that matches the target net 128.10.3.9 so the next hop is to deliver the packet directly since R1 is connected to that LAN

Answer 3:

① It tries to match (43.0.0.5 & 255.255.255.0)==43.0.0.0 that is different than 216.109.112.0 and so on

② Since no entry matches, it uses the default entry to deliver the packet. The next hop is 40.0.0.1(R3)

Routing Protocol

- Each routing protocol has it's own set of features, issues and problems
- Routing protocols can be differentiated based on some key characteristics
- Each routing protocol has a different impact on network and router resources

Routing Algorithms

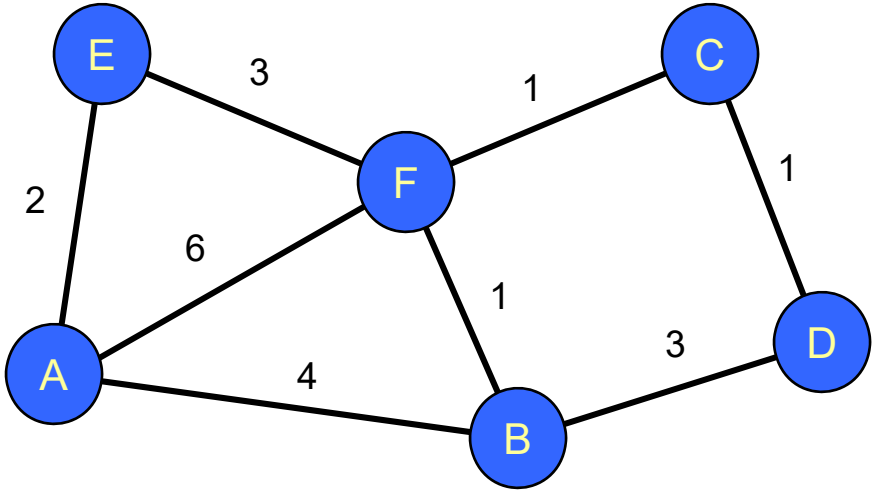
- Objective of routing algorithms is to calculate 'best' routes
- Routing algorithms for both datagrams and virtual circuits should satisfy:
  - Correctness, simplicity, stability, optimality, robustness, fairness
- Impossible to satisfy everything at the same time
- Optimization criteria:
  - Hop count, delay, throughput, "cost"
- Decision time:
  - Once per session (VCs), once per packet (datagram)
- Decision place:
  - Each node (distributed routing)
  - Central node (centralized routing)
  - Sending node (source routing)

Shortest-Path Algorithm

- Routing algorithms use a shortest-path algorithm to calculate the route with the least-cost
- 3 components:
  - Measurement – nodes (routers) measure the current characteristics such as delay, throughput, and "cost"
  - Protocol – nodes disseminate the measured information to other nodes
  - Calculation – nodes run a least-cost routing algorithm to recalculate their routes
- Goal: Given a network were each link between two nodes i and j is assigned a cost. Find the path with the least-cost between nodes i and j
- Parameters:
  - N – set of nodes
  - d_{ij} – cost of link between node i and node j; $d_{ij} = \infty$, if nodes i and j and not connected

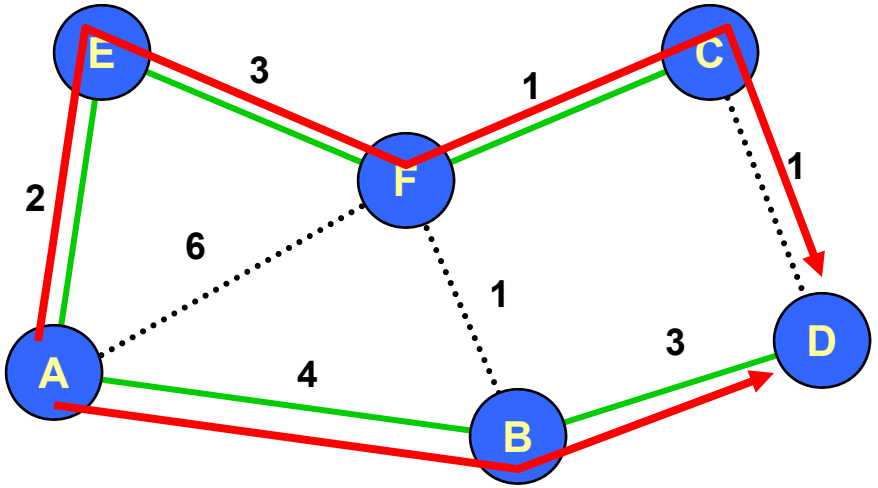
Graph Model

- Router represents as node
- Direct link between routers represented by edge
  - Symmetric links ⇒ undirected graph
- Edge cost `c(x, y)` denotes measure of a link
  - Delay, cost, or congestion level
- Task
  - Determine least-cost path from every node to every other node
  - Path cost `d(x, y)` denotes the sum of link costs



Forwarding Table for Node A

Forwarding Table for A		
Dest	Cost	Next Hop
A	0	A
B	4	B
C	6	E
D	7	B
E	2	E
F	5	E



- Properties
  - Some set of shortest paths forms tree (Shortest path spanning tree)
  - Solution is not unique. Both A-B-D and A-E-F-C-D have cost 7

Ways to Compute Shortest Paths

- Distributed way
  - Distance Vector (DV) Method
    - No one has copy of graph
    - Nodes construct their own tables iteratively
    - Each sends information about its table to neighbors
  - Link State (LS) Method
    - Every node collects complete graph structure
    - Each computes shortest paths from it
    - Each generates own routing table
- Centralized way
  - Collect graph structure in one place
  - Use standard graph algorithm
  - Disseminate routing tables

Distance Vector Method

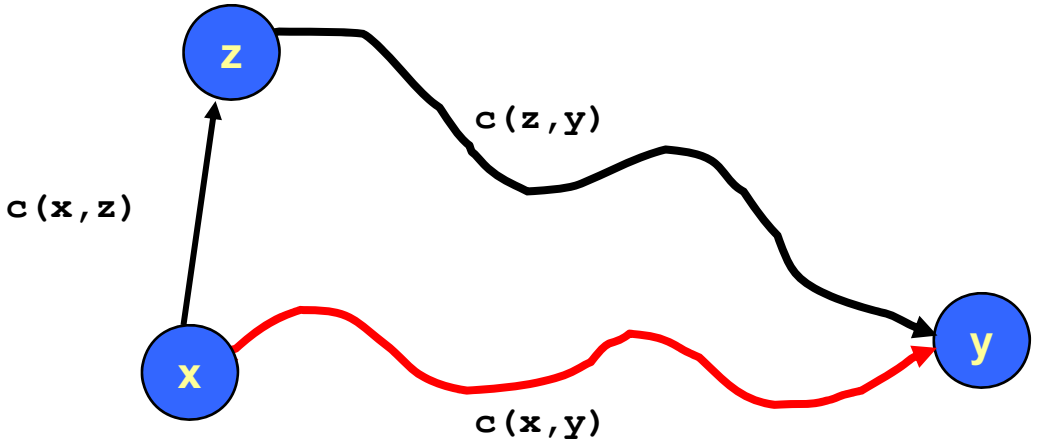
- Basic Idea
  - Each node periodically sends its own DV estimate to neighbors
  - When a node `x` receives new DV estimate from neighbor, it updates its own DV using Bellman-Ford Algorithm

`d(x,y)← min{c(x,y)+d(x,y)} for each node N except Y`

- Under "natural" conditions, the estimates of `d(x,y)` converge to the actual least cost `c(x,y)+d(x,y)`

```
1  // Bellman-Ford Algorithm
2  for every node, x
3    for every neighbor, z
4      for every destination, y
5        d(x,y) ← Update(x,y)
6      end
7    end
8  end
```

Distance Vector Update



- Update d(x, y)

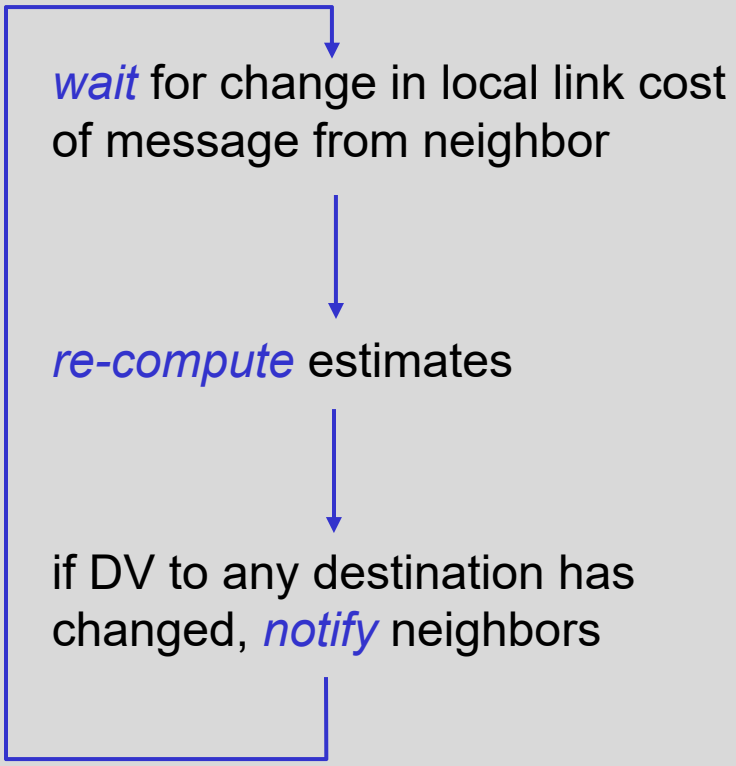
```
// Cost of path from x to y with first hop z
d(x,y) = c(x,z) + c(z,y);

if d(x,y) < c(x,y) {
  // a better path is found
   return d(x,y);
  // update new cost and next hop
   nexthop(z);
}
else {
   // old path is remained
   return c(x,y);
   // remain the old cost and next hop
   nexthop(y);
}
```

Distance Vector Iteration

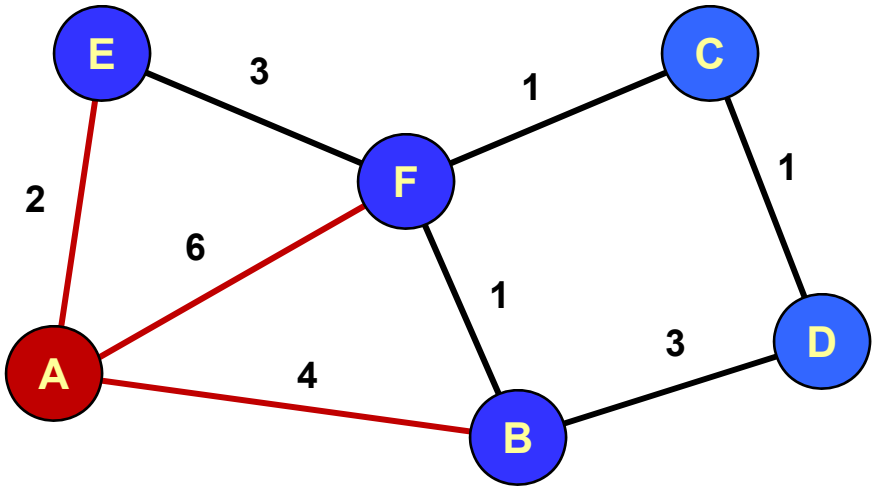
- Iterative, asynchronous
  - Each local iteration caused by
    - Local link cost change
    - DV update message from neighbor
- Distributed
  - Each node notifies neighbors only when its DV changes
    - Neighbors then notify their neighbors if necessary

Each node:



Distance Vector: Forwarding Table

Initial Forwarding Table for A		
Dest	Cost	Next Hop
A	0	A
B	4	B
C	∞	–
D	∞	–
E	2	E
F	6	F



- Forwarding table
  - At any time, entries have cost/next hop of best known path to destination
  - Only entries have directly connected to nodes
  - Use cost ∞ when no path known

At Initial Stage

Optimum 1-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	∞	–	C	∞	–
D	∞	–	D	3	D
E	2	E	E	∞	–
F	6	F	F	1	F

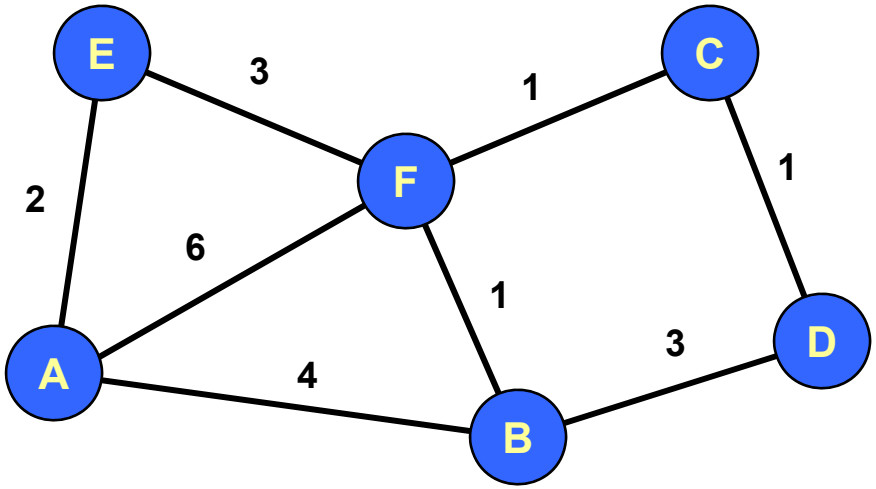


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	∞	–	A	∞	–	A	2	A	A	6	A
B	∞	–	B	3	B	B	∞	–	B	1	B
C	0	C	C	1	C	C	∞	–	C	1	C
D	1	D	D	0	D	D	∞	–	D	∞	–
E	∞	–	E	∞	–	E	0	E	E	3	E
F	1	F	F	∞	–	F	3	F	F	0	F

1st Iteration

Optimum 2-hop paths

Table for A		
Dst	Cst	Hop
A	0	A
B	4	B
C	7	F
D	7	B
E	2	E
F	5	E

Table for B		
Dst	Cst	Hop
A	4	A
B	0	B
C	2	F
D	3	D
E	4	F
F	1	F

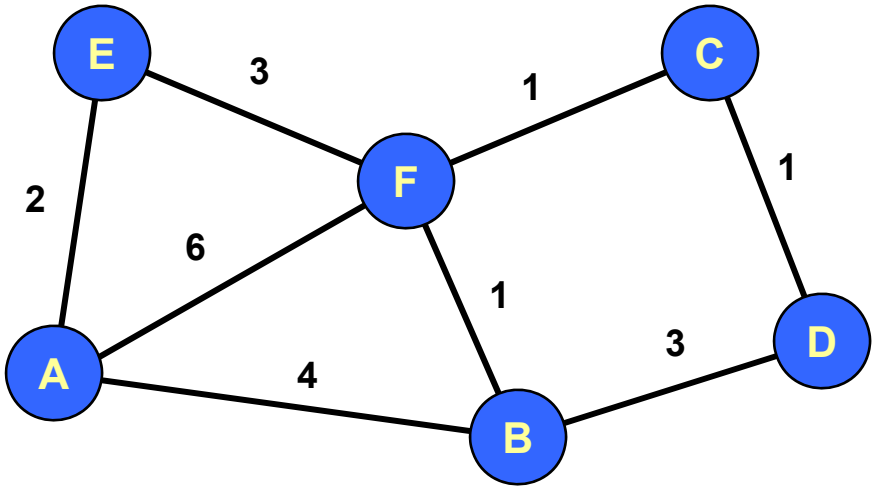


Table for C		
Dst	Cst	Hop
A	7	F
B	2	F
C	0	C
D	1	D
E	4	F
F	1	F

Table for D		
Dst	Cst	Hop
A	7	B
B	3	B
C	1	C
D	0	D
E	∞	–
F	2	C

Table for E		
Dst	Cst	Hop
A	2	A
B	4	F
C	4	F
D	∞	–
E	0	E
F	3	F

Table for F		
Dst	Cst	Hop
A	5	B
B	1	B
C	1	C
D	2	C
E	3	E
F	0	F

2nd Iteration

Optimum 3-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	6	E	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F

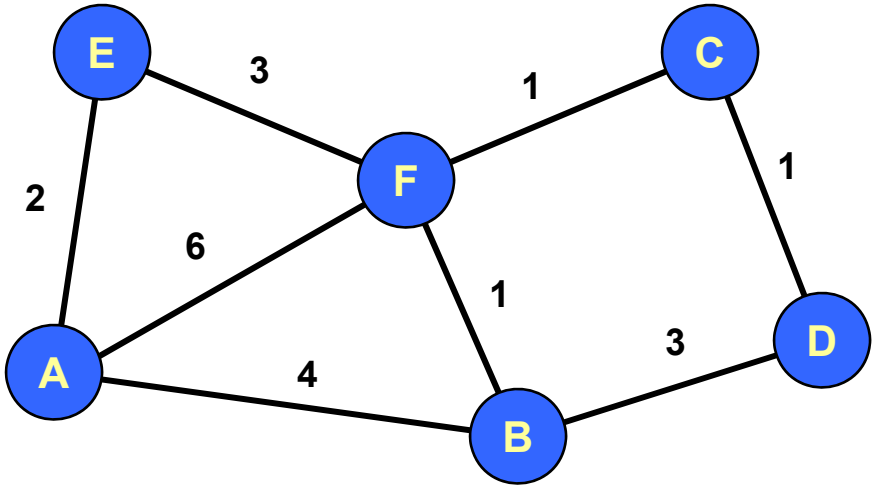
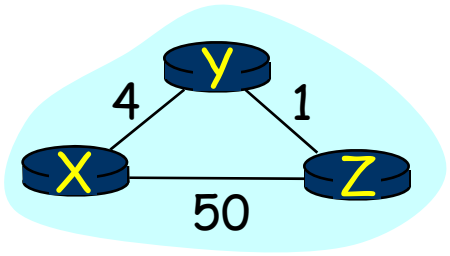


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	6	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	5	F	D	2	C
E	4	F	E	5	C	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

Example



d(x,z)= min{c(x,y)+c(y,z), c(x,z)+c(z,z)}
= min{4+1, 50+0} = 5

node x table

from	cost to x	cost to y	cost to z
x	0	4	50
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	4	5
y	4	0	1
z	50	1	0

from	cost to x	cost to y	cost to z
x	0	4	5
y	4	0	1
z	5	1	0

node y table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	4	0	1
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	4	50
y	4	0	1
z	50	1	0

from	cost to x	cost to y	cost to z
x	0	4	5
y	4	0	1
z	5	1	0

node z table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	∞	∞	∞
z	50	1	0

from	cost to x	cost to y	cost to z
x	0	4	50
y	4	0	1
z	5	1	0

from	cost to x	cost to y	cost to z
x	0	4	5
y	4	0	1
z	5	1	0

time

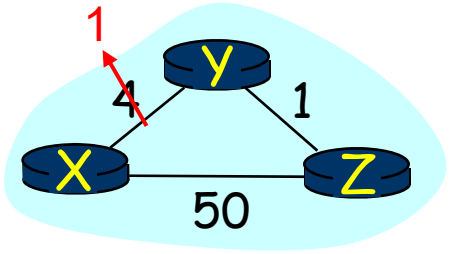
Distance Vector: Link Cost Changes (1)

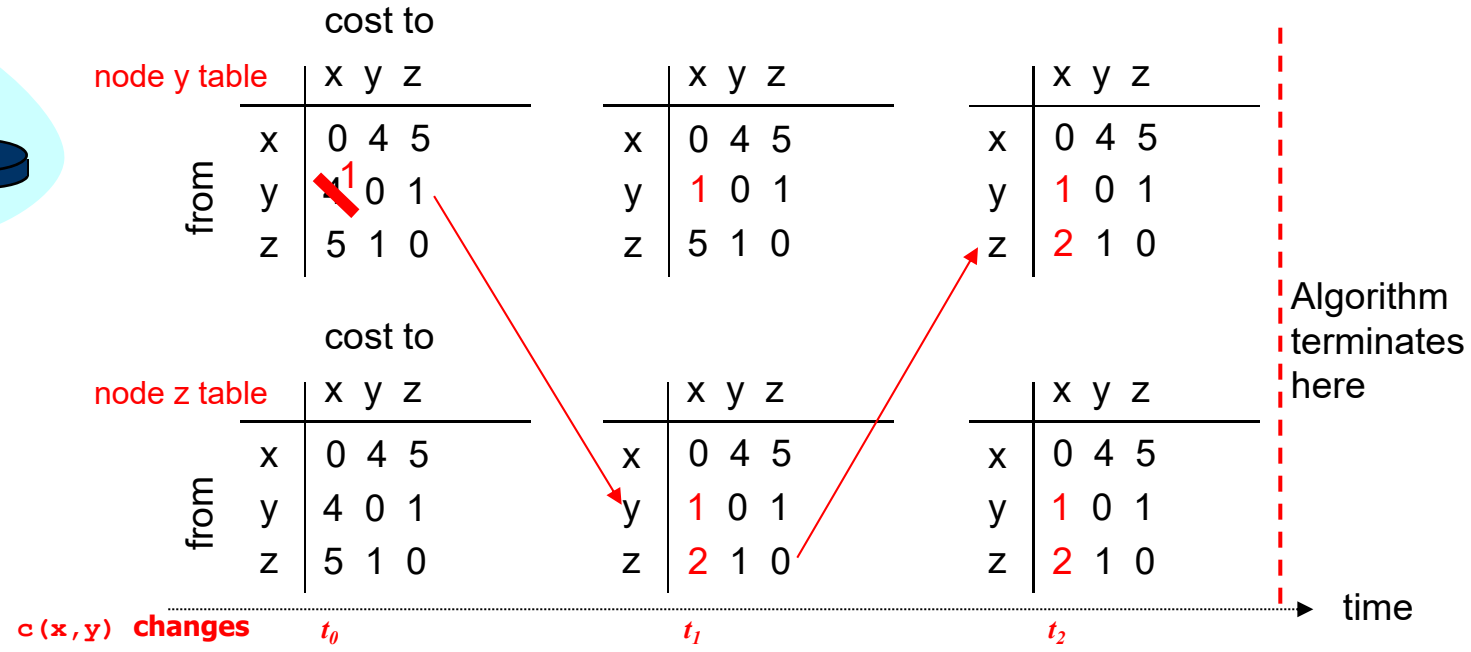
- Link cost changes
  - Node detects local link cost change
  - Updates distance table
  - If cost change in least-cost path, notify neighbors

At t_0, *y* detects the link-cost change, updates its DV, and informs its neighbors

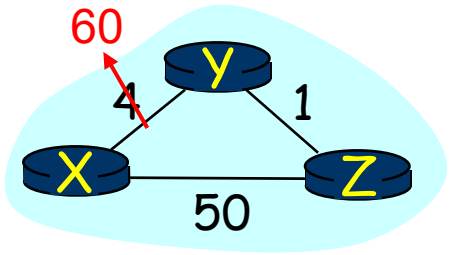
At t_1, *z* receives the update from *y* and updates its table. It computes a new least cost to *x* and sends its neighbors its DV

At t_2, *y* receives *z*'s update and updates its distance table. *y*'s least costs do not change and hence *y* does *not* send any message to *z*





Distance Vector: Link Cost Changes (2)



Advantage	Disadvantage
DV travels fast	Count to infinity problem

d(x,y)= min{c(x,y)+c(x,x), c(x,z)+c(z,y)}
= min{60+0, 5+1} = 6

d(x,z)= min{c(z,x)+c(z,z), c(x,y)+c(y,z)}
= min{50+0, 6+1} = 7

d(y,x)= min{c(y,x)+c(x,x), c(y,z)+c(z,x)}
= min{60+0, 1+7} = 8

d(z,x)= min{c(z,x)+c(x,x), c(z,y)+c(y,x)}
= min{50+0, 1+6} = 7

D(z,x)= min{c(z,x)+c(x,x), c(z,y)+c(y,x)}
= min{50+0, 1+8} = 9

node x table

from	t_0 x	y	z	t_1 x	y	z	t_2 x	y	z	t_3 x	y	z	t_4 x	y	z
x	0	6	5	0	6	5	0	6	7	0	6	7	0	6	9
y	4	0	1	6	0	1	6	0	1	8	0	1	8	0	1
z	5	1	0	5	1	0	7	1	0	7	1	0	9	1	0

node y table

from	t_0 x	y	z	t_1 x	y	z	t_2 x	y	z	t_3 x	y	z	t_4 x	y	z
x	0	4	5	0	6	5	0	6	5	0	6	7	0	6	7
y	6	0	1	6	0	1	8	0	1	8	0	1	10	0	1
z	5	1	0	5	1	0	7	1	0	7	1	0	9	1	0

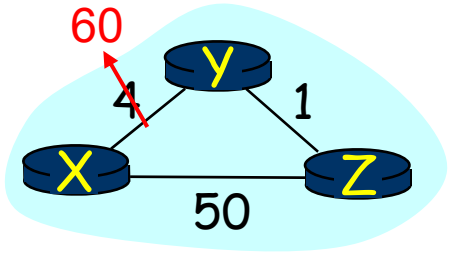
node z table

from	t_0 x	y	z	t_1 x	y	z	t_2 x	y	z	t_3 x	y	z	t_4 x	y	z
x	0	4	5	0	6	5	0	6	5	0	6	7	0	6	7
y	4	0	1	6	0	1	6	0	1	8	0	1	8	0	1
z	5	1	0	7	1	0	7	1	0	9	1	0	9	1	0

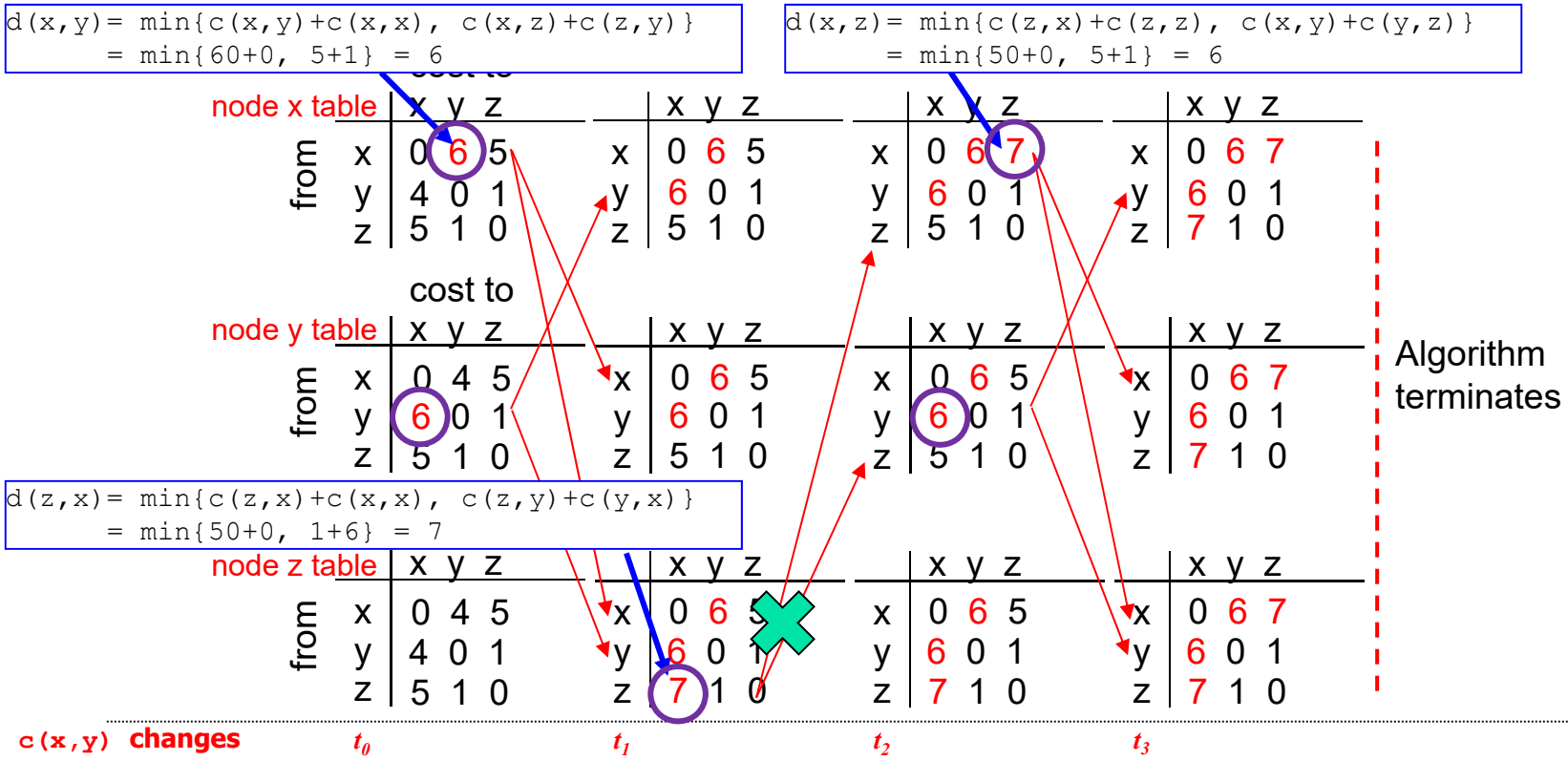
Algorithm continues on!

c(x,y) changes — t_0 t_1 t_2 t_3 t_4 — time

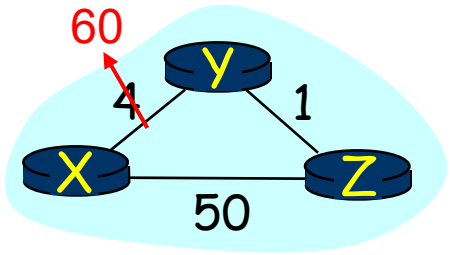
Distance Vector: Split Horizon



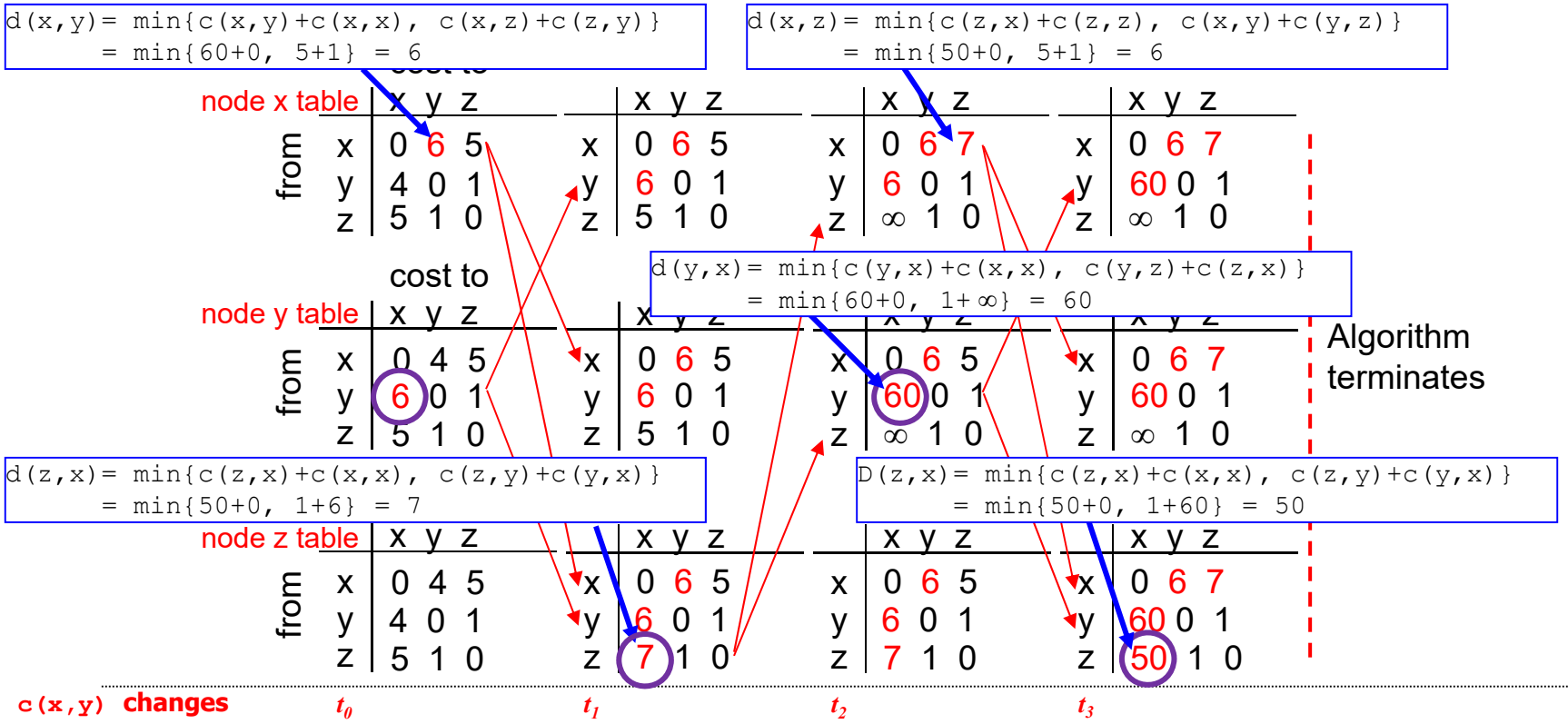
- If Z routes through Y to get to X
  - Z does not advertise its route to X back to Y



Distance Vector: Poison Reverse Failures



- If Z routes through Y to get to X
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
  - Immediate notification of unreachability, rather than split horizon timeout waiting for advertisement
  - Will this completely solve count to infinity problem? Answer is YES

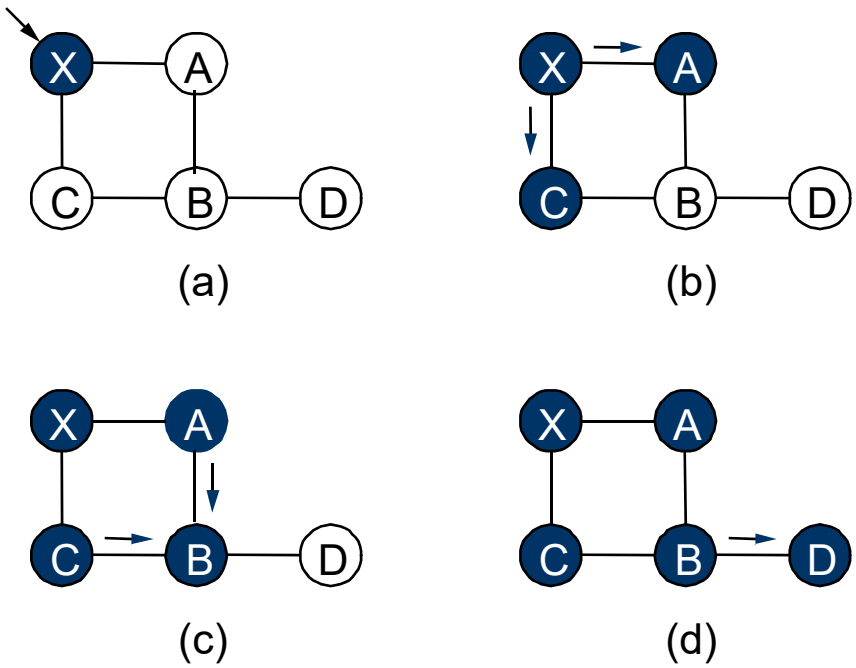


Link State Method

- Every node gets complete copy of graph
  - Every node “floods” network with data about its outgoing links
- Every node computes routes to every other node
  - Using single-source, shortest-path algorithm
- Process performed whenever needed
  - When connections die / reappear

Sending Link State by Flooding

- X wants to send information
  - Sends on all links
- When A or C receives information from X
  - Send on all links other than X

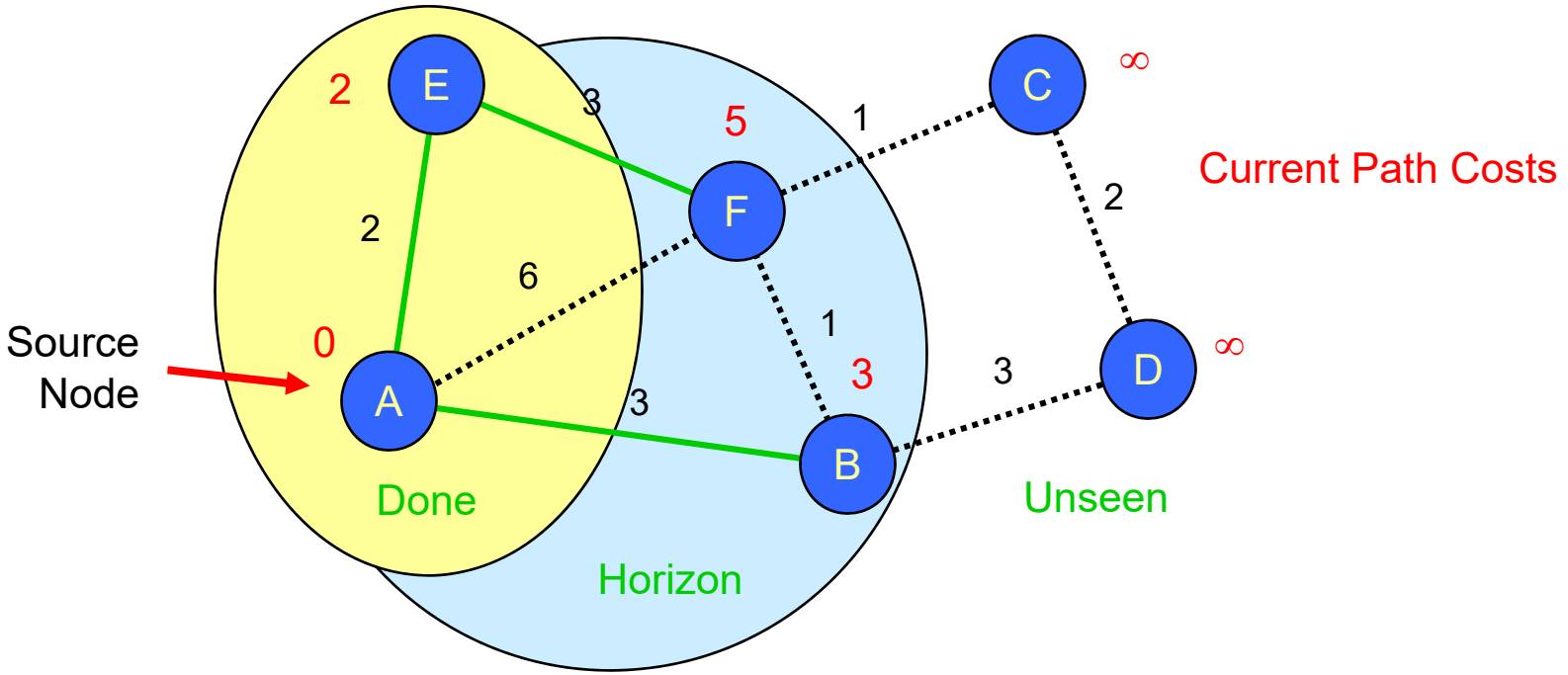


Dijkstra's Algorithm

- Given
  - Graph with source node s and edge costs `c(u,v)`
  - Determine least cost path from s to every node `v`
- Shortest path first algorithm
  - Traverse graph in order of least cost from source

```
1   Initialization:
2     N = {A}
3     for all nodes v
4       if v adjacent to A
5         then D(v) = c(A,v)
6       else D(v) = infinity
7   Loop
8     find w not in N such that D(w) is a minimum
9     add w to N
10    update D(v) for all v adjacent to w and not in N:
11      D(v) = min( D(v), D(w) + c(w,v) )
12    /* new cost to v is either old cost to v or known
13     shortest path cost to w plus cost from w to v */
14  until all nodes in N
```

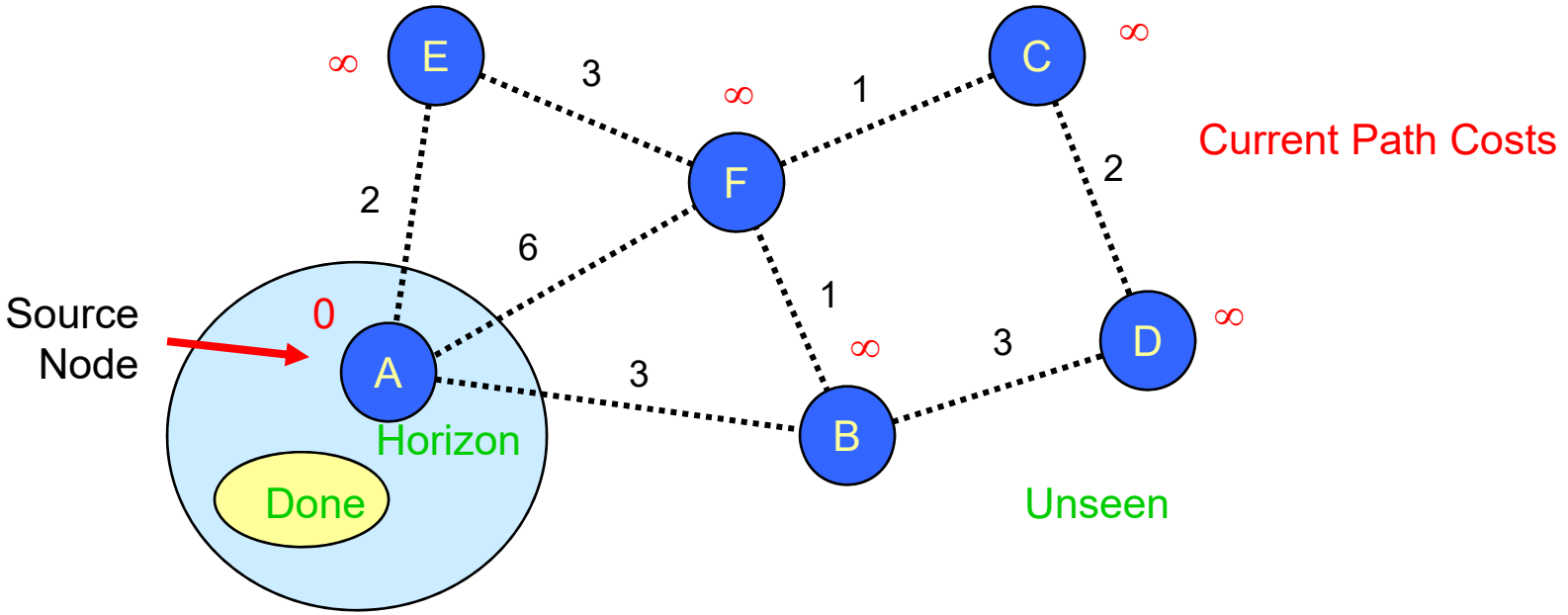
Dijkstra's Algorithm: Concept



- Node Sets
  - Done: Already have least cost path to it
  - Horizon: Reachable in 1 hop from node in Done
  - Unseen: Cannot reach directly from node in Done
  - Label: d(v) = path cost from s to v
  - Path: Keep track of last link in path

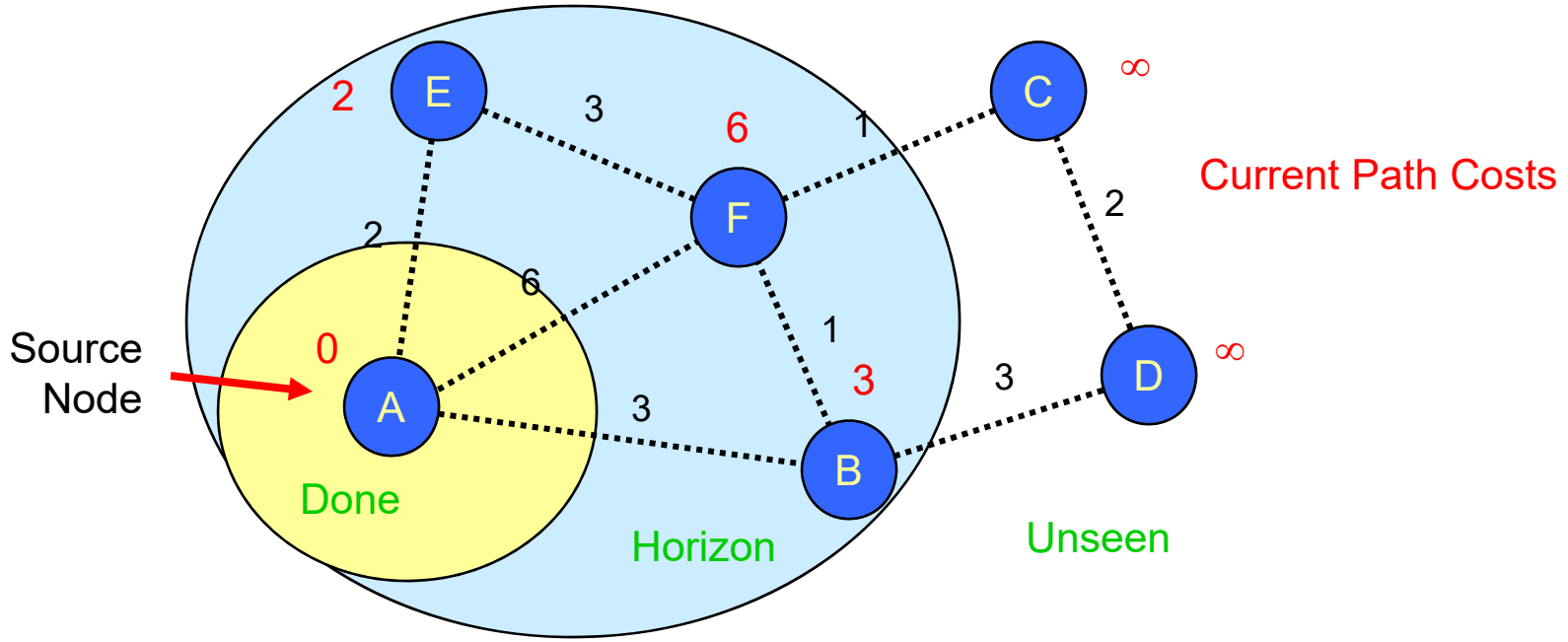
Dijkstra's Algorithm: (1/6)

- No nodes done
- Source in horizon



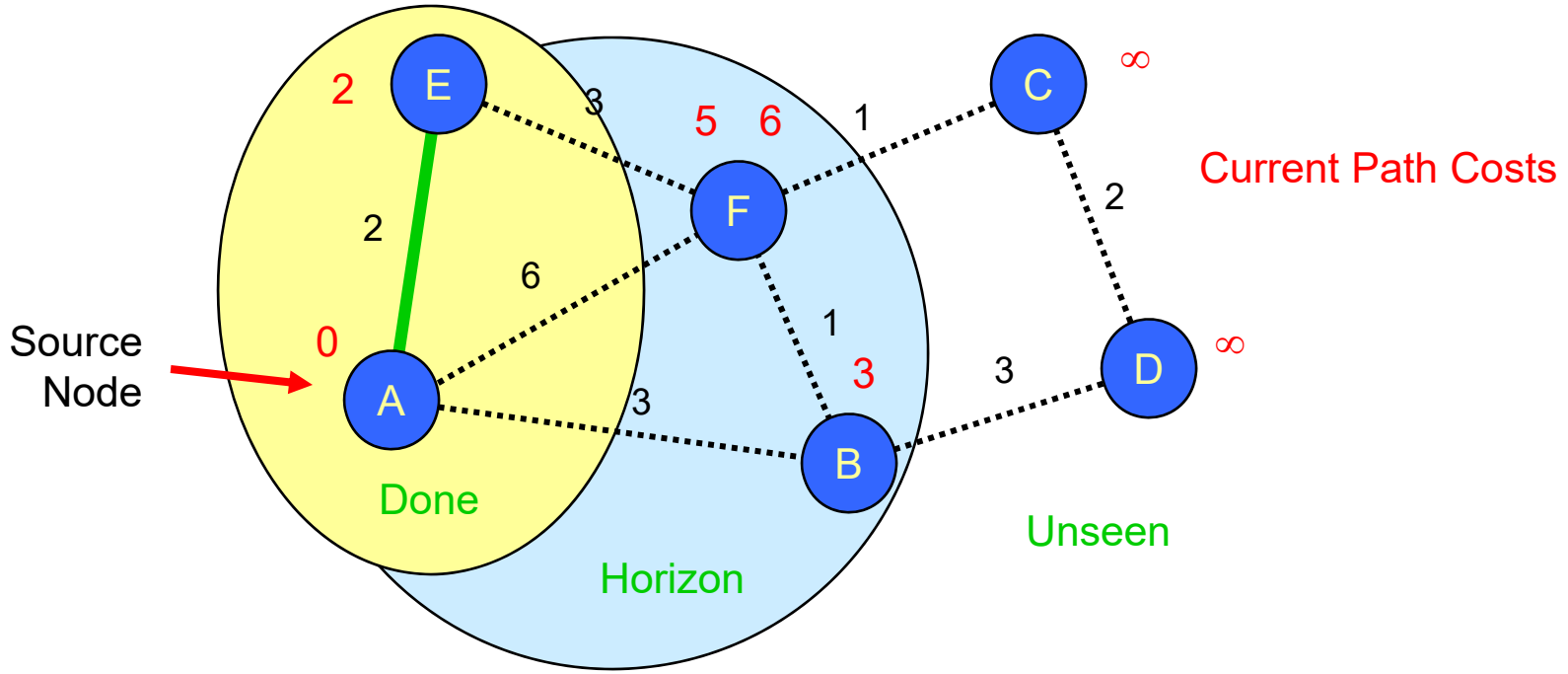
Dijkstra's Algorithm: (2/6)

- d(v) to node A shown in red
  - Only consider links from done nodes



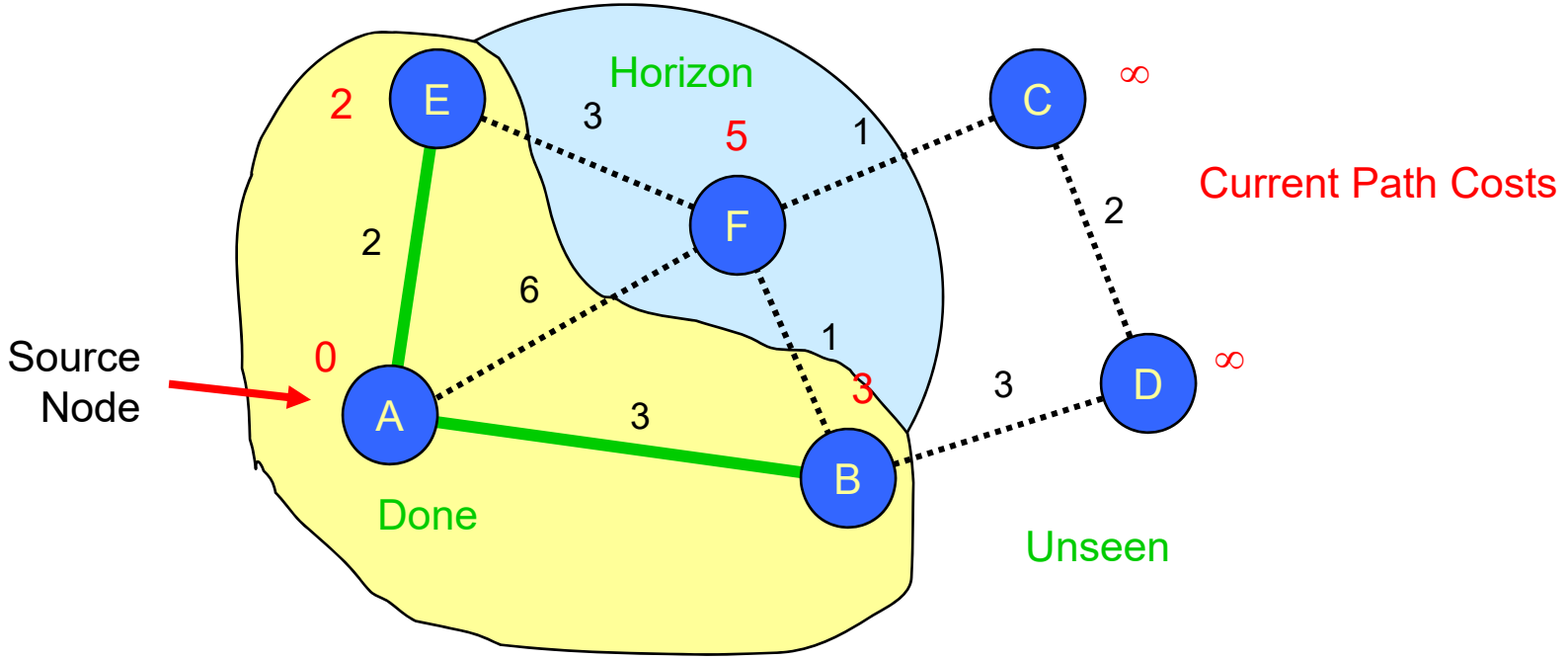
Dijkstra's Algorithm: (3/6)

- Select node v in horizon with minimum d(v)
- Add link used to add node to shortest path tree
- Update d(v) information



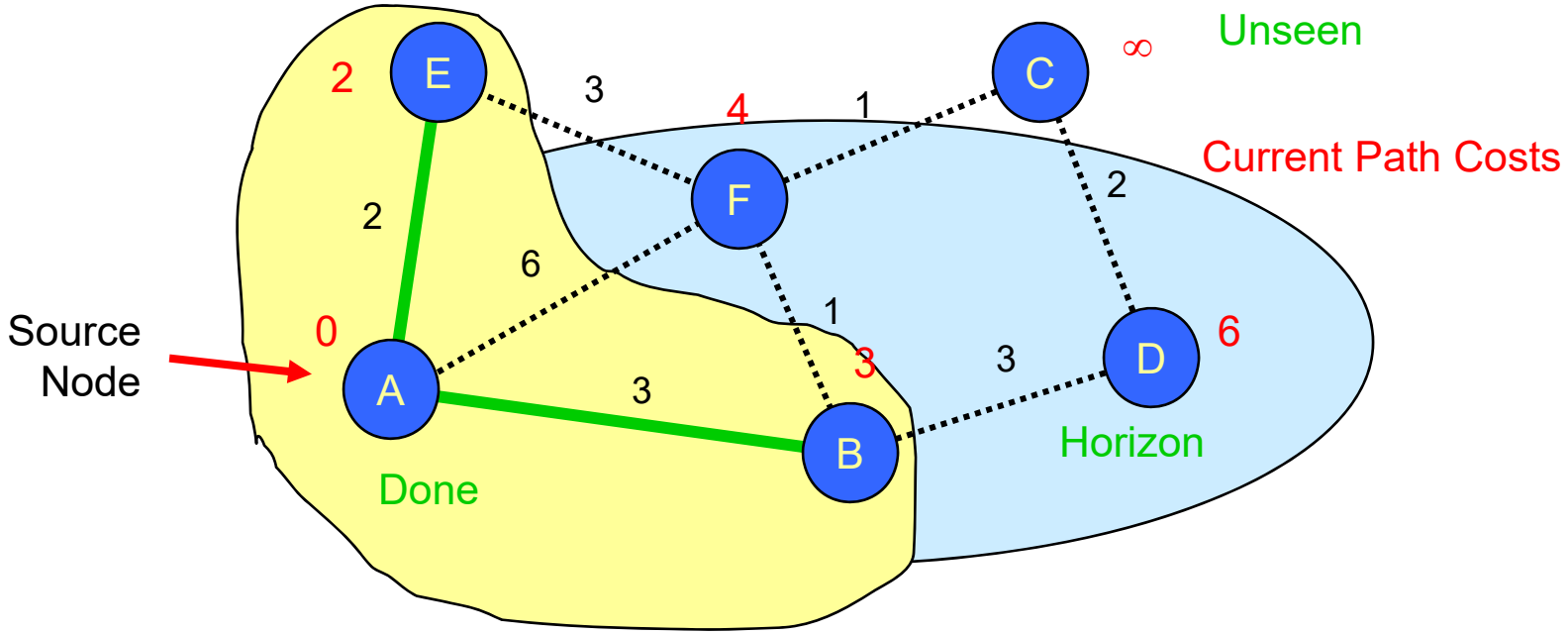
Dijkstra's Algorithm: (4/6)

- Repeat …



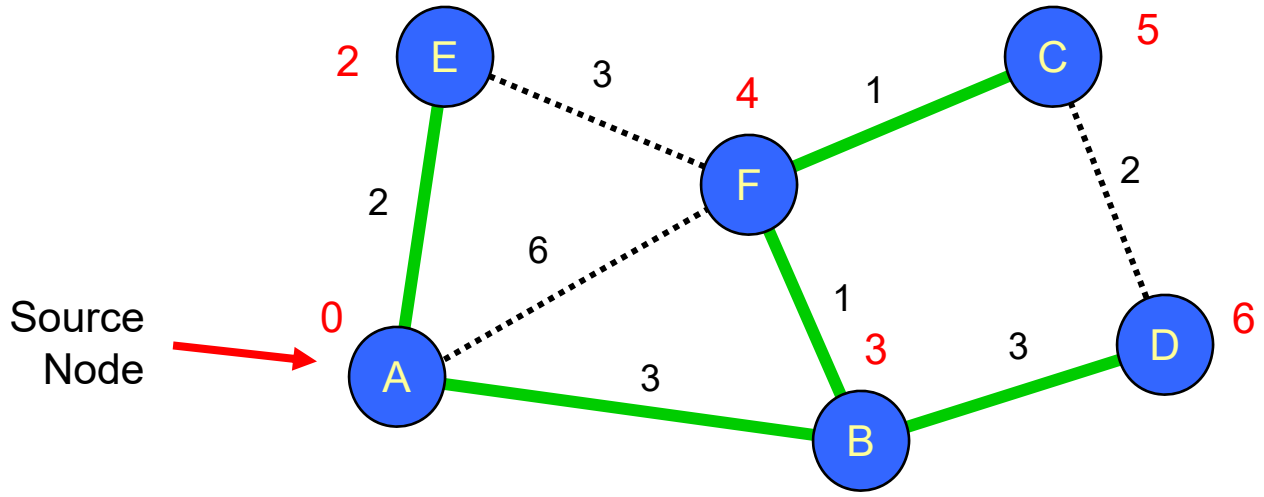
Dijkstra's Algorithm: (5/6)

- Update d(v) values
  - Can cause addition of new nodes to horizon



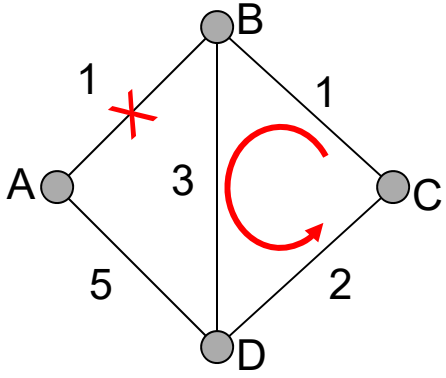
Dijkstra's Algorithm: (6/6)

- Final tree shown in green color



Link State Characteristics

- With consistent Link State Database (LSDB), all nodes compute consistent loop-free paths
- LS can still have transient loops



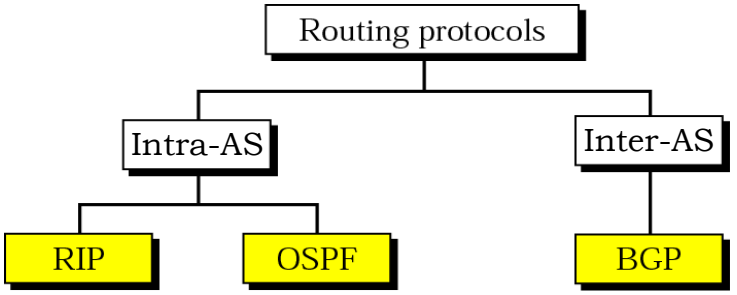
Packet from C→A
may loop around BDC
if B knows about failure
and C & D do not

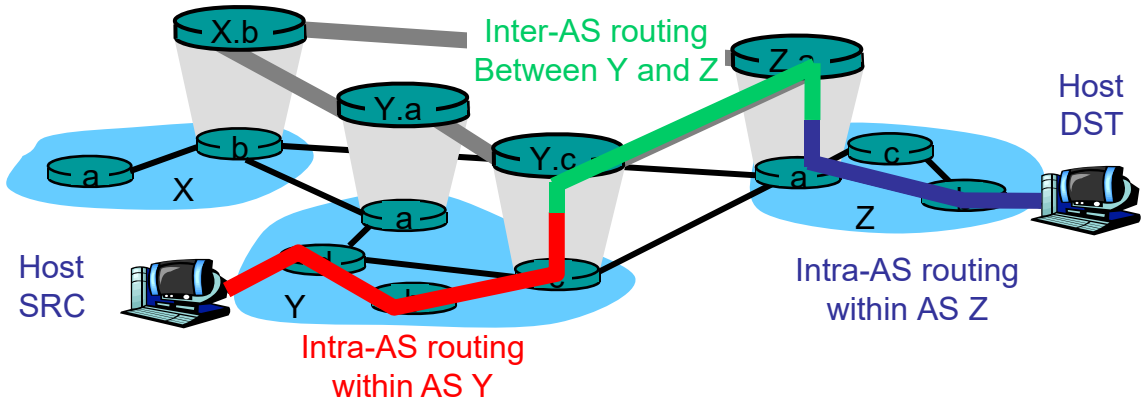
Comparison of DV and LS Algorithms

Algorithm	Distance-Vector	Link-State
Message complexity	Exchange between neighbors only	With n nodes, E links, O(nE) messages
Speed of Convergence	Convergence time varies • May have routing loops • Count-to-infinity problem • Faster with triggered updates	Relatively fast • Complex computation, but can forward before computation • May have transient loops
Space requirements	DV maintains only neighbor state	LS maintains entire topology
Robustness: router malfunctions	Node can advertise incorrect path cost • Each node's table used by others (error propagates)	Node can advertise incorrect link cost • Each node computes its own table

Routing in the Internet

- Global Internet: A hierarchy of Autonomous Systems (ASs) (enterprise ASs interconnected through ISP's ASs)
- Hierarchical routing
  - Intra-AS Routing (within network)
    - RIP (Routing Information Protocol)
    - OSPF (Open Shortest Path First)
  - Inter-AS Routing (between network)
    - BGP (Border Gateway Protocol)





Why Intra- and Inter-AS routing different?

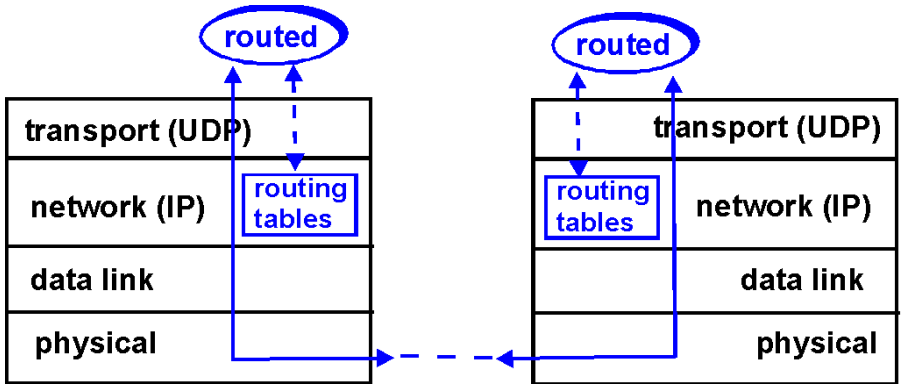
- Policy
  - Intra: under same administrative control, so, policy-based routing is less important
  - Inter: concerned with policies (e.g., which provider to select/avoid)
- Scalability
  - Intra: scalability is less of a concern within an AS. A large AS can be divided into two ASs, e.g., “areas” in OSPF
  - Inter: ability of routing algorithm and table to scale for routing among large numbers of networks
- Performance
  - Intra: focused on performance metrics; needs to keep costs low
  - Inter: routing is policy-oriented; quality of routes is secondary. Also, it is difficult to propagate performance metrics efficiently (latency, privacy, etc)
- We need BOTH!

Routing Information Protocol (RIP)

- Like distance vector algorithm
  - Cost metric is hop count (assume each link costs exactly 1), which is limited to 16
  - Updates as for distance vector algorithm. (Easier, because 1 is added for every link)
  - Messages exchanged every 30 seconds; "Advertisements" – these contain entire routing tables! (dest, nexthop, hopcount)
  - Link failure and recovery method: If a neighbor is silent for 180 seconds, its column is modified (to infinity) and the table is advertised

- Advertisements encapsulated in UDP packets (reliability not required; advertisements are periodically repeated)

- RIP routing tables managed by an application process called routed (daemon)



RIP Table Example

- To get routing table on Unix/Win, type `netstat -rn`
- E.g., a Router RIP table:

```
Destination     Gateway            Flags Ref   Use      Interface
--------------- ------------------ ----- ----- ------- ---------
127.0.0.1       127.0.0.1          UH    0     26492    lo0
192.168.2.      192.168.2.5        U     2     13       fa0
193.55.114.     193.55.114.6       U     3     58503    le0
192.168.3.      192.168.3.5        U     2     25       qaa0
224.0.0.0       193.55.114.6       U     3     0        le0
default         193.55.114.129     UG    0     143454
```

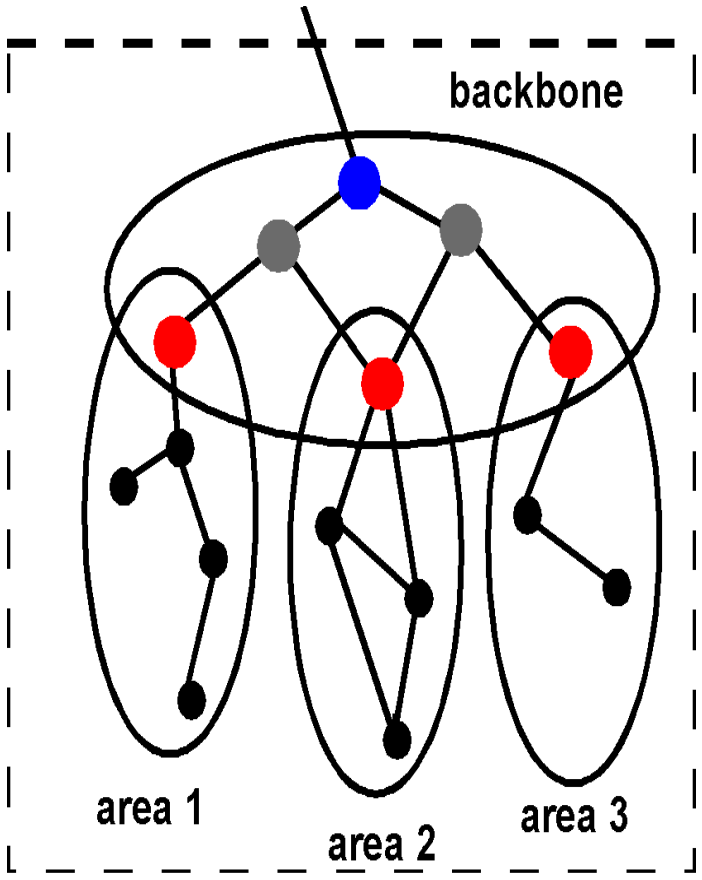
  - 3 attached class C networks (LANs) via fa0, le0 and qaa0
  - Router only knows routes to attached LANs
  - Default is the default gateway “to go up”
  - Route multicast address: 224.0.0.0
  - 1st entry: Loopback interface (for debugging)
  - Learn to use `netstat` by examples:
    http://www.cs.unh.edu/cnrg/lin/linuxProject/resource/netstatCookbook.htm

Open Shortest Path First (OSPF)

- Uses the link state algorithm, i.e.,
  - Link state packet dissemination for every 30 minutes (or more often);
  - Topology map at each node;
  - Route computation using the Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbor router (gives link state)
- Advertisements disseminated to the ENTIRE AS (via flooding) in IP packets
- Advanced features (not in RIP)
  - Security. All OSPF messages are authenticated (to prevent malicious intrusion); TCP connections used
  - Multiple same-cost paths allowed (only one path must be chosen to carry all traffic in RIP)
  - Multiple cost metrics for different TOS for each link (e.g., satellite link cost set "low" for best effort; high for real time)
  - Integrated uni- and multicast support. Multicast OSPF (MOSPF) uses the same topology database as OSPF
  - Hierarchical OSPF in single AS (large routing domain)

Hierarchical OSPF

- OSPF AS: 2 level hierarchy (local area and backbone)
  - Link state advertisements do not leave respective areas
  - Nodes in each area have detailed area topology; they only know direction (shortest path) to networks in other areas
- "Internal routers" perform intra-AS routing only
- "Area Border routers" route packets to other areas
- "Backbone routers" run an OSPF routing algorithm limited to the backbone
- "Boundary routers" connect to other ASs



Border Gateway Protocol (BGP)

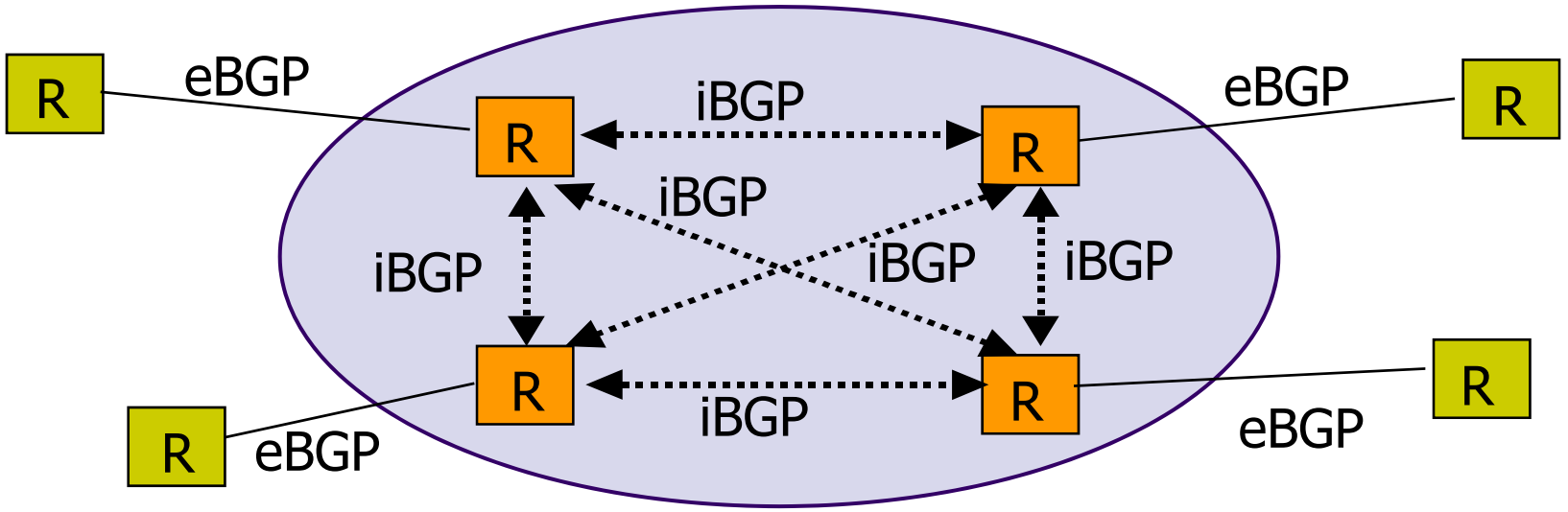
- BGP (de facto standard) is path vector protocol: advertises a sequence of AS numbers to the destination network
- Each Border Gateway (BG) broadcasts to neighbors (peers) the entire path (i.e., sequence of AS's) to destination
- E.g., BG_X may store the following path to destination BG_Z
  - Path (X,Z) = X,Y1,Y2,Y3,…,Z
- Now, suppose BG_X sends its path (X,Y1,Y2,Y3,…,Z) to peer BG_Y
- BG_Y may or may not select the path offered by BG_X, because of cost, policy or loop prevention reasons
- If BG_Y selects the path advertised by BG_X, then
  - Path (Y,Z) = Y, Path (X,Z)

Note: path selection based not so much on cost (e.g., no. of AS hops), but mostly on administrative and policy issues (e.g., do not route packets through competitor's AS)

BGP (cont.)

- Peers exchange BGP messages using TCP (port 179)
- Initially, BGP peers exchange entire BGP routing table
  - Incremental updates sent subsequently
  - Reduces bandwidth usage and processing overhead
  - Keep alive messages sent periodically (30 seconds)
- BGP defines 4 types of messages
  - OPEN : opens a TCP connection to peer and authenticates sender
  - UPDATE : advertises new path (or withdraws old)
  - KEEPALIVE : keeps connection alive in absence of UPDATES; also serves as ACK to an OPEN request
  - NOTIFICATION : reports errors in previous message; also used to close a connection
- Internal BGP (iBGP) between BGP routers in same AS
- External BGP (eBGP) connections across AS borders
- IETF RFC1771 at http://www.ietf.org/rfc/rfc1771.txt

iBGP and eBGP



- eBGP to exchange reachability information in different AS's
  - eBGP peers directly connected
- iBGP to ensure net reachability info is consistent among the BGP speakers in the same AS

Directory Service and Naming

- Directory service (name service) maps the names of network resources to their respective network addresses that is a network accessible database with limited functionality
  - Small amount of information in each request/reply
  - Limited functionality (compared to a complete database system)
- Example of a directory service
  - NIS, X.500, LDAP, DNS, etc
- Original UNIX naming system stores info in `/etc`
  - Does not scale well for large network
- Network naming services:
  - Information stored centrally (client-server model)
  - Usernames, passwords, hostnames/IP address, etc
  - Binds names to objects
  - Resolves names to objects. E.g., www.cs.nyu.edu is 128.122.80.245

Directory vs. Database

- Directory is often described as Database
- But directories have special characteristics different from general databases
  - Directories are accessed much more than updated. Hence directories are optimized for read access
  - Directories are not suited for information that changes rapidly (e.g., no. of jobs in a printer queue)
  - Many directories don't support transactions
  - Directories normally limits the type of information that can be stored
  - Databases use powerful query languages like SQL but directories normally use very simple access methods
  - Hence directories can be optimized to economically provide more applications with rapid access

Network Information Service (NIS)

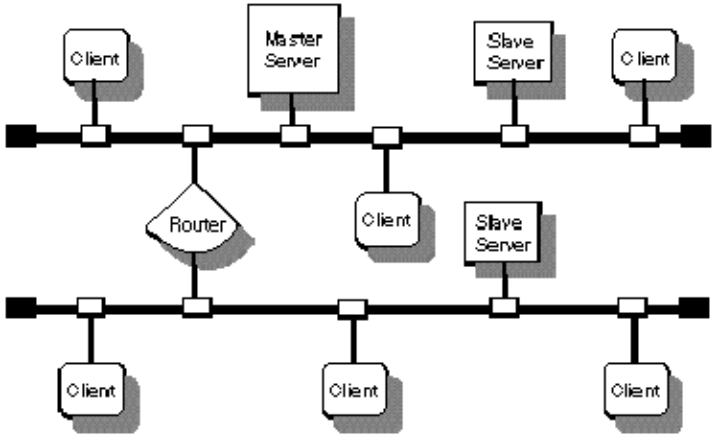
- Developed by Sun Microsystems – originally Yellow Pages (YP)
- NIS provides a mechanism for keeping important files synchronized between hosts on a network
- NIS is a RPC-based client-server system that allows a group of machines within an NIS domain to share a common set of configuration files
- What does NIS do?
  - Allows networked machines to have a common interface regardless of the workstation that you log into
  - Allows you to coordinate the distribution of database information throughout your networked environment
  - Focused on making network administration more manageable by providing centralized control over a variety of network information
- Client-server model. Servers are replicated (can be master server or slave server). NIS+ is similar to NIS, but more features and secure

NIS Advantages and Disadvantages

- Advantage
  - Unnecessary for administrators to be aware of NIS's internal data formats
- Disadvantages
  - Not suitable for managing a large network of machines
  - Consume a fair amount of network bandwidth
  - If a slave server is down or inaccessible when a map is changed, the slave's copy will not be updated
  - Not secure

NIS Environment

- NIS uses a client-server arrangement. Systems can have the following roles
  - Master server – A system that stores the master copy of the NIS database files, or maps. Only the master maps can be modified, whereas slave servers provide read-only access. Each domain can have only one master server
  - Slave server – A system that obtains and stores copies of the master server's NIS maps. Each domain can have multiple slave servers distributed throughout the network
  - Client – Any system that queries NIS servers for NIS database information. Clients do not store and maintain copies of the NIS maps for their domain locally



NIS Maps

- Information distributed by NIS is stored in database files called maps
- Most of the NIS maps represent files that were traditionally stored in the `/etc` directory, include the following:
  - `aliases`
  - `group`
  - `hosts`
  - `netgroups`
  - `networks`
  - `passwd`
  - `protocols`
  - `rpc`
  - `services`
- Each NIS map contains a set of keys and associated values
- Each NIS map has a map name, used by programs to access data in the map

NIS Domains

- Named set of NIS maps is called a domain
- Domain name is required for retrieving data from an NIS database. It is set at the time the system is booted
- NIS domain is an administrative entity that consists of a master server, one or more slave servers, and numerous clients
- All systems in a domain share the same set of NIS database files
- NIS uses domains to arrange the workstations, users, and networks in its namespace

LDAP

- LDAP stands for Lightweight Directory Access Protocol
- Specialized database optimized for reading and searching
- What can be stored in LDAP
  - Passwords, phone numbers, date-of-birth, JPEG photos
- Client-server model (again)
- LDAP directory service is global
- OpenLDAP is an open source implementation

LDAP Advantage and Disadvantage

- LDAP is well suited for
  - Information that is referenced by many entities and applications
  - Information that needs to be accessed from more than one location
    - Roaming, e.g. by “Road Warriors”
    - Preference information for web “portals”
  - Information that is read more often than it is written
- LDAP is not well suited for
  - Information that changes often (it is not a relational database)
  - Information that is unstructured (it is not a file system)

LDAP Information Model

- LDAP entry is a collection of attributes with a unique Distinguished Name (DN)

```
uid=jane,ou=People,dc=cims,dc=
nyu,dc=edu
```

- Each attribute has a type and one or more values

```
telephoneNumber: 212-995-1234
```

- Values of the `objectClass` attributes decide what attributes are required/allowed

```
objectClass: posixAccount
```

- `objectClasses` are defined in schema

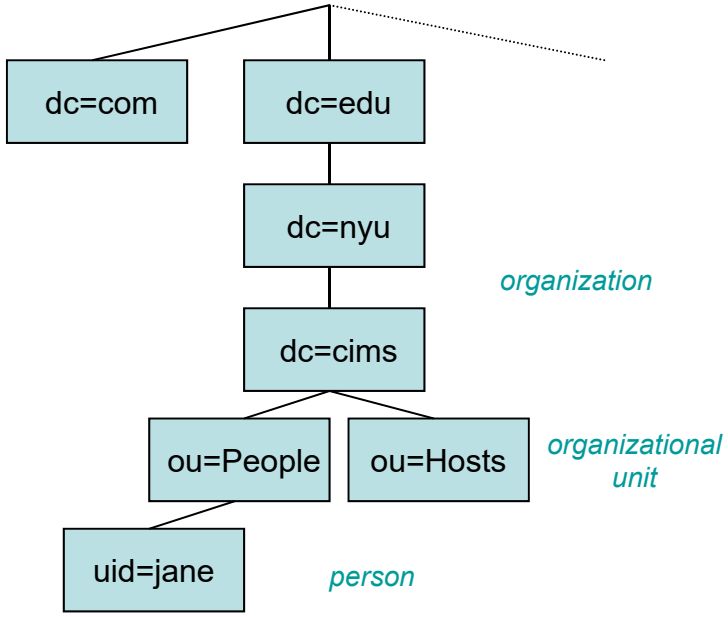


Figure: Entries are arranged in a hierarchical structure

Accessing LDAP

- Add, modify, and delete entries with `ldapadd`, `ldapmodify`, and `ldapdelete`
- Search the LDAP database with `ldapsearch`
  - Bind as some DN or anonymously

```
ldapsearch -D "cn=Directory Manager" -h ldaphost
-b "dc=cims,dc=nyu,dc=edu" "uidNumber=9876" gecos
```

- Access to information is controlled by an access control list, e.g., password hashes are not available through anonymous bind

Domain Name System (DNS)

- DNS provides mapping between host name & IP address
- Humans prefer to use computer names instead of IP addresses
- E.g.,: www.yahoo.com instead of 204.71.200.68
- One host name may map to multiple IP addresses. Why? One host will have multiple IP addresses if it has multiple network interfaces
- Before DNS the mappings name to IP address where stored in a file `/etc/hosts`
- Network administrators used to exchange updates in the `/etc/hosts` file
  - This solution is not scalable

DNS (cont.)

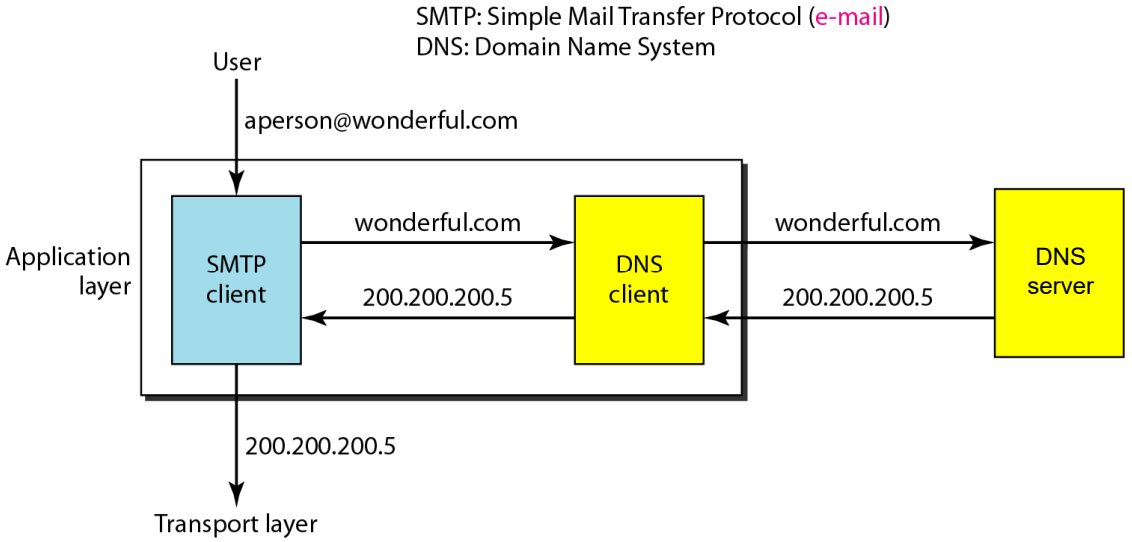


Figure: Example of using the DNS service

- DNS is a service that translates host names to IP addresses
- DNS uses a distributed lookup algorithm and contacts as many servers as necessary
- Host names are divided in domains: host.dom2.dom1.dom0
- E.g.,: ector.cs.purdue.edu with the most general domain at the right

ICANN

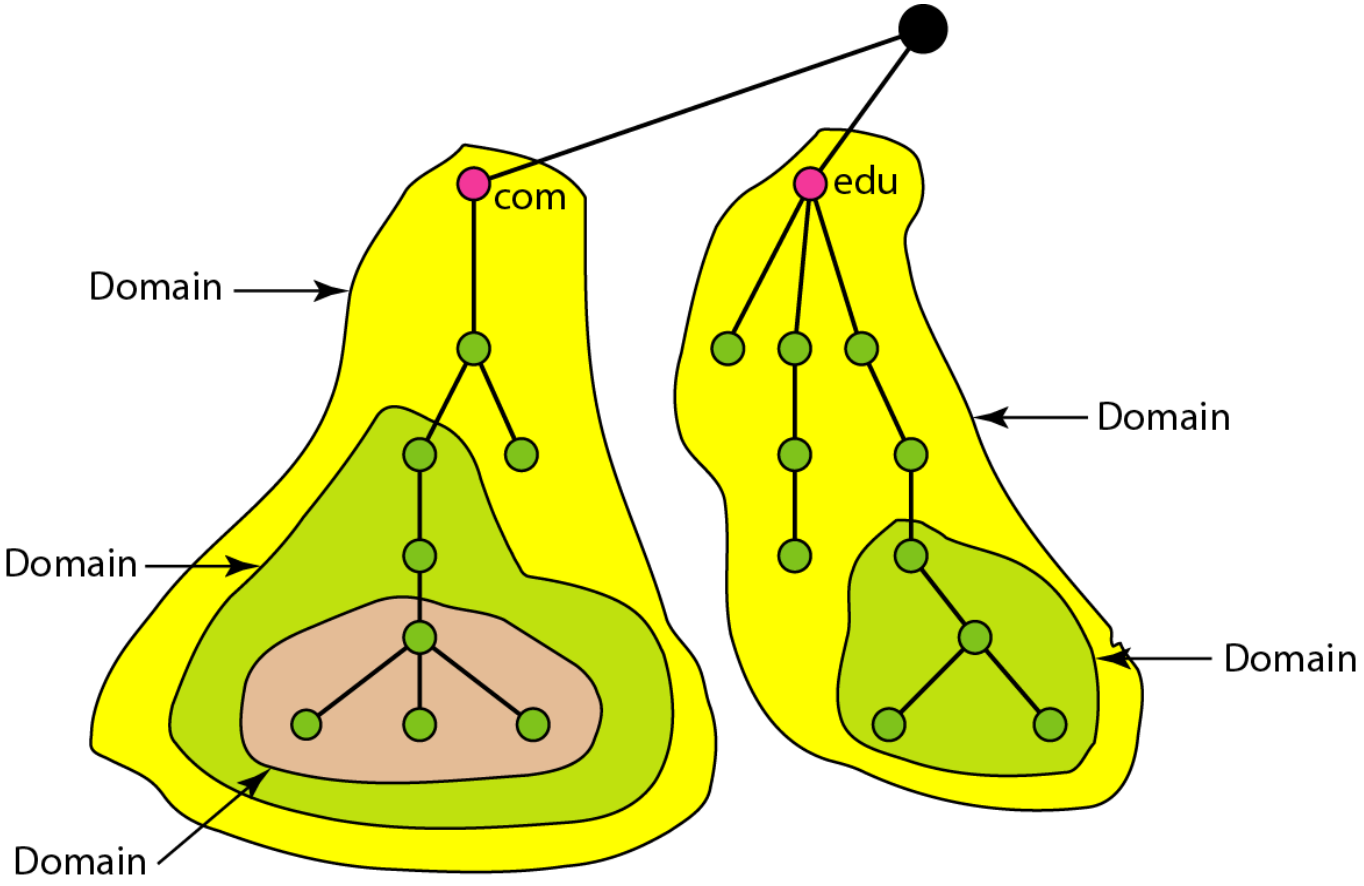
- ICANN stands for Internet Corporation for Assigned Names and Numbers
- Before ICANN, Jon Postel was the *The Central Authority* in charged of assigning domain names and port numbers as well as editing RFCs (Request for Comments)
- Once a register desires a domain name, the central authority (ICANN) registered the domain name in a top-level domain (names are controlled by international copyright/trademark laws)

Top Level Domain

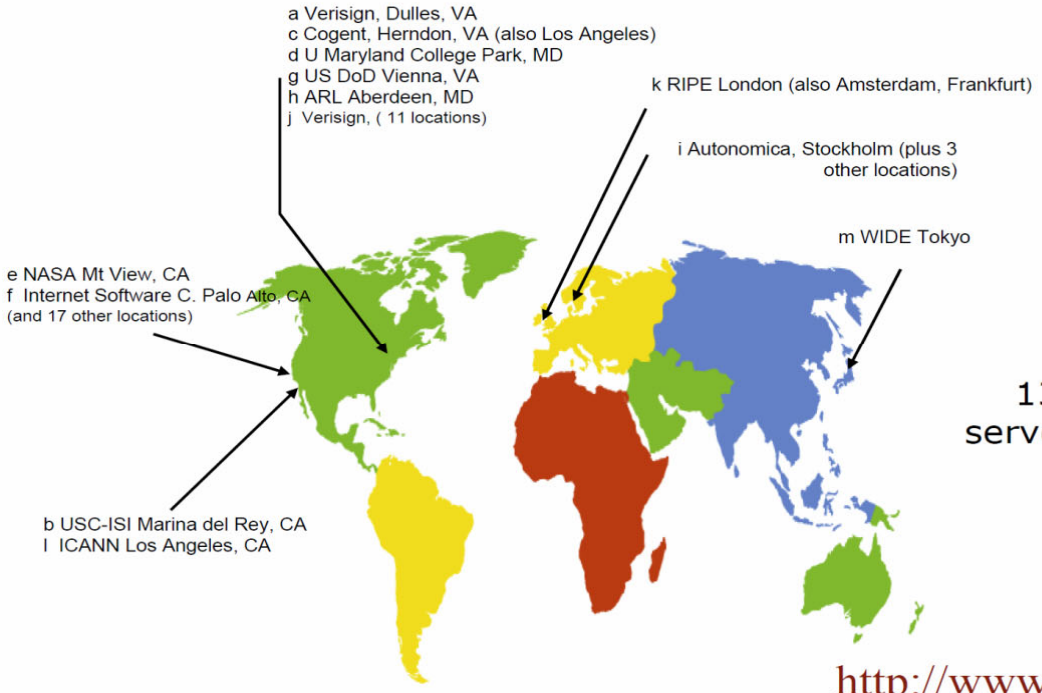
Domain Name	Description
.com	Commercial Organization
.edu	Educational Institution
.gov	Government organization
.mil	Military Group
.net	Major network support center
.org	Organization
.arpa	ARPA domain
.<Country code>	A country
.aero	Air-Transport Industry
.biz	Business
.coop	Non-profit cooperatives
.info	Unrestricted
.museum	Museums
.name	Individuals
.pro	professionals

New

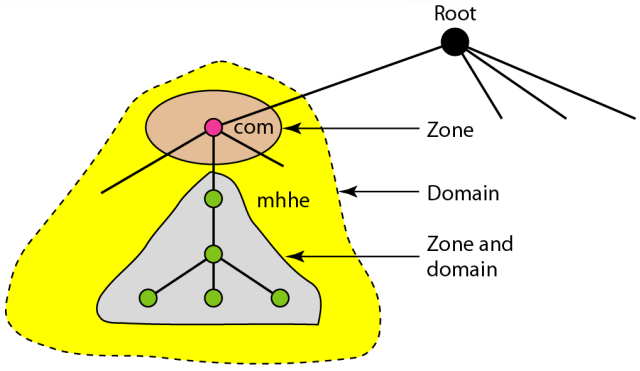
Domain Names



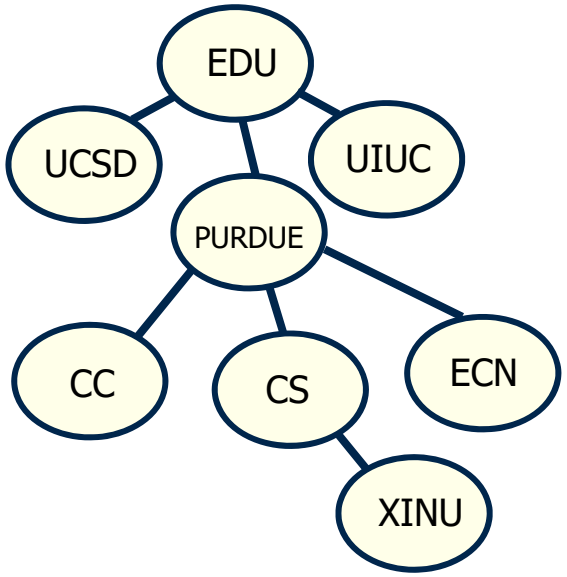
DNS Root Servers



- Responsible for 'root' zone
- Approx. dozen root name servers worldwide
  - Currently {a-m}.root-servers.net
- Local name servers contact root servers when they cannot resolve a name
  - Configured with well-known root servers



DNS Hierarchy

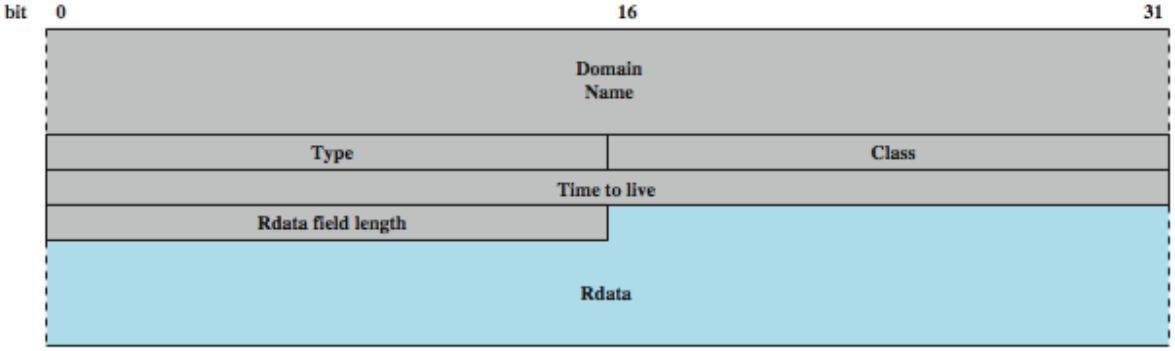


Individual groups within the organization can choose a hierarchical structure that is appropriate to the group

- Each node in hierarchy stores a list of names that end with same suffix
  - Suffix = path up tree
- E.g.,: ector.cs.purdue.edu
  - purdue.edu was assigned by a central authority
  - cs was assigned by the Purdue authority
  - ector was assigned by the network admin at the Computer Science Department
- Domain names correspond to a naming hierarchy

DNS Database

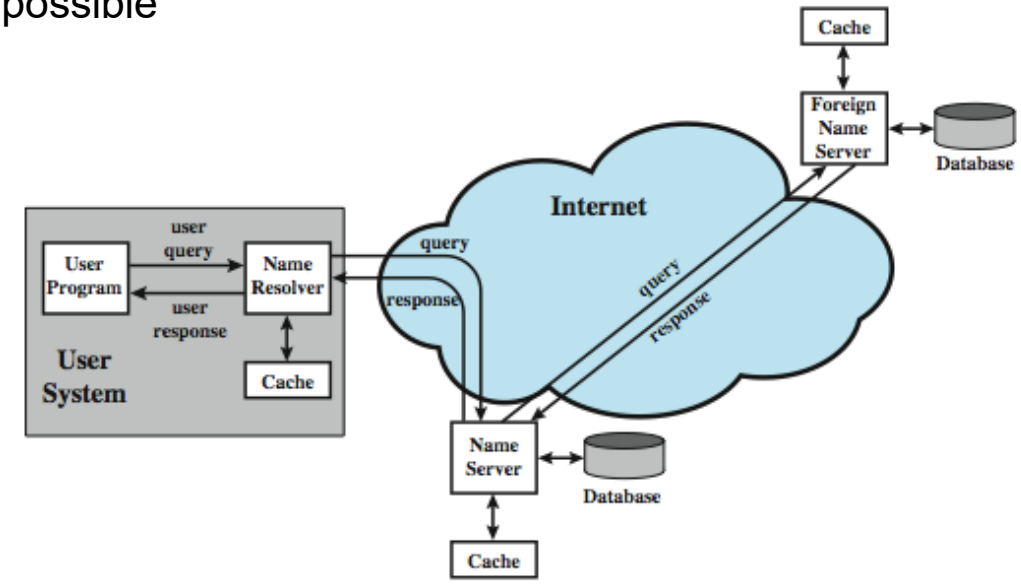
- Hierarchical database
- Containing resource records (RRs)



- Features
  - Variable-depth hierarchy for names
  - Distributed database
  - Distribution controlled by database
- Provides name-to-address directory service for network applications

DNS Operation

- DNS operation typically includes the following steps:
  - ① A user program requests an IP address for a domain name
  - ② A resolver module in the local host or local ISP formulates a query for a local name server in the same domain as the resolver
  - ③ The local name server checks to see if the name is in its local database or cache, and, if so, returns the IP address to the requestor. Otherwise, the name server queries other available name servers, starting down from the root of the DNS tree or as high up the tree as possible
  - ④ When a response is received at the local name server, it stores the name/address mapping in its local cache and may maintain this entry for the amount of time specified in the time to live field of the retrieved RR
  - ⑤ The user program is given the IP address or an error message Variable-depth hierarchy for names



DNS Client/Server

- All domains are linked together to form a unified organization
- DNS uses backup servers
- ISPs offer DNS to subscribers
- Default DNS server is configured by the administrator
- In some networks the default name server is returned by DHCP

Servers/Resolvers

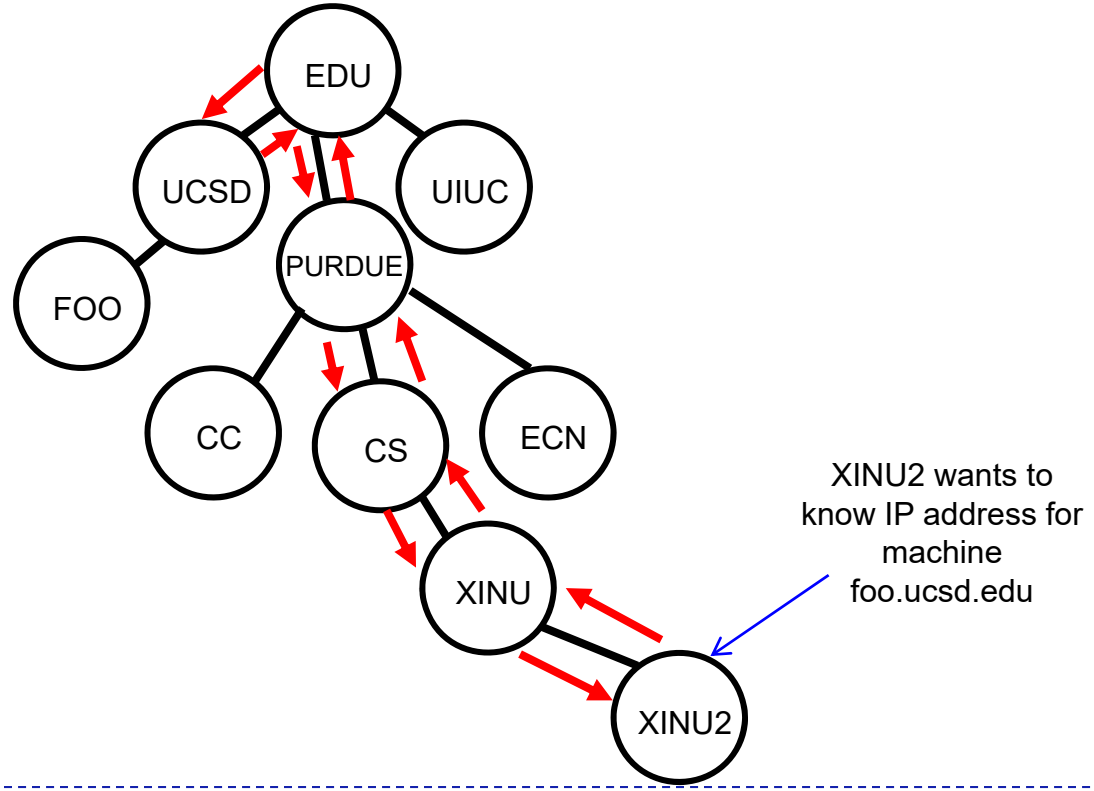
- Each host has a resolver
  - Typically a library that applications can link to
  - Local name servers hand-configured (e.g., `/etc/resolv.conf`)
- Name servers
  - Either responsible for some ‘root’ zone or
  - Local servers – do lookup of distant host names for local hosts
  - Typically answer queries about local zone

DNS Lookup

- DNS Lookup can use either TCP or UDP
- Application willing to translate name to IP address becomes the DNS client
- DNS client sends the request to the local DNS server
- If server knows the answer it returns it. Otherwise there are 2 kinds of DNS resolution:
  - Recursive resolution
  - Iterative resolution

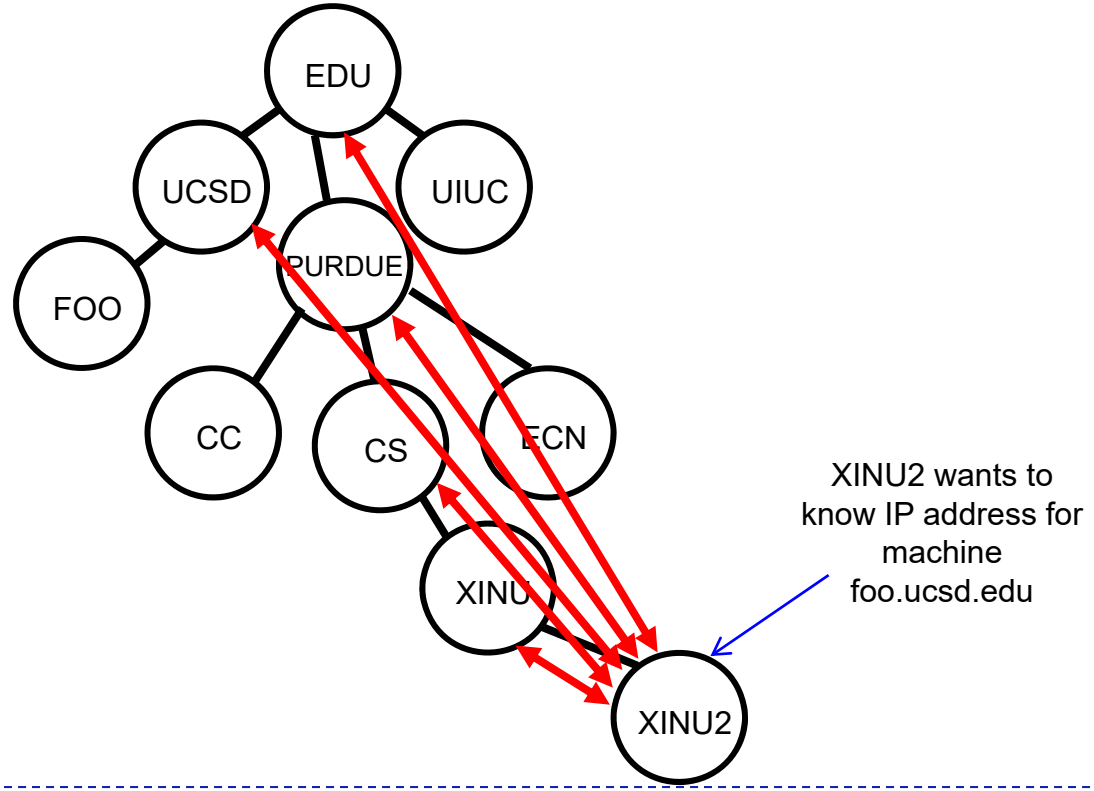
DNS Lookup: Recursive Resolution

- Server contacts a domain server that can resolve the name and returns an answer
- Server may ask another DNS server and so on



DNS Lookup: Iterative Resolution

- In DNS Lookup Iterative Resolution, if a server cannot find the answer, it replies with the IP address of a server that the client should contact next to find the answer



DNS Lookup (cont.)

- DNS Lookup process is also called name resolution
- A client must know how to contact one DNS server
- IP address of the default DNS server is configured by network administrator or by DHCP
- DNS server should know the address of one root server and may know the address of the server immediately above
- Parent name servers know the address of their children
- There is a high cost for queries of non local hosts. This queries in some cases have to go to the root server

DNS Caching

- Caching is the key to make an efficiency in DNS resolution
- Without caching, DNS resolution would be a large component of the Internet traffic
- Both DNS Clients and Servers do name caching
- How long should the cache persist?
  - When a server replies, the response include a TTL field
- Changes in the domain table may take hours to propagate

DNS Types

- DNS Server stores a type with each entry
- When a resolver looks up a name, the resolver must specify the type that is desired
- DNS server will return the IP address that is associated to that name and that type
- DNS Types:
  - Type A – Address
    The value is the IP address for the named computer
  - Type NS – Name Server
    The value is the name of authoritative name server for this domain. E.g., purdue.edu
  - Type MX – Mail Exchanger
    The value is the IP address of the mail server. E.g., grr@cs.purdue.edu
  - Type CNAME – Computer NAME
    The value is another domain name. It is used to establish an alias. E.g., www.cs.purdue.edu is an alias of lucan.cs.purdue.edu

Using nslookup: DNS Client Program

```
C:¥>nslookup
Default Server: ns.jaist.ac.jp
Address:  150.65.1.1

> set querytype=A
> yahoo.com
Server:  ns.jaist.ac.jp
Address:  150.65.1.1

Non-authoritative answer:
Name:    yahoo.com
Addresses:  72.30.2.43
          98.137.149.56
          209.191.122.70
          67.195.160.76
          69.147.125.65

>
```

Using `nslookup`: DNS Client Program

```
> set querytype=NS
> yahoo.com
Server:  ns.jaist.ac.jp
Address:  150.65.1.1

Non-authoritative answer:
yahoo.com       nameserver = ns1.yahoo.com
yahoo.com       nameserver = ns4.yahoo.com
yahoo.com       nameserver = ns2.yahoo.com
yahoo.com       nameserver = ns8.yahoo.com
yahoo.com       nameserver = ns5.yahoo.com
yahoo.com       nameserver = ns6.yahoo.com
yahoo.com       nameserver = ns3.yahoo.com

ns1.yahoo.com   internet address = 68.180.131.16
ns2.yahoo.com   internet address = 68.142.255.16
ns3.yahoo.com   internet address = 121.101.152.99
ns4.yahoo.com   internet address = 68.142.196.63
ns5.yahoo.com   internet address = 119.160.247.124
ns6.yahoo.com   internet address = 202.43.223.170
ns8.yahoo.com   internet address = 202.165.104.22

>
```

Using `nslookup`: DNS Client Program

```
> set querytype=MX
> yahoo.com
Server:  ns.jaist.ac.jp
Address:  150.65.1.1

Non-authoritative answer:
yahoo.com       MX preference = 1, mail exchanger = a.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = b.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = c.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = d.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = e.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = f.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = g.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = h.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = i.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = j.mx.mail.yahoo.com
yahoo.com       MX preference = 1, mail exchanger = k.mx.mail.yahoo.com

yahoo.com       nameserver = ns4.yahoo.com
yahoo.com       nameserver = ns3.yahoo.com
yahoo.com       nameserver = ns8.yahoo.com
yahoo.com       nameserver = ns2.yahoo.com
yahoo.com       nameserver = ns5.yahoo.com
yahoo.com       nameserver = ns1.yahoo.com
yahoo.com       nameserver = ns6.yahoo.com
a.mx.mail.yahoo.com     internet address = 67.195.168.31
b.mx.mail.yahoo.com     internet address = 74.6.136.65
c.mx.mail.yahoo.com     internet address = 206.190.54.127
d.mx.mail.yahoo.com     internet address = 209.191.88.254
e.mx.mail.yahoo.com     internet address = 67.195.168.230
f.mx.mail.yahoo.com     internet address = 98.137.54.237
g.mx.mail.yahoo.com     internet address = 98.137.54.238
h.mx.mail.yahoo.com     internet address = 66.94.236.34
ns1.yahoo.com   internet address = 68.180.131.16
```

Using `nslookup`: DNS Client Program

```
> set querytype=CNAME
> yahoo.com
Server:  ns.jaist.ac.jp
Address:  150.65.1.1

yahoo.com
        primary name server = ns1.yahoo.com
        responsible mail addr = hostmaster.yahoo-inc.com
        serial  = 2010092500
        refresh = 3600 (1 hour)
        retry   = 300 (5 mins)
        expire  = 1814400 (21 days)
        default TTL = 600 (10 mins)

>
```

DHCP

- Three protocols
  - RARP (until 1985, no longer used), request IP address
  - BOOTP (1985-1993)
  - DHCP (since 1993). Only DHCP is widely used today
- Dynamic Host Configuration Protocol (DHCP)
  - For assigning IP addresses dynamically and usually temporarily
  - IP addresses are assigned on-demand
  - Avoid manual IP configuration
  - Support mobility of laptops

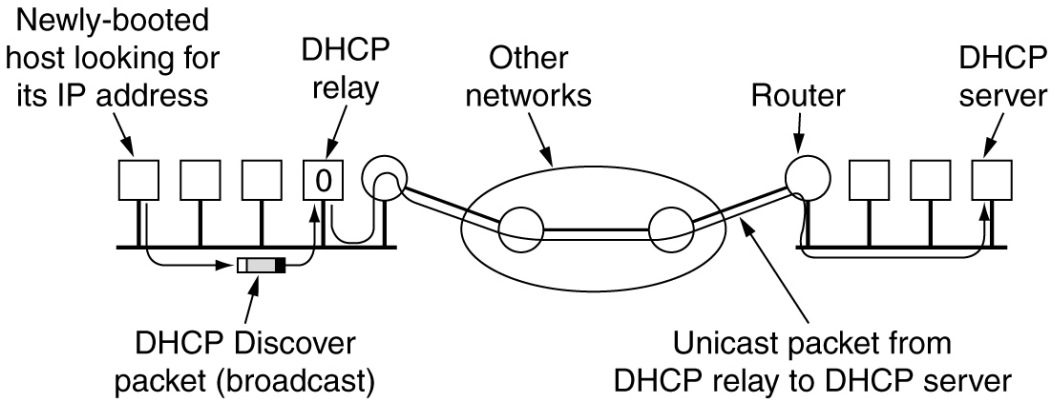


Figure: DHCP operation

DHCP

- Before DHCP, an administrator had to manually configure the following parameters to add a computer to the Internet:
  - Local IP address – current address
  - Subnet mask – used to send packets to hosts in same LAN
  - Default router – used to send packets to hosts outside the LAN
  - Default DNS server – used to convert names to IP addresses
- In UNIX the command used to set these parameters is `ifconfig`
- In Windows is `ipconfig` or Control Panel
- When a machine boots or when the network card is first connected, the OS broadcasts a DHCP request
- DHCP runs on UDP using ports 67 and 68

DHCP Steps

- Step 1. DHCP Discover
  - Client broadcasts on the local physical subnet to find available servers. The client creates a UDP packet with the broadcast destination of 255.255.255.255 and also requests its last-known IP address. The server may ignore this optional parameter
- Step 2. DHCP Offer
  - Server determines the configuration based on the client's hardware address and broadcasts it to the client. It includes the IP address that the server is suggesting that the client can use
- Step 3. DHCP Request
  - Client selects the IP address suggested in the DHCP Offer packets
- Step 4. DHCP Acknowledge
  - Server acknowledges the request and broadcasts the acknowledgement on the local subnet. The client is expected to configure its network interface with the supplied options

DHCP Uses

- DHCP is now extensively used by ISPs to assign addresses to customers
- Also, network address translation (NAT) boxes and wireless base stations provide DHCP services to computers
- Server will do its best to return the same IP address used before
- DHCP can be used to return other parameters
- Note: predecessor of DHCP is BOOTP that was used for diskless workstations

Announcement

- Next is Chapter 9 Wide-area Networks and Security
- 10:50 ~ 12:30 on 9 November (Wednesday)