Computer Networks

Network Layer

Presented by Hung Ba Ngo

Tháng 08/2015

Objectives

- Role of routers in wide area networks
- Services provided to the transport layer
- Routers
- Routing algorithms
- Internet protocol suite

Requisites

- Learners have to be able to
  - Describe a general architecture of layer 3 internetwork and roles of routers in layer 3 internetwork
  - Present services that the network layer provides to the transport layer
  - Explain store and forward packet switching method used by routers
  - Explain principle fields of a routing table
  - Differentiate routing algorithms
  - Implement routing algorithms Dijkstra, Ford-Fulkerson, Distance Vector, Link state

Requisites

- Learners have to be able to
  - Create an IP address plan for an IP network
  - Define subnetwork using classful or classless methods
  - Create manually routing table for routers in an IP network
  - Describe functions of protocols ARP, RARP and ICMP in IP protocol suite

Limitations of data link layer

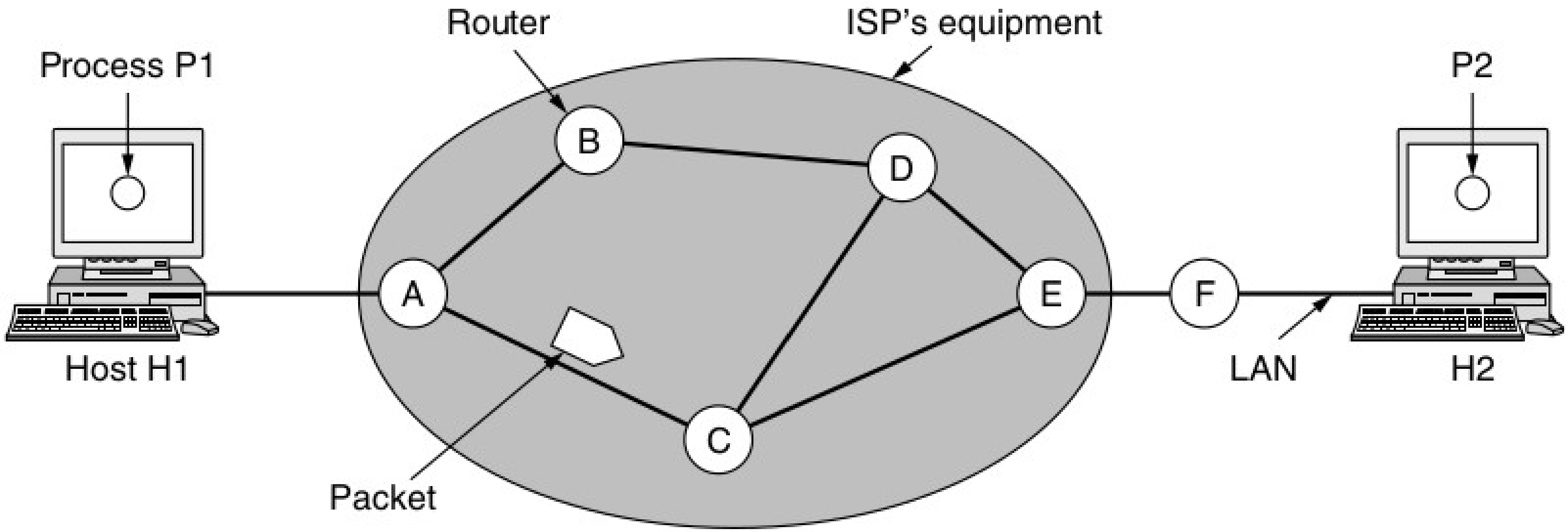
- Just moving frames from one end of a wire to the other
- Have a limitation of number of hosts per a network segment
- Difficult in interconnecting many heterogeneous networks

Roles of network layer

- Provide a host-to-host transmission in a WAN consisting of many heterogeneous LANs
- Moving a packet from one host to another host via many intermediate links
- Find appropriate routes for packets to avoid congestion

Design issues in the network layer

- Store-and-forward packet switching



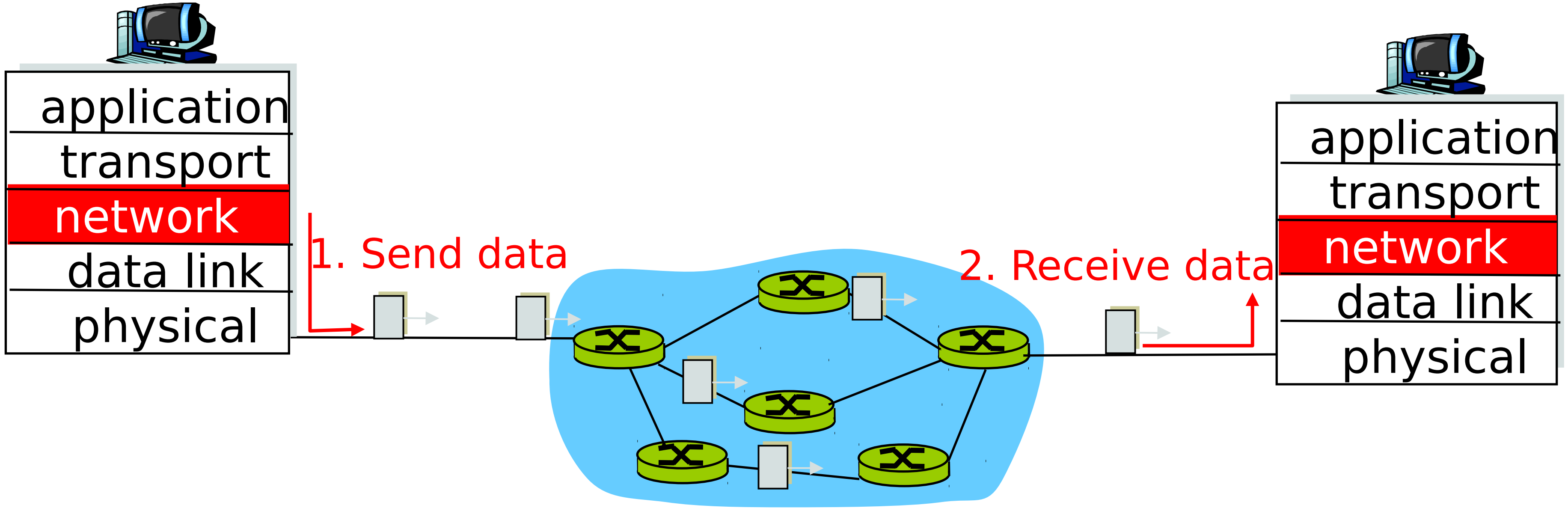
http://www.monitis.com/traceroute/

Design issues in the network layer

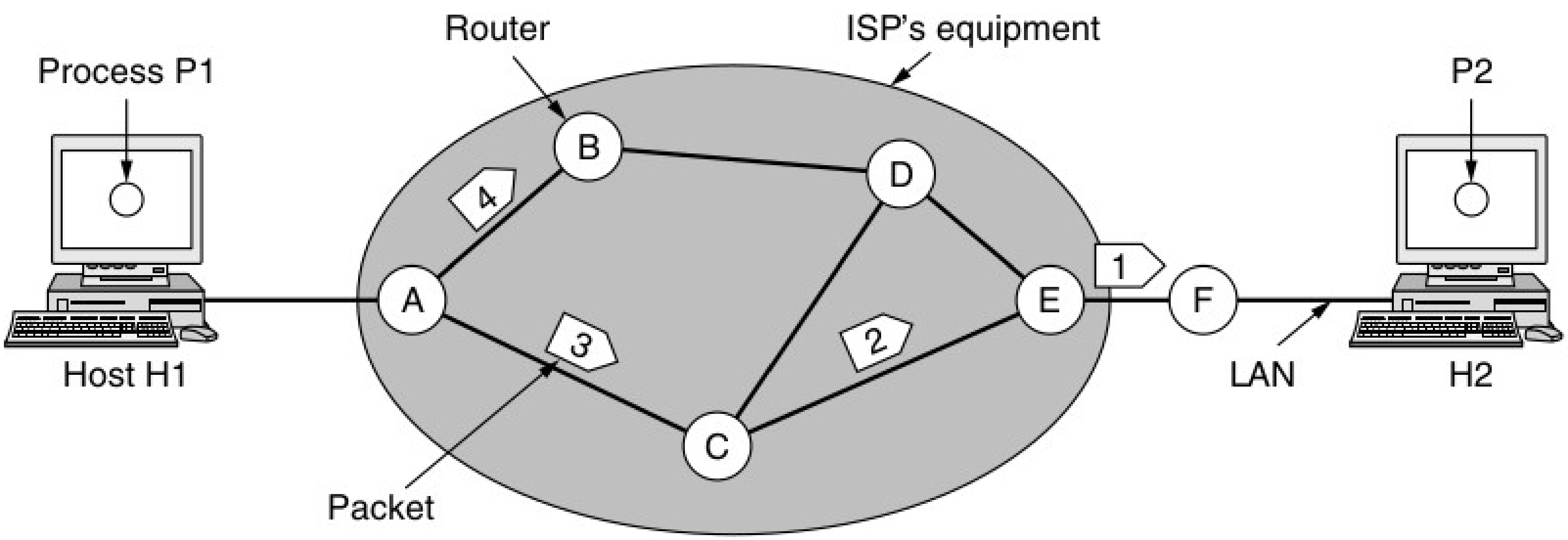
- Services need to be carefully designed with the following goals in mind
  - The services should be independent of the router technology.
  - The transport layer should be shielded from the number, type, and topology of the routers present
  - The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.
- Two services provided to the transport layer
  - Connectionless service
  - Connection-oriented service

Implementation of connectionless service

- Packets are frequently called *datagrams* and network is called a *datagram network*
- No need to establish a path from the source router all the way to the destination router before sending packets
- Data are segmented in to packets that are independently sent in the network through different paths to the same destination



Implementation of connectionless service



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

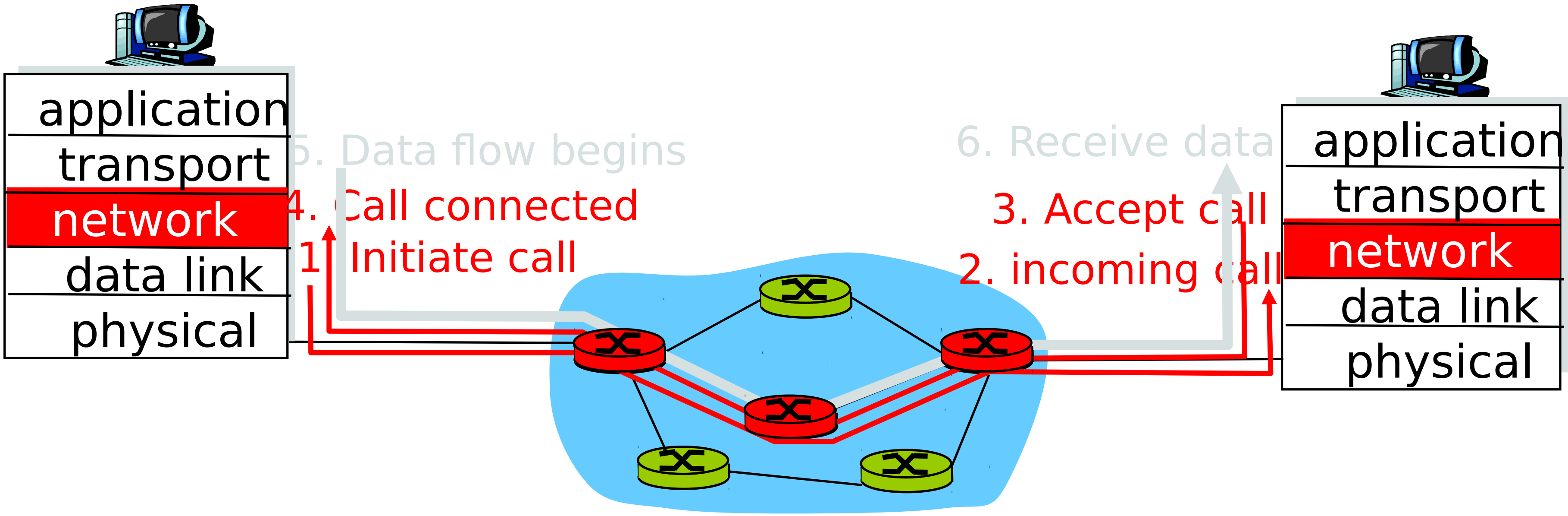
A	C
B	D
C	C
D	D
E	–
F	F

The algorithm that manages the tables
and makes the routing decisions is called the routing algorithm.

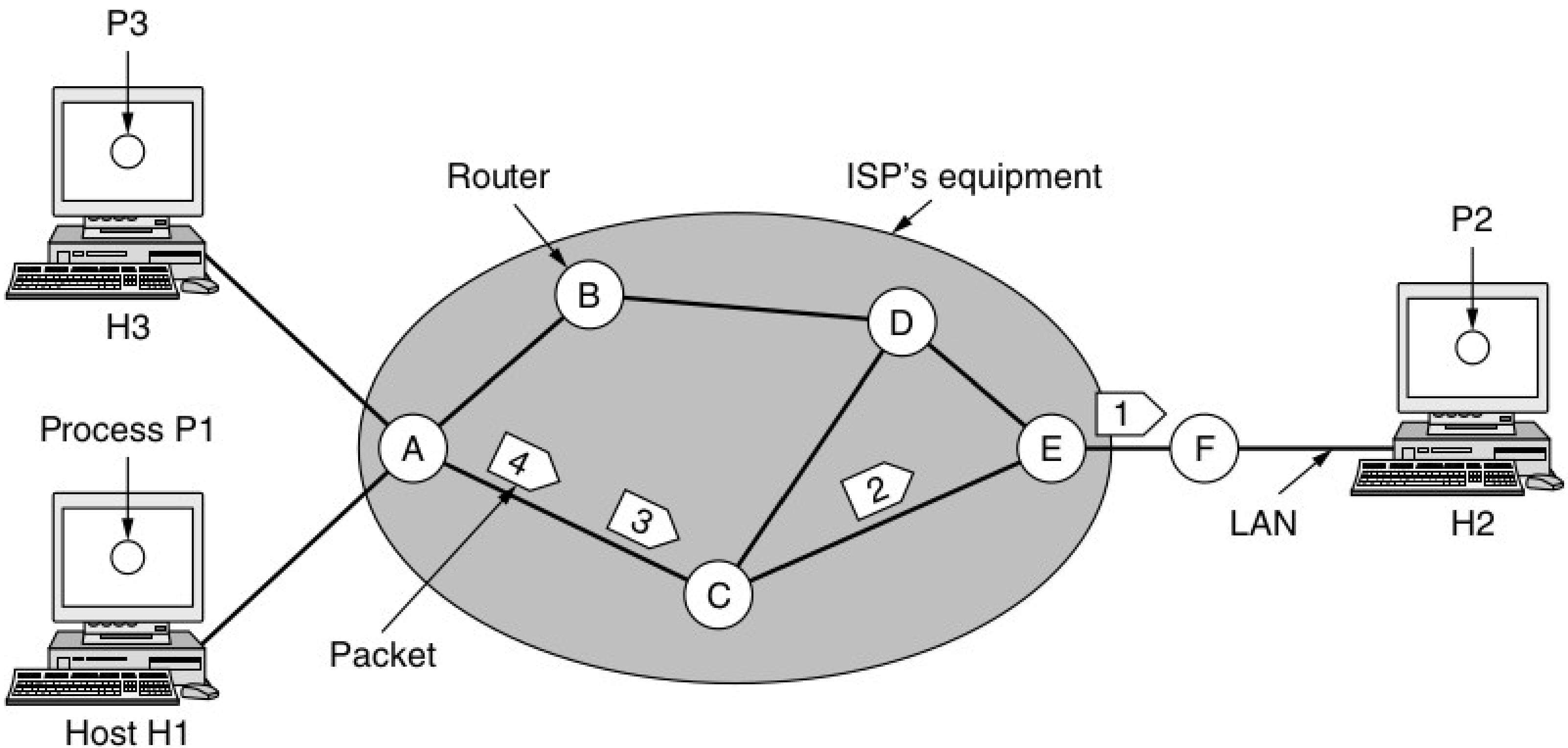
Implementation of Connection-Oriented Service

- A path from the source router all the way to the destination router must be established before any data packets can be sent
- This connection is called a *virtual circuit*, in an analogy with the physical circuits set up by the telephone system, and the network is called a *virtual-circuit network*
- When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- Established route is used for all traffic flowing over the connection
- When the connection is released, the virtual circuit is terminated.
- Each packet carries an identifier telling which virtual circuit it belongs to.

Implementation of Connection-Oriented Service



Implementation of Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Comparison of Virtual-Circuit and Datagram Networks

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

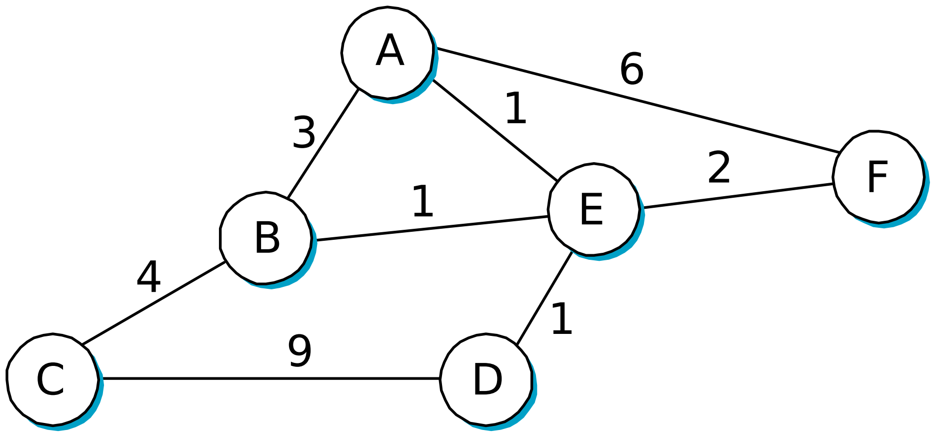
Routing Algorithms

Routing algorithms

- Routing algorithm is the part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on
- In datagram network, the decision must be made a new for every arriving data packet since the best route may have changed since last time
- In virtual circuits network, routing decisions are made only when a new virtual circuit is being set up
- Two process in inside a router
  - Handling each packet as it arrives, looking up the outgoing line to use for it in the routing tables – forwarding process
  - Filling in and updating the routing tables played by routing algorithm

Routing

- Objective: Choosing an appropriate route (a chain of router) through an internetwork from sender host to destination host
- Need to model the network into a graph
  - Nodes: Routers, switchs, HUBs, or sub networks
  - Edges: network links
  - Costs on each edge: Cost for transmitting data via the edge



- Cost of a route is the sum of edges costs on the route
- If there is no route between two nodes, the cost of route is infinite

General objective of routing algorithms

- Finding a route quickly and exactly
- Being able to adapt to the change of network topology
- Being able to adapt to the change of network traffic
- Being able to avoid network congestion
- Low cost in computing

Types of routing algorithm

- Centralized routing: There is a NIC (Network Information Center) responsible for calculating and updating routing table for all routers in the network
- Distributed routing: Each router has to calculate and update its routing table. Routers need to exchange routing information
- Static routing: A router can't update its routing table when network topology is changed. Usually network administrator will create and update routing table manually
- Dynamic routing: Routing table will be updated automatically by the router when there is a network topology change

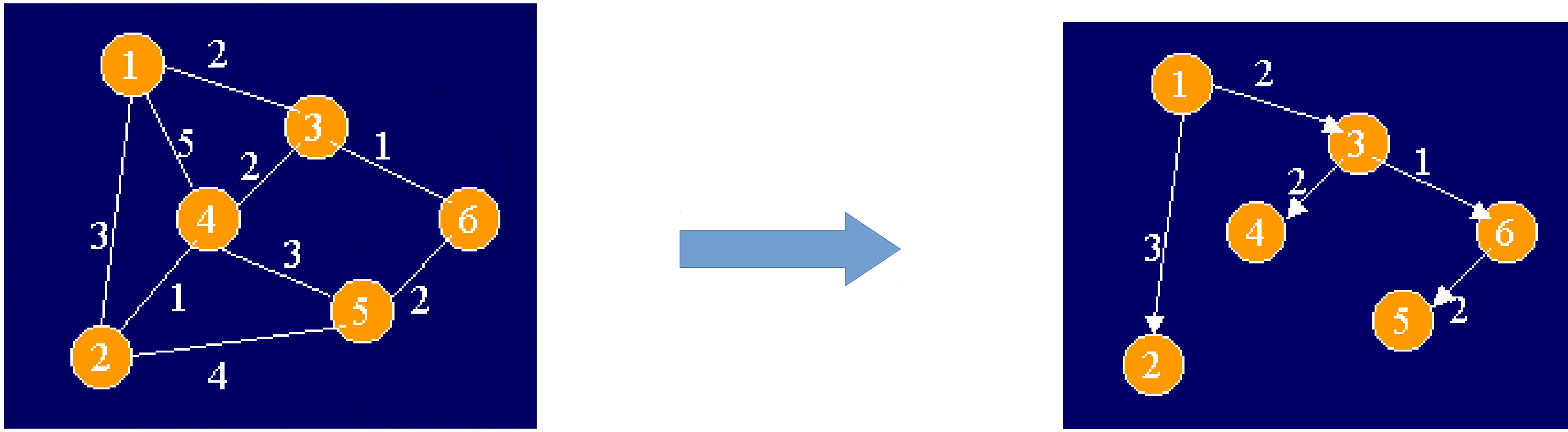
Finding shortest path with Dijkstra algorithms

- Objective: Finding shortest paths from one source node to all other destination nodes in a network
- Being an optimized algorithm – centralized routing algorithm
- Being called
  - S: Source node predefined
  - N: Set of nodes which shortest path (from S) are found
  - D_i: Cost of a shortest path from node S to node i
  - l_{ij}: Cost of edge connecting node i and node j. It will be infinite ∞ if there is no edge between i and j
  - P_i: Parent node of node i

Finding shortest path with Dijkstra algorithms

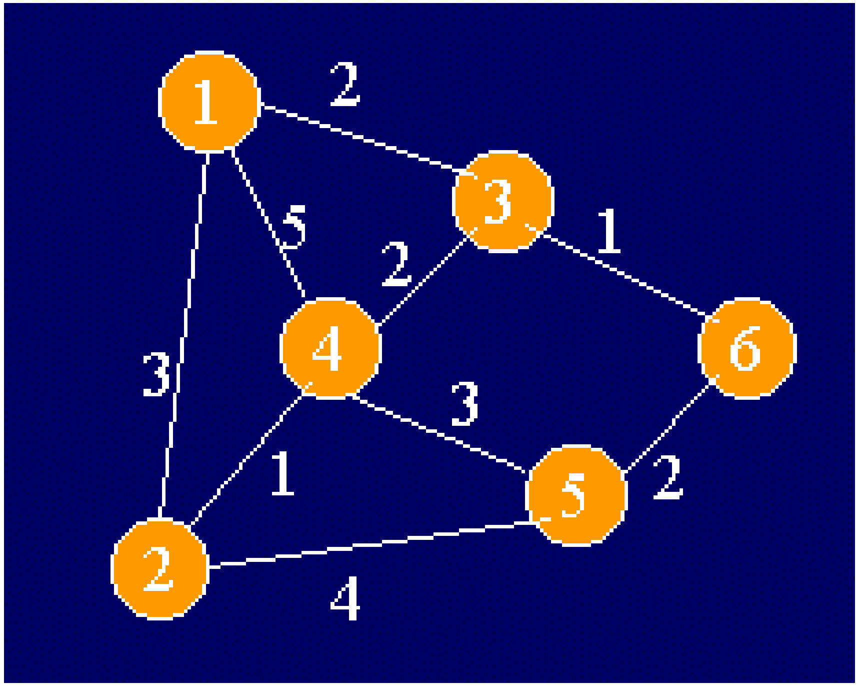
- step 1: Initial step
  - $N=\{S\}$; $D_s=0$;
  - Với $\forall i \neq S$: $D_i=l_{si}$, $P_i=S$
- step 2: Finding a neighborhood nodes
  - Find node **i** $\notin N$, which $D_i = \min(D_j)$ với $j \notin N$
  - Add node **i** into the set N
  - If N contains all node of the graph then stop the algorithm. Otherwise do step 3
- Step 3: Recalculating cost of shortest paths
  - For each node $j \notin N$, recalculate $D_j = \min\{D_j, D_i + l_{ij}\}$; $P_j=i$;
  - Return to step 2

Finding shortest path with Dijkstra algorithms



Lần lặp	N	D_2	D_3	D_4	D_5	D_6	P_2	P_3	P_4	P_5	P_6
Khởi tạo	{1}	3	2	5	∞	∞	1	1	1	1	1
1	{1,3}	3	**2**	4	∞	3	1	1	3	1	3
2	{1,3,2}	**3**		4	7	3	1		3	2	3
3	{1,3,2,6}			4	5	**3**			3	6	3
4	{1,3,2,6,4}			**4**	5				3	6	
5	{1,3,2,6,4,5}				**5**					6	

Finding shortest path with Dijkstra algorithms



- Hãy tìm đường đi ngắn nhất từ nút 6 về các nút còn lại và xây dựng bảng chọn đường cho router 6

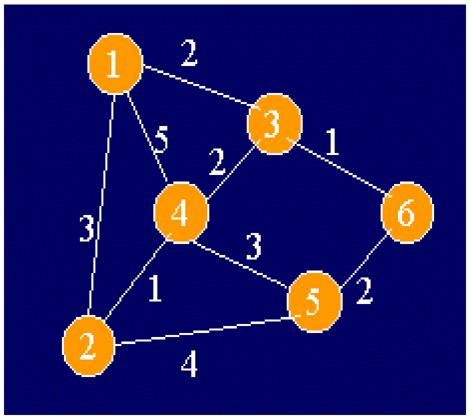
Finding shortest path with Fulkerson algorithms

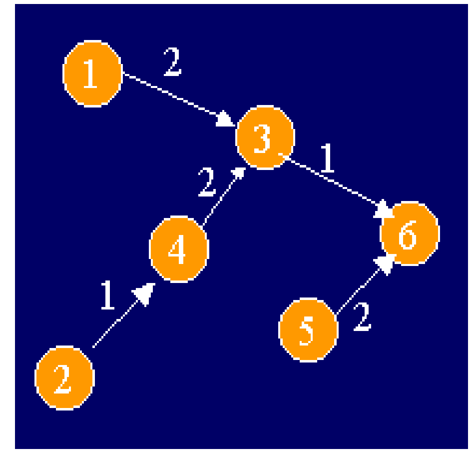
- Objective: Finding shortest paths from all nodes to a destination node
- Being an optimized algorithm – distributed routing algorithm
- Being called
  - d: Destination node predefined
  - D_i: Cost of a shortest path from node i to node d
  - C_i: Child node of node i

Finding shortest path with Fulkerson algorithms

- Step 1: initial step
  - $D_d = 0$;
  - For every $\forall i \neq d$, assign $D_i = \infty$; $C_i = -1$;
- Step 2: Update cost of shortest from i to d
  - $D_i = \min\{ l_{ij} + D_j \}$ với $\forall j \neq i \Rightarrow C_i = j$;
  - Repeate step 2 until no D_i changed

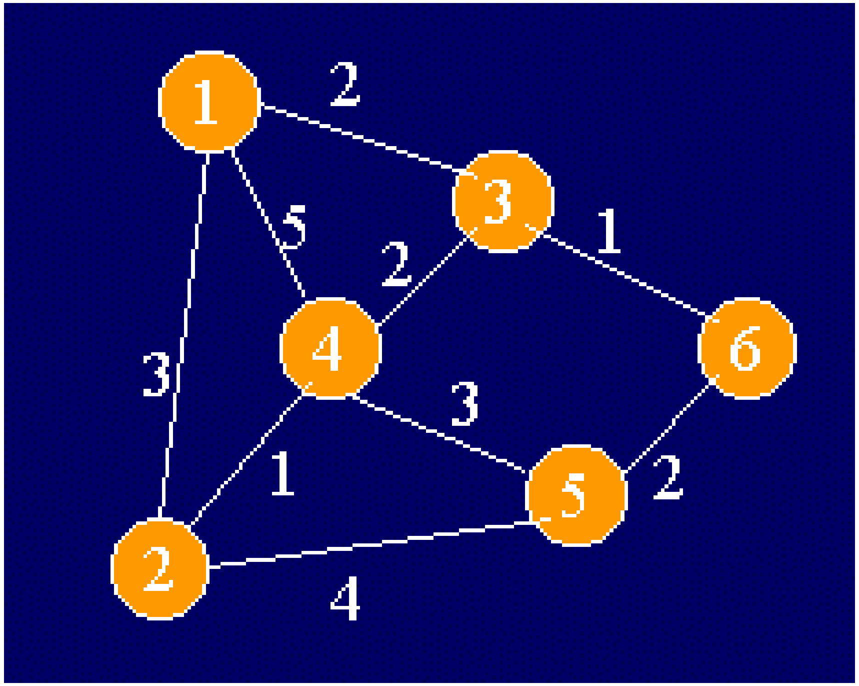
Finding shortest path with Fulkerson algorithms





Lần lặp	D_1	D_2	D_3	D_4	D_5	C_1	C_2	C_3	C_4	C_5
Khởi tạo	∞	∞	∞	∞	∞	-1	-1	-1	-1	-1
1	∞	∞	**1**	**3**	**2**	-1	-1	**6**	**3**	**6**
2	**3**	**4**	1	3	2	**3**	**4**	6	3	6
3	3	4	1	3	2	3	4	6	3	6

Finding shortest path with Fulkerson algorithms

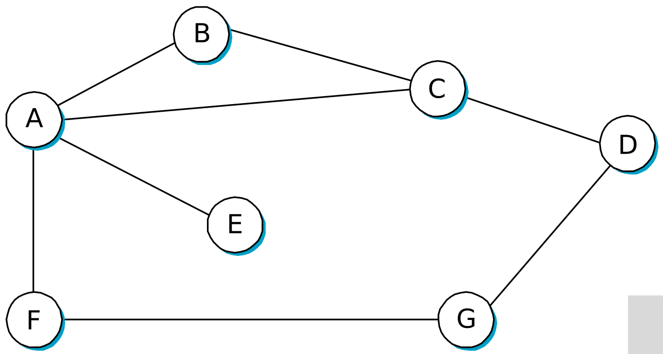


- Tìm đường đi ngắn nhất về các nút 1,2,3,4,5
- Xây dựng bảng chọn đường cho các router 1,2,3,4,5

Distance vector algorithm

- Each router maintains a routing table containing one entry for each router in the network.
  - Entry [destination, Next hop, Cost]
- Cost is distance – hops
- Each router is supposed to know costs of links connecting to its neighborhood
- Cost of a broken/down link is infinite
- A router is required to exchange its routing table with neighborhood periodically
- A router will update its routing table with new shorter routes learned from routing tables of its neighborhood

Distance vector algorithm



Initially, each node will assigns 1 for cost of the link to a neighborhood, and ∞ for links to other router

	Distance to nodes						
	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

Distance vector algorithm

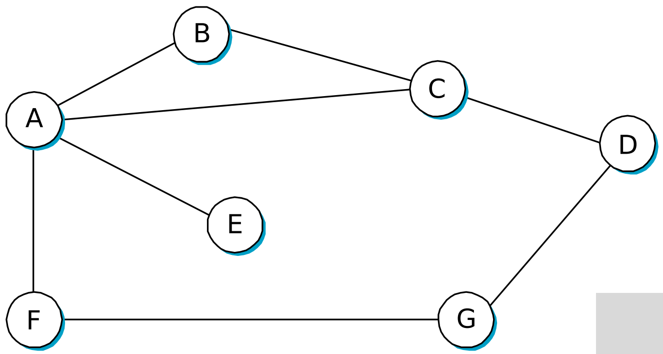
	Distance to nodes						
	A	**B**	**C**	**D**	**E**	**F**	**G**
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Destination	Cost	Next hop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

Destination	Cost	Next hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Routing table of node A

Distance vector algorithm



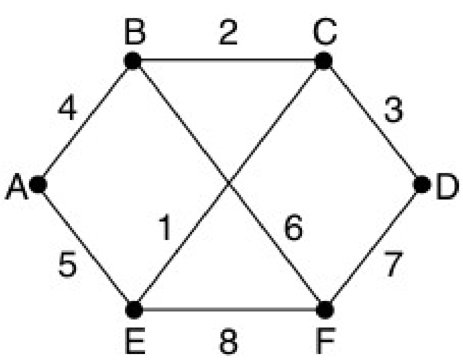
	Distance to nodes						
	A	**B**	**C**	**D**	**E**	**F**	**G**
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Distance vector algorithm

- Issues:
  - When will a router send it routing table to all its neighborhoods ?
    - Periodically, for example: every 30 seconds
    - When the rooting table is modified
  - How to know a neighborhood is alive ?
    - Send a hello message periodically
    - Not receive routing table sent by neighborhood
  - What does a router do when one of its a link is down ?
    - Set the cost of route containing the link to infinite and send its routing table to all its neighborhoods
  - Problem of convergence

Link state algorithm

- Each router must do the following things
  - Discover its neighbors and learn their network addresses
  - Set the distance or cost metric to each of its neighbors
  - Construct a packet telling all it has just learned
  - Send this packet to and receive packets from all other routers
  - Compute the shortest path to every other router.



Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

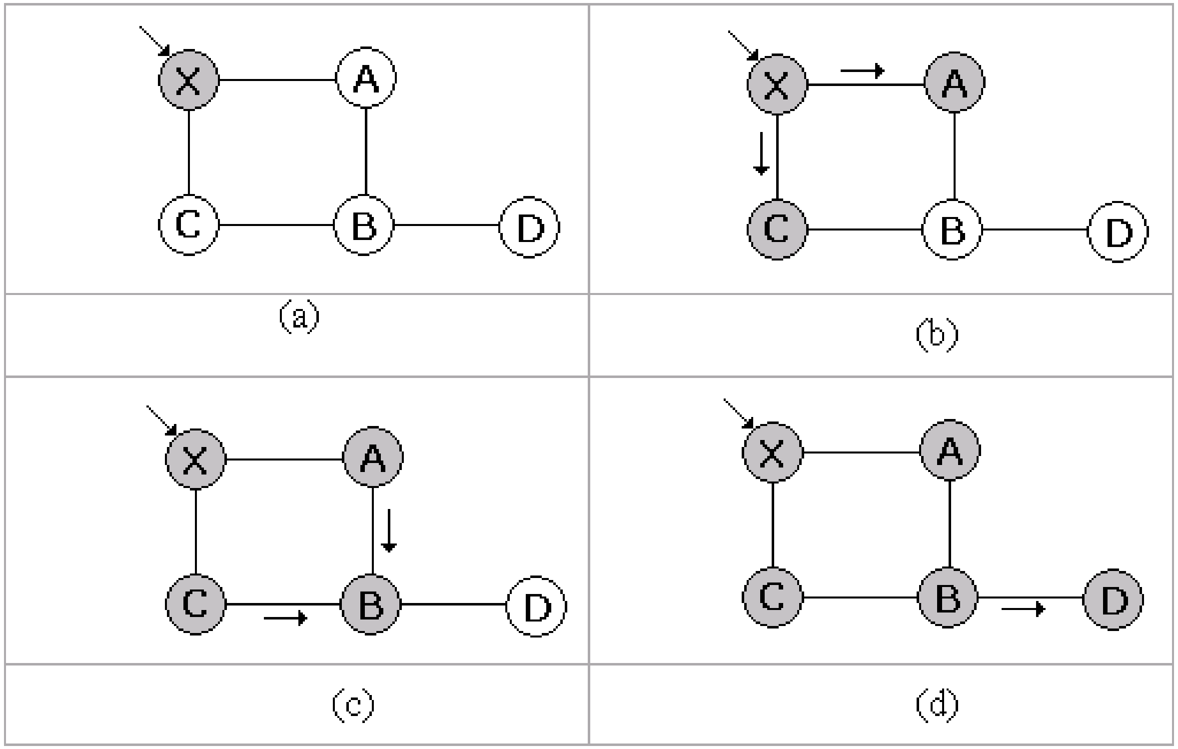
D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

Link state algorithm

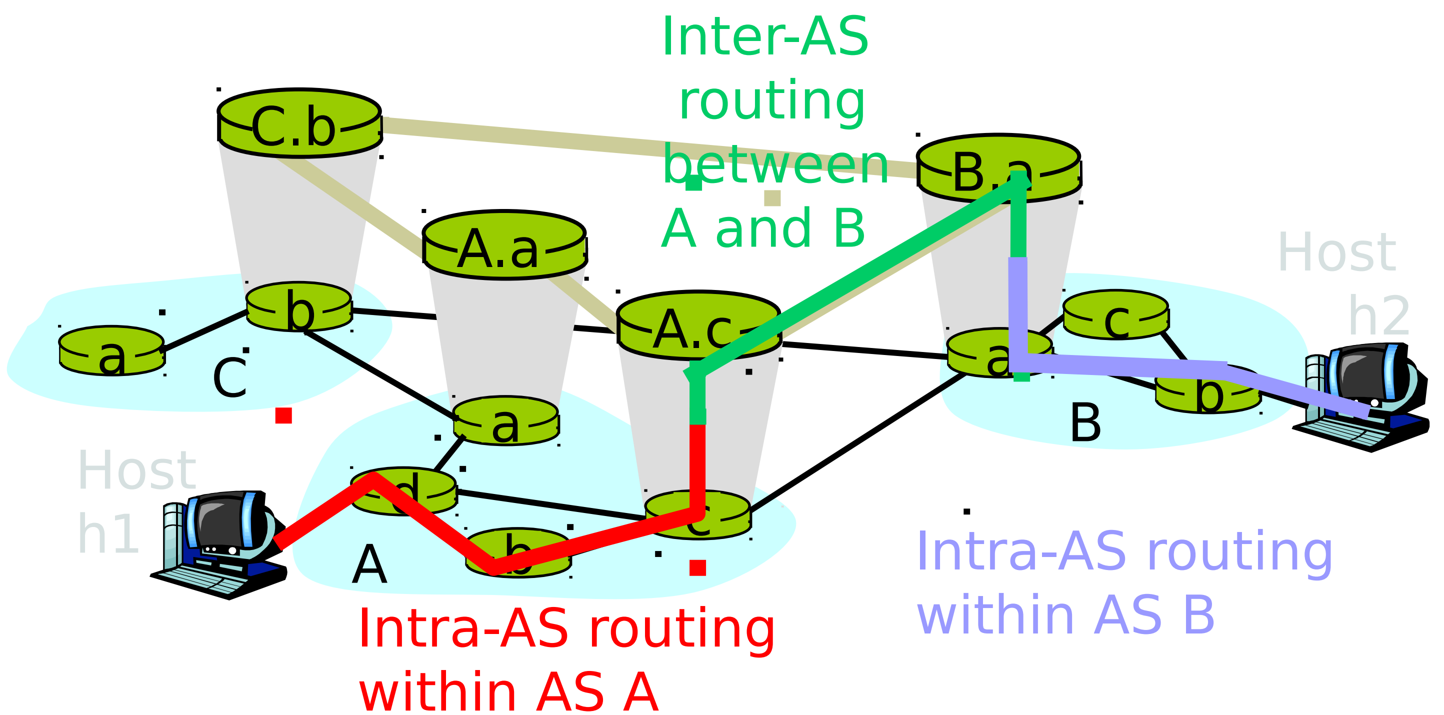
- Reliable flooding protocol to send link state to all routers



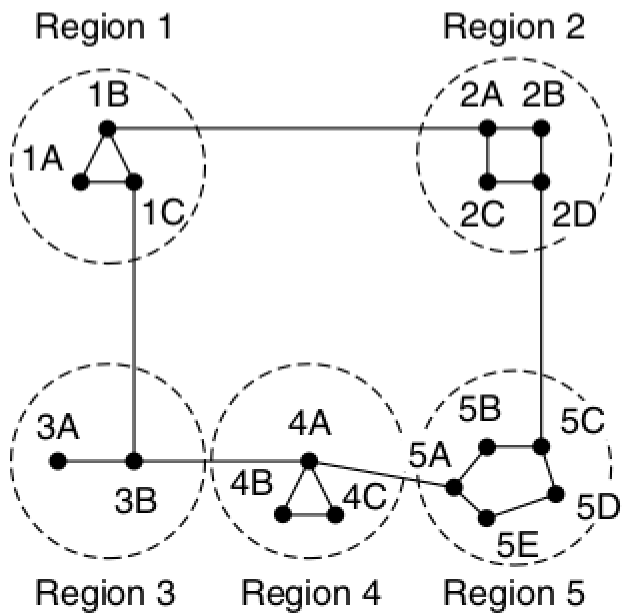
Hierarchical routing

- As networks grow in size
  - the router routing tables grow proportionally
  - router memory consumed by ever-increasing tables
  - CPU time is needed to scan routing tables and more bandwidth is needed to send status reports about them
  - routing will have to be done hierarchically
- When hierarchical routing is used
  - the routers are divided into what we will call domains
  - Each router knows all the details about how to route packets to destinations within its own domain but knows nothing about the internal structure of other domains

Hierarchical routing



Hierarchical routing



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4