WAN Technologies and Routing Strategies

Updated: 11/09/2012

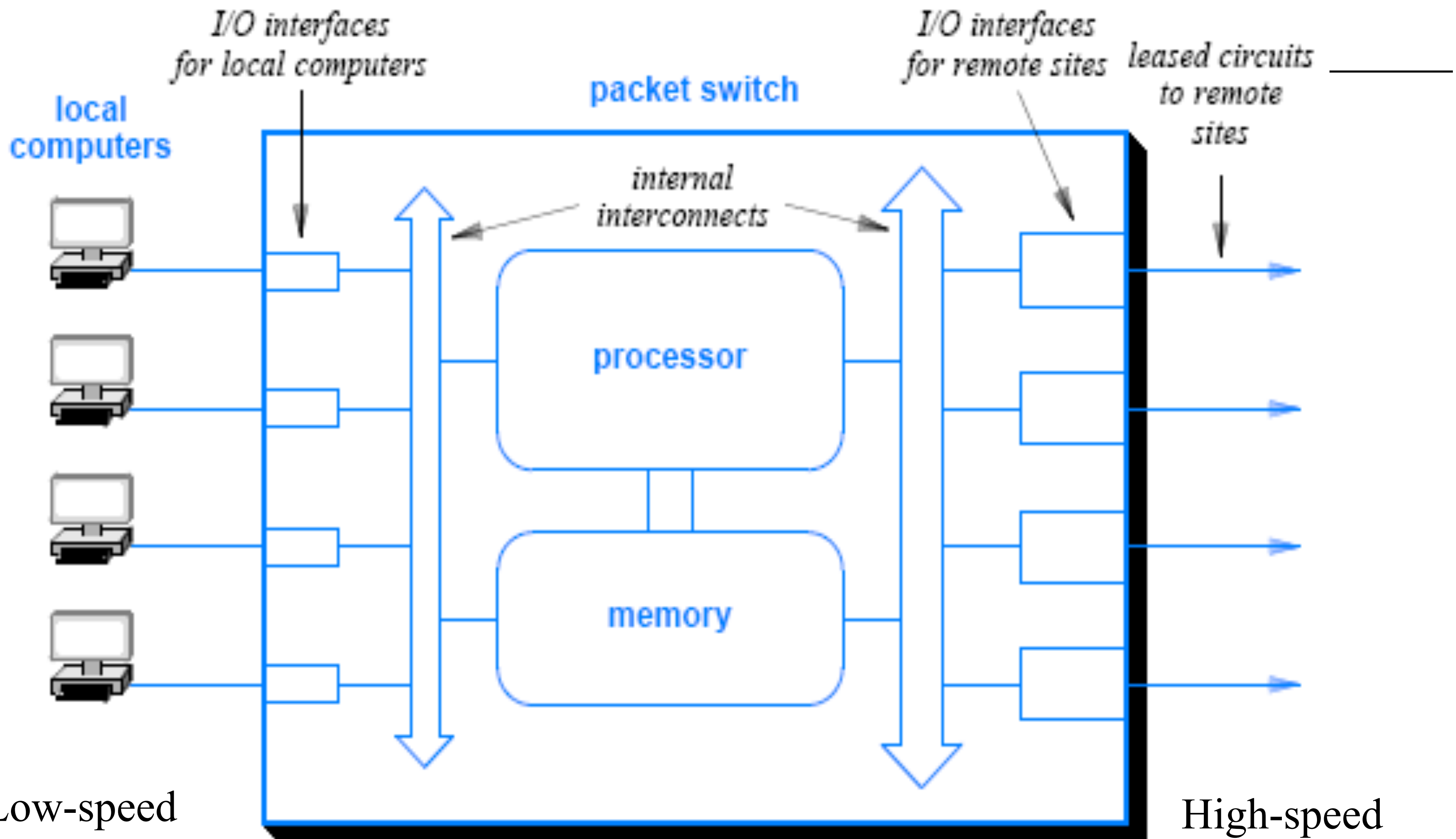
Large Spans and Wide Area Networks

- The key issue that separates WAN technologies from LAN technologies is scalability
  - A WAN must be able to grow as needed to connect many sites
  - spread across large geographic distances
- A technology is not classified as a WAN unless it can deliver reasonable performance for a large scale network
  - A WAN does not merely connect to many computers at many sites
  - It must provide sufficient capacity to permit all computers to communicate
- Thus, a satellite bridge that connects a pair of PCs and printers is merely an extended LAN

Traditional WAN Architecture

- Pre-LAN WAN designers chose to create a special-purpose hardware device that could be placed at each site
- A packet switch provides
  - local connections for computers at the site
  - as well as connections for data circuits that lead to other sites
- A packet switch consists of a small computer system
- Early packet switches were constructed from conventional computers
  - with a processor, memory, and I / O devices used to send and receive packets
  - the packet switches used in the highest-speed WANs require special-purpose hardware

Traditional WAN Architecture

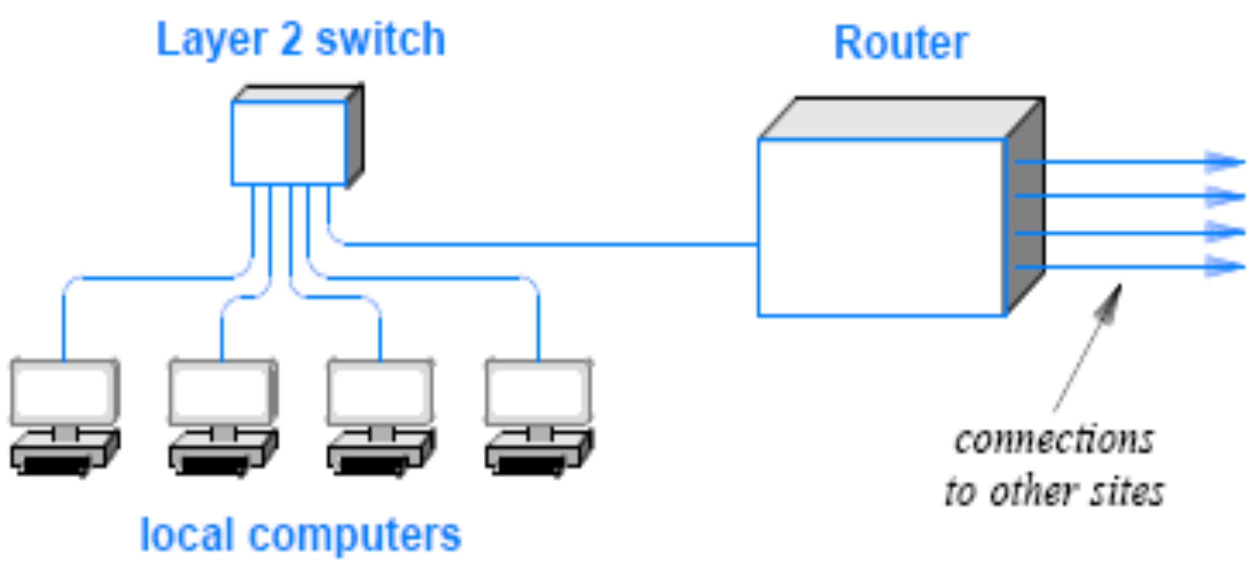


Low-speed Connections

High-speed Connections

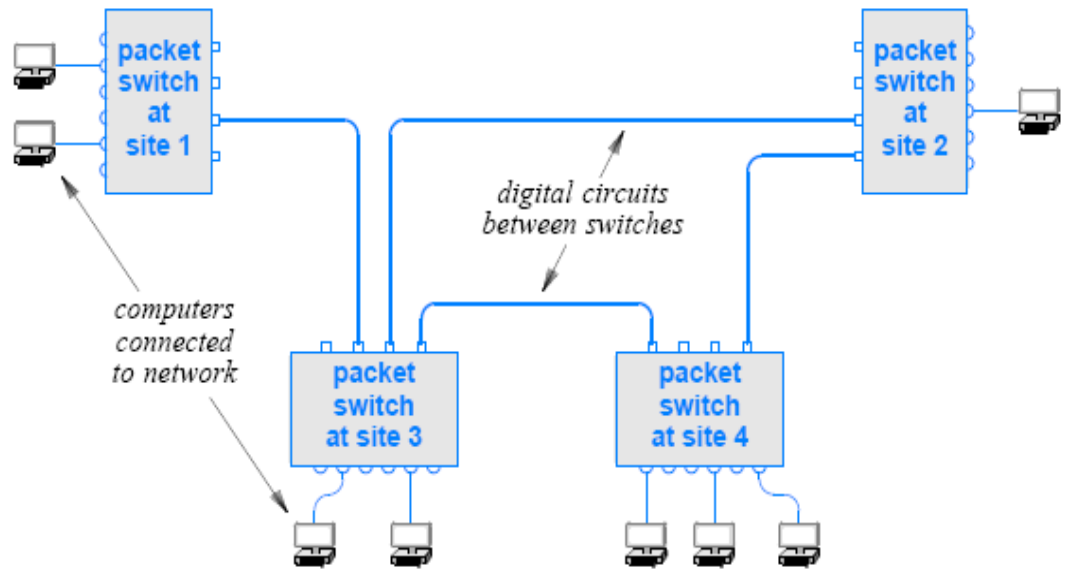
Modern WAN Architecture

- Since the advent of LAN technology, most modern WANs separate a packet switch into two parts:
  - a Layer 2 switch that connects local computers
  - a router that connects to other sites



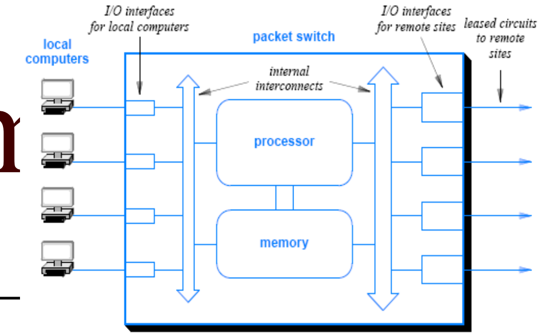
Forming a WAN

- A WAN can be formed by interconnecting a set of sites
- The exact details of the interconnections depend on
  - the data rate needed
  - the distance spanned
  - and the delay



- Many WANs use leased data circuits
- A network designer must choose a topology
  - For a given set of sites, many topologies are possible

Store and Forward Paradigm

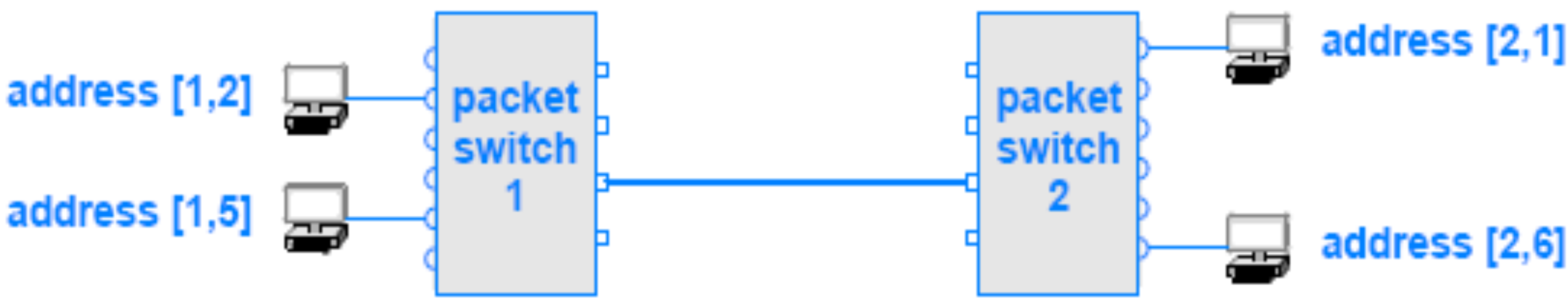


- WAN is to allow
  - as many computers as possible to send packets simultaneously
- The fundamental paradigm used to achieve simultaneous transmission is known as store and forward
- To perform store and forward processing
  - a packet switch buffers packets in memory
- The store operation occurs when a packet arrives:
  - I / O hardware in the switch places a copy of the packet in memory
- The forward operation occurs once a packet has arrived and is waiting in memory. The processor
  - examines the packet
  - determines its destination
  - and sends the packet over the I / O interface that leads to the destination

Addressing in a WAN

- WANs addresses follow a key concept that is used in the Internet: hierarchical addressing
  - Hierarchical addressing divides each address into two parts:

    (site, computer at the site)
  - In practice, instead of a identifying a site, each packet switch is assigned a unique number
    - first part of an address identifies a packet switch
    - second part identifies a specific computer

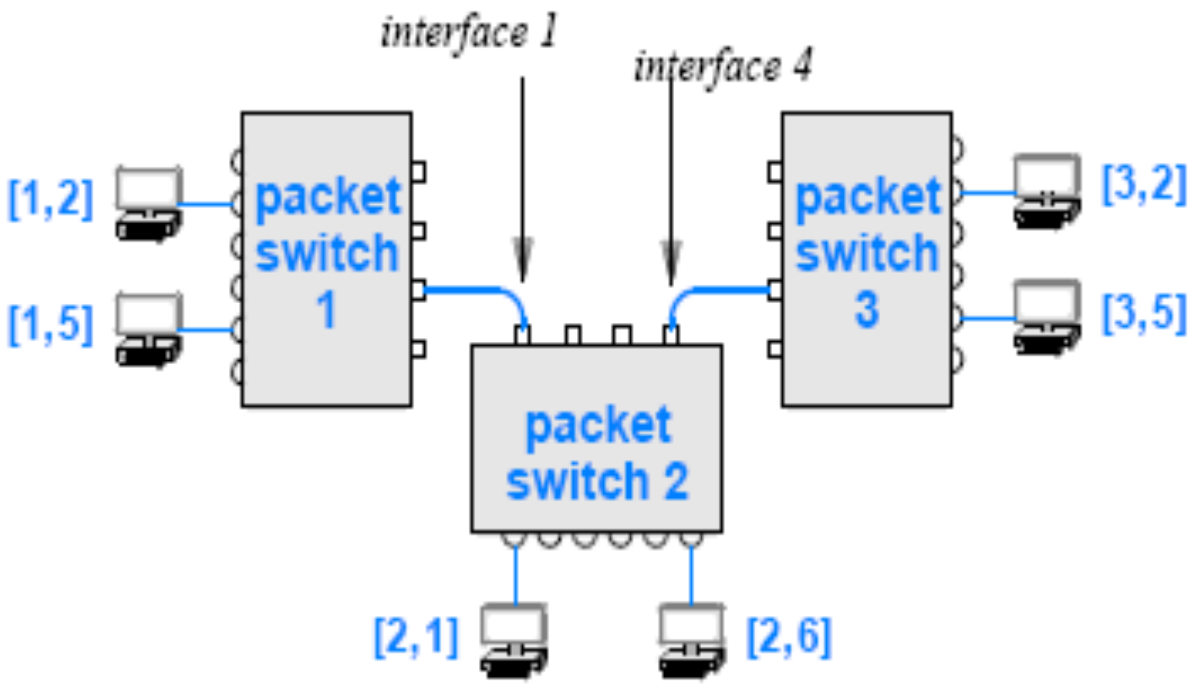


- A computer connected to port 6 on packet switch 2 is assigned address [2, 6]

Next-Hop Forwarding

- What is the importance of hierarchical addressing?
- When a packet arrives
  - a switch must choose an outgoing path over which to forward it
- To make the choice, a packet switch
  - examines the destination address in the packet
  - and extracts the packet switch number
    - If the number in the destination address is identical to the packet switch's own ID the packet is intended for a computer on the local packet switch
    - Otherwise, the packet is intended for a computer on another switch

Next-Hop Forwarding



(a)

- one entry per packet switch instead of one entry per destination computer

to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

(b)

When packet with [3,5] arrives, the switch extracts 3 → send it to Interface (port) 4; as indicated by the Forwarding Table.

Next-Hop Forwarding

- Using only one part of a two-part hierarchical address to forward a packet has two practical consequences
  - First, the computation time required to forward a packet is reduced because the forwarding table can be organized as an array that uses indexing instead of searching
  - Second, the forwarding table contains one entry per packet switch instead of one entry per destination computer
    - The reduction in table size can be substantial, especially for a large WAN that has many computers attached to each packet switch
- A two-part hierarchical addressing scheme allows packet switches to use only the first part of the destination address until the packet reaches the final switch
  - Once the packet reaches the final switch
    - the switch uses the second part of the address to choose a specific computer

Forwarding Mechanism and Routing

- Performance
  - simplest is “minimum hop”
  - can be generalized as “least cost”
    - Cost is assigned based on the designed objective: delay, Queue built-up, TH, hop-count, etc.
- Decision time and place
  - Time – when the routing decision is made
    - Packet (datagram) or virtual circuit basis (session based)
    - fixed or dynamically changing
  - Place – which node makes the decisions
    - distributed - made by each node (most common)
    - centralized
    - source
- Networking information (next)

Network Information Source and Update Timing

- routing decisions usually based on knowledge of network (not always)
  - distributed routing
    - using local knowledge, info from adjacent nodes, info from all nodes on a potential route
  - central routing
    - collect info from all nodes
- issue of update timing
  - how often updated?
  - fixed - never updated
  - adaptive - regular updates

Information Type	Dissemination
Topology	
Load	Local Information
Link Cost	Adjacent Nodes
Congestion	Global
Etc.	

Routing Metrics

- **Hop count:** Each router through which a packet must pass is considered a hop. Counting the hops on a route gives an indication of the path's length. The lower the path length, the better the route.
- **Ticks:** Each tick represents one-eighteenth of a second and represents a delay across a route.
- **Cost:** The cost of a path is an arbitrary value associated with each link crossed on the path. Slower links typically have a higher cost associated with them than do faster links. The route with the lowest total path cost is typically the route selected as the fastest.
- **Bandwidth:** The maximum throughput of a link, in terms of bits per second, is considered its bandwidth. The route with the highest bandwidth is considered to be the fasted route possible. This is not always the case, because a high-bandwidth link may already have too many users sending data across the link, effectively slowing the link. A link with a lower bandwidth may not have as many users and be able to send the data instantly.
- **Delay:** The summation of many factors results in a delay rating, a commonly used metric. These factors include link bandwidth, router queue length, network congestion, and physical distance.
- **Load:** This is a dynamic factor that is based on such items as router processor utilization and packets processed per second. Although it's an effective metric, the monitoring of these items may require high resource demand.

Routing Algorithm Classification (Different Views)

- Global or local
  - Global: Entire network is known
  - Local: Partial knowledge about the network
- Centralized or decentralized
  - One node maintain view of the network
- Static or Dynamic
  - Frequent route change vs. fix routes

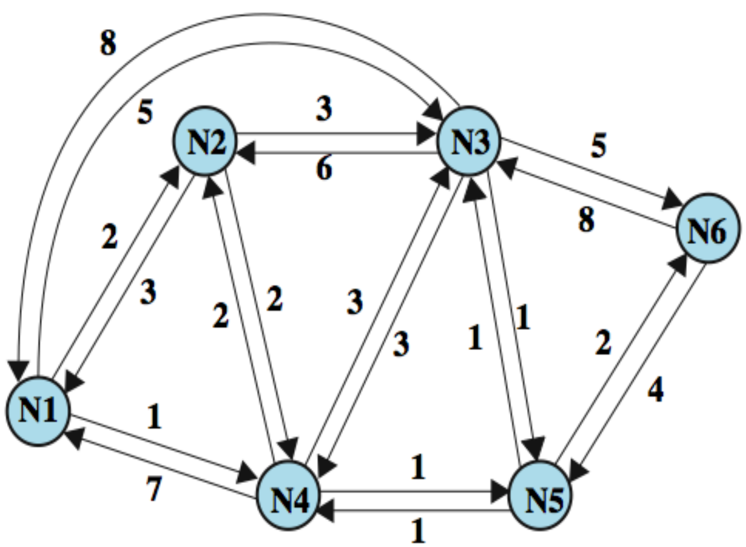
Routing Mechanisms-
How Forwarding Tables are Setup

- Fixed Configuration
- Flooding
- Random Routing
- Distributed Adaptive (Dynamic) Routing
  - There are two general forms:
    - Link-State Routing (LSR), which uses Dijkstra's algorithm
    - Distance-Vector Routing (DVR), which uses another approach

http://weierstrass.is.tokushima-u.ac.jp/ikeda/suuri/dijkstra/DijkstraApp.shtml?demo1

http://net237.tripod.com/link.htm

Fixed Routing Tables



- Only next node is known
-Not much processing is required

CENTRAL ROUTING DIRECTORY

NEXT-HOP from 6 to 2

To Node \ From Node	1	2	3	4	5	6
1	—	1	5	2	4	5
2	2	—	5	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
1	2
2	2
3	5
5	5
6	5

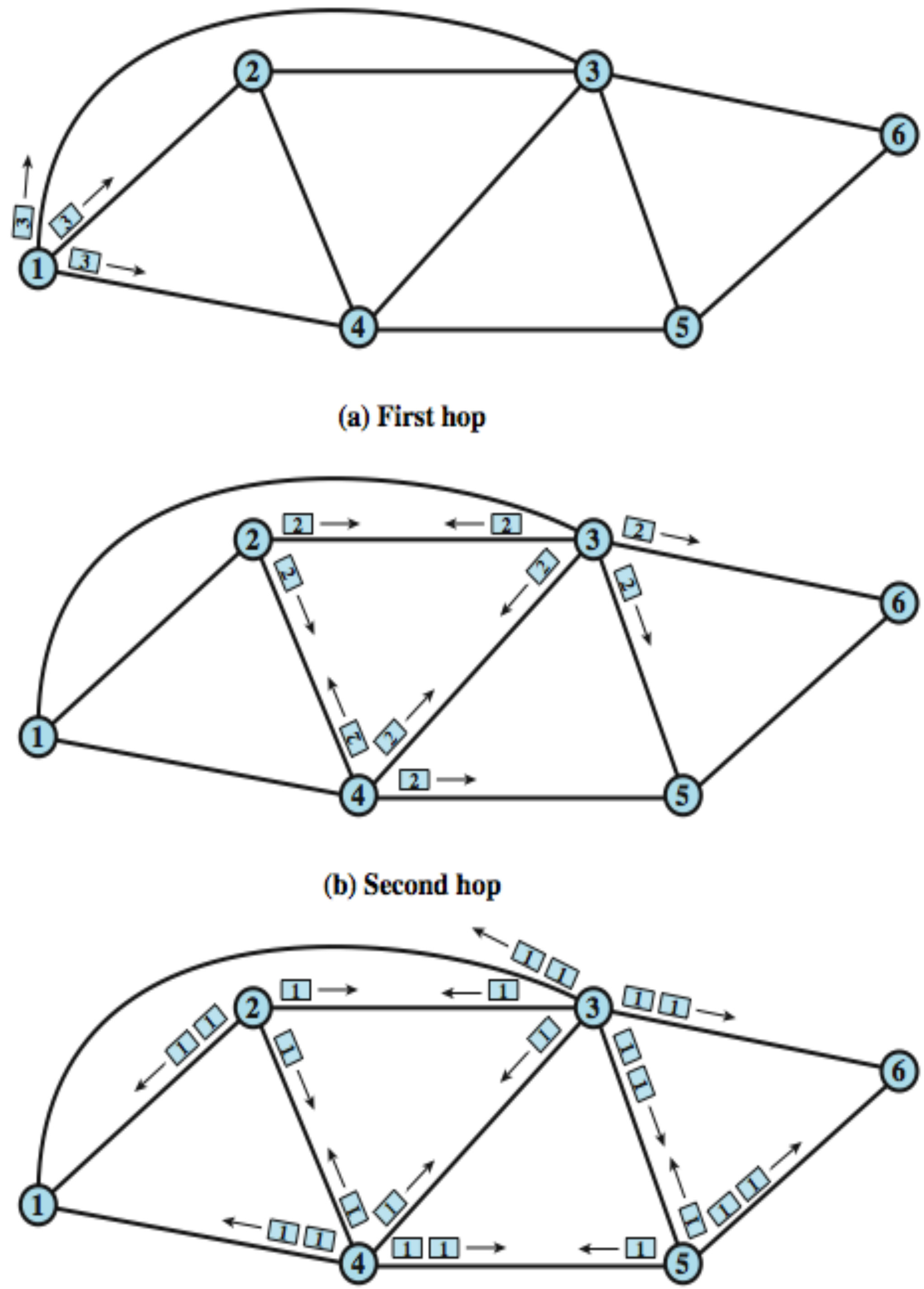
Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5

Flooding Example



(a) First hop

(b) Second hop

(c) Third hop

Assume hop-count is 3

Issues:

- Low link utilization
- High contention
- Packet duplication

Adaptive Routing

- used by almost all packet switching networks
- routing decisions change as conditions on the network change due to failure or congestion
- requires info about network
- disadvantages:
  - decisions more complex
  - tradeoff between quality of network info and overhead
  - reacting too quickly can cause oscillation
  - reacting too slowly means info may be irrelevant

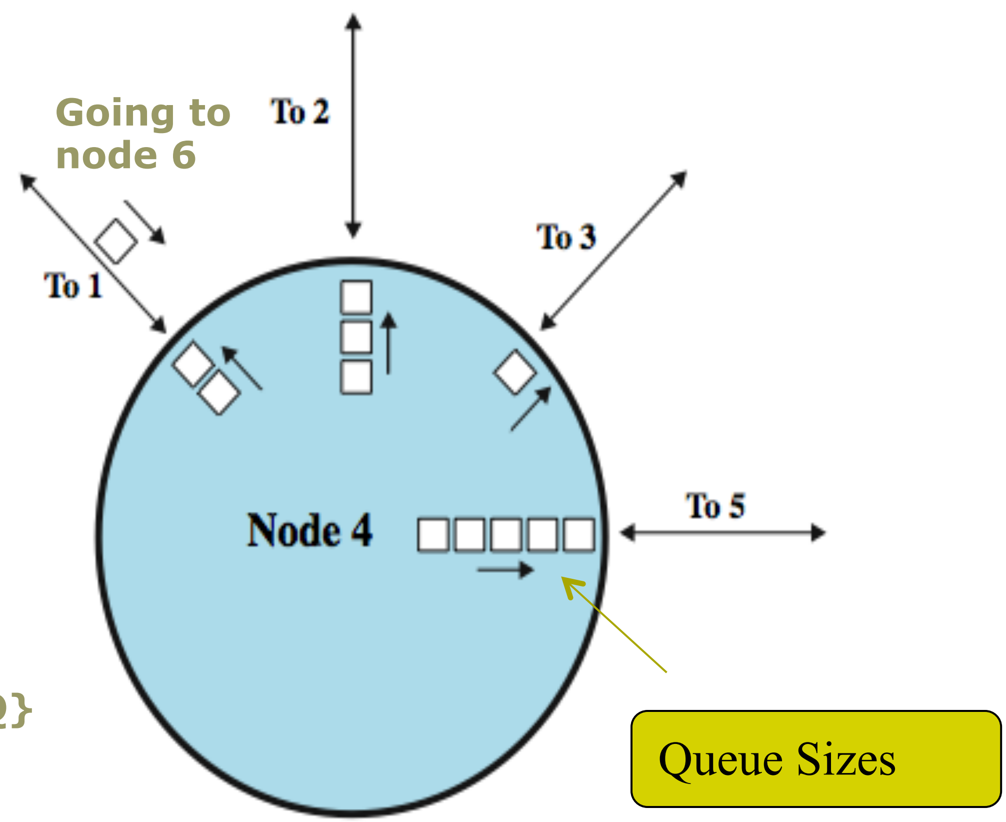
Classification of Adaptive Routing Strategies

- Base on information sources
  - Local (isolated)
    - Rarely used - does not make use of network info
    - Route to outgoing link with shortest queue
    - Can include bias for each destination
  - Adjacent nodes
    - Takes advantage of delay / outage info
    - Distributed or centralized
  - All nodes
    - like adjacent

Isolated Adaptive Routing – Biased-Routing Strategy

Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0



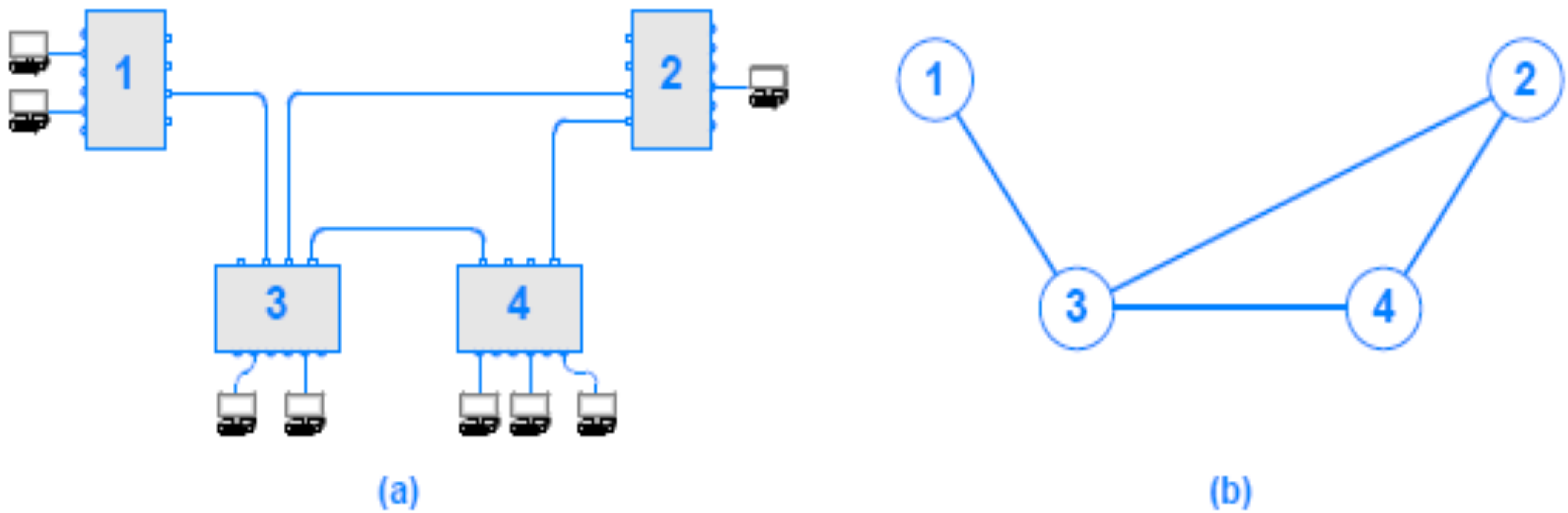
Next Node = min{Bias6 + Q}

Node 1: 9+2 = 11
Node 2: 6+3 = 9
Node 3: 3+1 = 4
Node 4: 0+5 = 5

Adaptive (Dynamic) Routing Updates in a WAN

- We use the term routing software to describe software that automatically reconfigures forwarding tables
- Route computation in a WAN is to think of a **graph** that models the network
  - software uses the graph to compute the shortest path to all possible destinations
- A graph representation is useful in computing next-hop forwarding
  - because graph theory has been studied and efficient algorithms have been developed
  - a graph abstracts away details, allowing routing software to deal with the essence of the problem
- When it computes next-hop forwarding for a graph
  - a routing algorithm must identify a link

Dynamic Routing Updates in a WAN

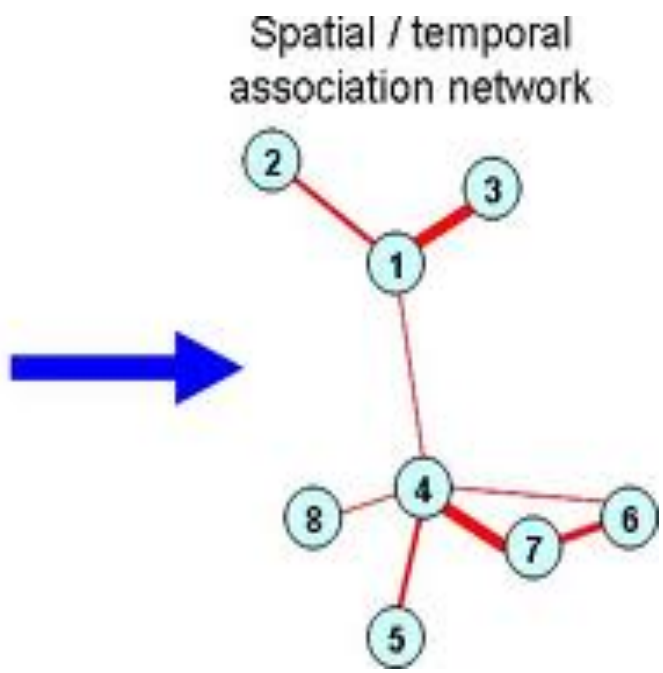


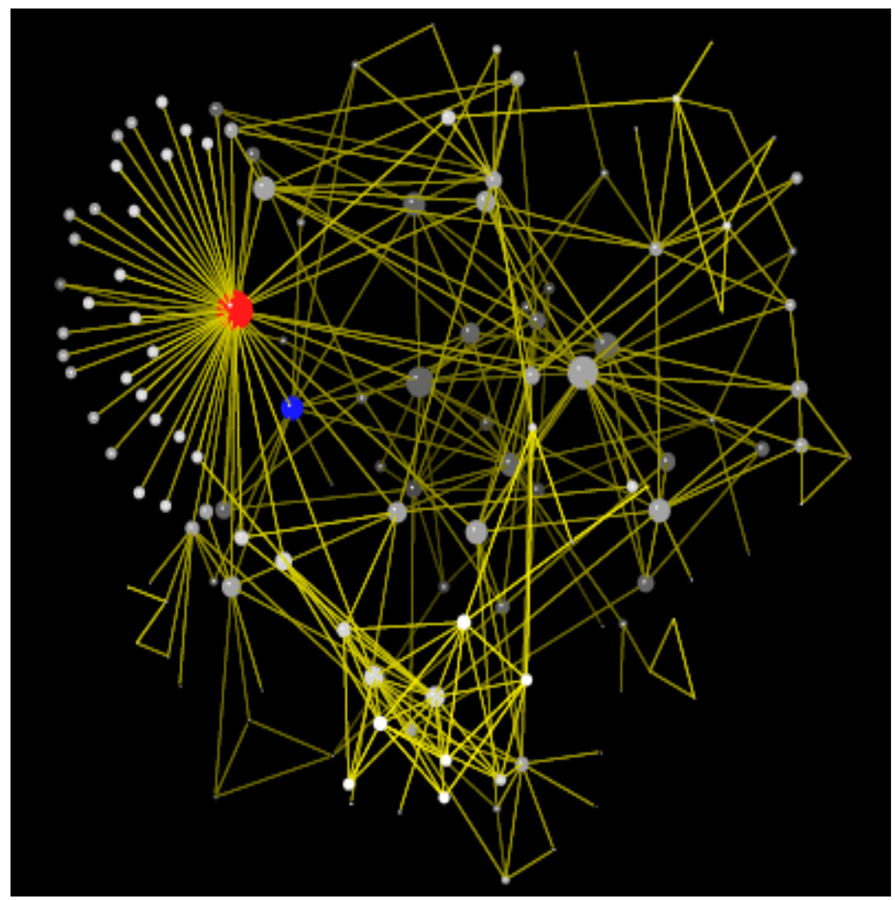
•Each node in the graph corresponds to a packet switch in the network (individual computers are not part of the graph)
•The edge or link exists between the corresponding nodes

Graph Examples

Encounter log

Node ID	Other ID	Cost Dur.	Dist.
1	2	3	5
1	3	10	2
1	4	1	10
4	5	4	4
4	6	2	8
4	7	10	3
6	7	5	2
4	8	3	3





Social Graph Used by Facebook!

Dynamic Routing Updates in a WAN

to reach	next hop
1	–
2	(1,3)
3	(1,3)
4	(1,3)

node 1

to reach	next hop
1	(2,3)
2	–
3	(2,3)
4	(2,4)

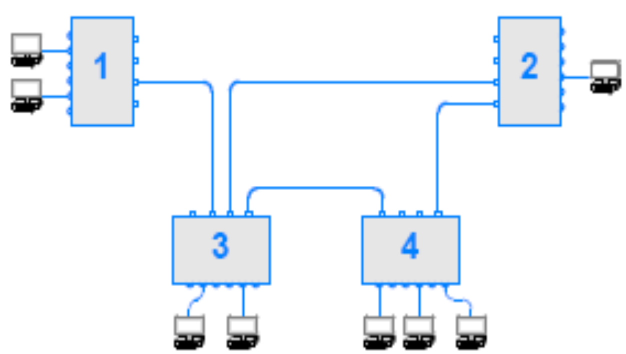
node 2

to reach	next hop
1	(3,1)
2	(3,2)
3	–
4	(3,4)

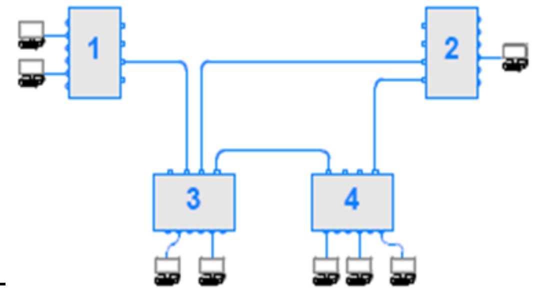
node 3

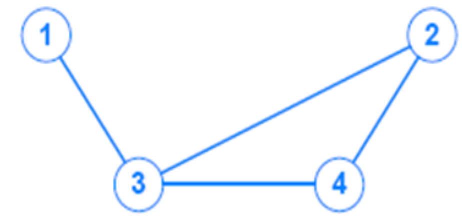
to reach	next hop
1	(4,3)
2	(4,2)
3	(4,3)
4	–

node 4



Default Routes





- Default route is a mechanism that allows a single entry in a forwarding table to replace a long list of entries that have the same next-hop value
  - Only one default entry is allowed in a forwarding table
  - and the entry has lower priority than other entries

to reach	next hop
1	–
*	(1,3)

node 1

to reach	next hop
2	–
4	(2,4)
*	(2,3)

node 2

to reach	next hop
1	(3,1)
2	(3,2)
3	–
4	(3,4)

node 3

to reach	next hop
2	(4,2)
4	–
*	(4,3)

node 4

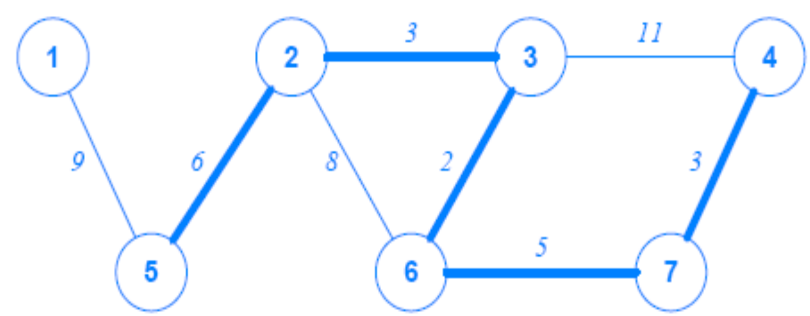
Distributed Route Computation

- In practice, WANs need to perform distributed route computation
  - Instead of a centralized program computing all shortest paths
    - each packet switch must compute its own forwarding table locally
  - All packet switches must participate in distributed route computation
- There are two general forms:
  - Link-State Routing (LSR), which uses Dijkstra's algorithm
  - Distance-Vector Routing (DVR)

Distributed Route Computation

Routing (LSR) – Shortest Path Routing

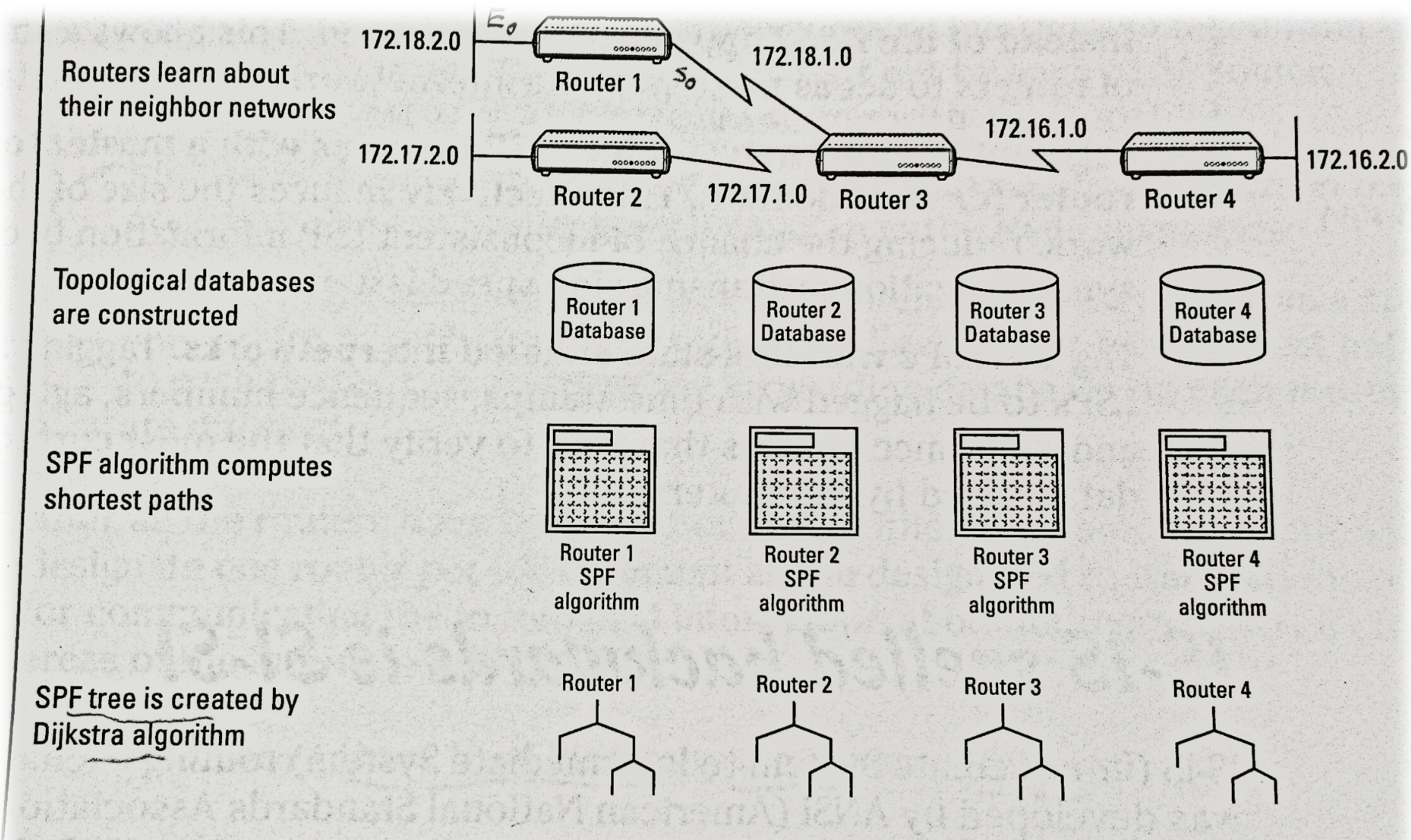
- Link-state routing or Link-status routing
  - the approach also known as Shortest Path First (SPF) routing
  - Dijkstra algorithm used it to characterize the way it works
    - actually all routing algorithms find shortest paths
- To use LSR, packet switches periodically send messages across the network that carry the status of a link
  - For example, packet switches 5 and 2 measure the link weight between them and send a status message
    - such as "the link between 5 and 2 is up"
  - Each status message is broadcast to all switches
- Every switch collects incoming status messag
  - and uses them to build a graph of the network



Dijkstra algorithm is used to find the shortest path between nodes

Distributed Route Computation

Routing (LSR) – 5 Steps of LSR (SPF = Shortest Path First)



Router 1

Network	Port	Disc
172.18.2.0	E0	0
172.18.1.0	S0	0
172.16.1.0	S0	1
172.17.1.0	S0	1
172.16.2.0	S0	2
172.17.2.0	S0	2

Router 2

Network	Port	Disc
172.17.2.0	E0	0
172.17.1.0	S0	0
172.16.1.0	S0	1
172.17.1.0	S0	1
172.16.2.0	S0	2
172.17.2.0	S0	2

Router 3

Network	Port	Disc
172.16.1.0	S0	0
172.17.1.0	S1	0
172.18.1.0	S2	0
172.16.2.0	S0	1
172.17.2.0	S1	1
172.18.2.0	S2	1

Router 4

Network	Port	Disc
172.16.2.0	E0	0
172.16.1.0	S0	0
172.17.1.0	S0	1
172.18.1.0	S0	1
172.17.2.0	S0	2
172.18.2.0	S0	2

Distributed Route Computation

Link-State Routing (LSR) - Advantages

- An LSR algorithm can adapt to hardware failures
- If a link between packet switches fails
  - the attached packet switches will detect the failure and broadcast a status message that specifies the link is down
- All packet switches receive the broadcast
  - change their copy of the graph to reflect the change in the link's status and re-compute shortest paths
- Similarly, when a link becomes available again
  - the packet switches connected to the link detect that it is working and start sending status messages that report its availability

Distributed Route Computation

Distance Vector Routing (DVR)

- As with LSR, each link in the network is assigned a weight
- The distance to a destination between two packet switches is defined to be the sum of weights along the path between the two
- DVR arranges for packet switches to exchange messages periodically (similar to LSR)
- In DVR, a switch sends a complete list of destinations and the current cost of reaching each
- When it sends a DVR message
  - a switch is sending a series of individual statements, of the form:

    "I can reach destination X, and its current distance from me is Y"

Distributed Route Computation

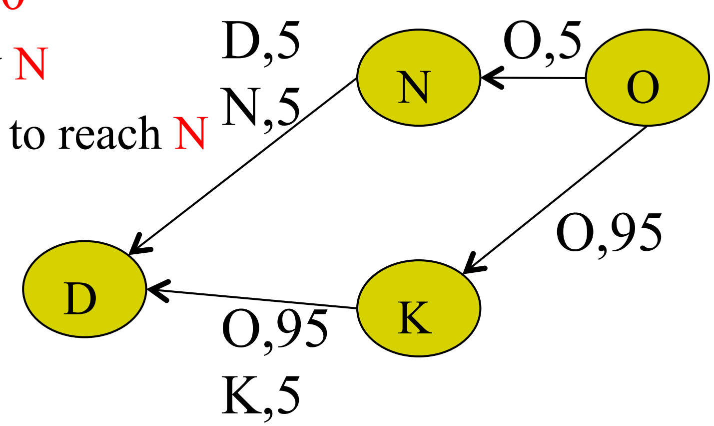
Distance Vector Routing (DVR) – Basic Operation

- DVR messages are not broadcast
  - Each switch periodically sends a DVR message to its neighbors
- Each message contains pairs of (destination, distance)=X,Dx
- Each packet switch must keep a list of possible destinations
  - along with the current distance to the destination and the next hop to use
  - the list of destinations and the next hop for each can be found in the forwarding table
- DVR software can be considered as maintaining an extension to the forwarding table that stores a distance for each destination (not just next hop)

Distributed Route Computation

Distance Vector Routing (DVR) – Updating Scheme

- When a message arrives at a switch from neighbor N
  - the switch examines each item in the message
  - The switch changes its forwarding table if the neighbor has a shorter path to some destination than the path currently being used
- Example:
  - if neighbor N advertises a path to destination D as having cost 5 and the current path through neighbor K has cost 100
    - the current next hop for D will be replaced by N
    - and the cost to reach D will be 5 plus the cost to reach N



Read the Example in Wiki:
http://en.wikipedia.org/wiki/Distance-vector_routing_protocol

Distributed Route Computation

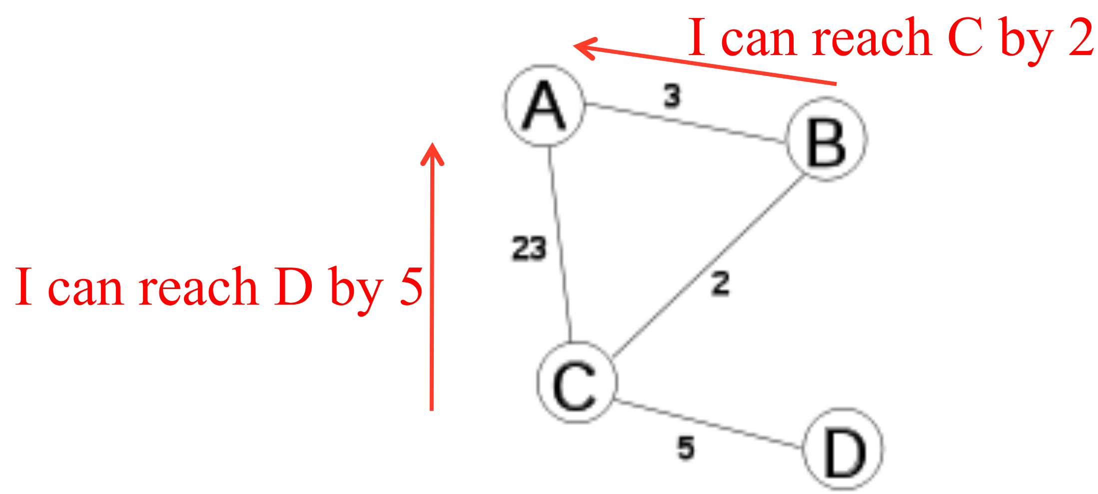
Distance Vector Routing (DVR) - Example

Example:

T=1

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D			28	

B and C send information
About nodes they can reach



Distributed Route Computation

Distance Vector Routing (DVR) - Example

Only knows about neighbors

T=0

from A	via A	via B	via C	via D
to A				
to B		3		
to C			23	
to D				

T=1

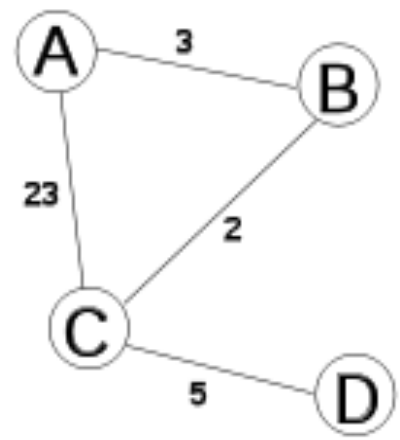
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D			28	

B and C send information
About nodes they can reach

T=2

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

B says I can also read D



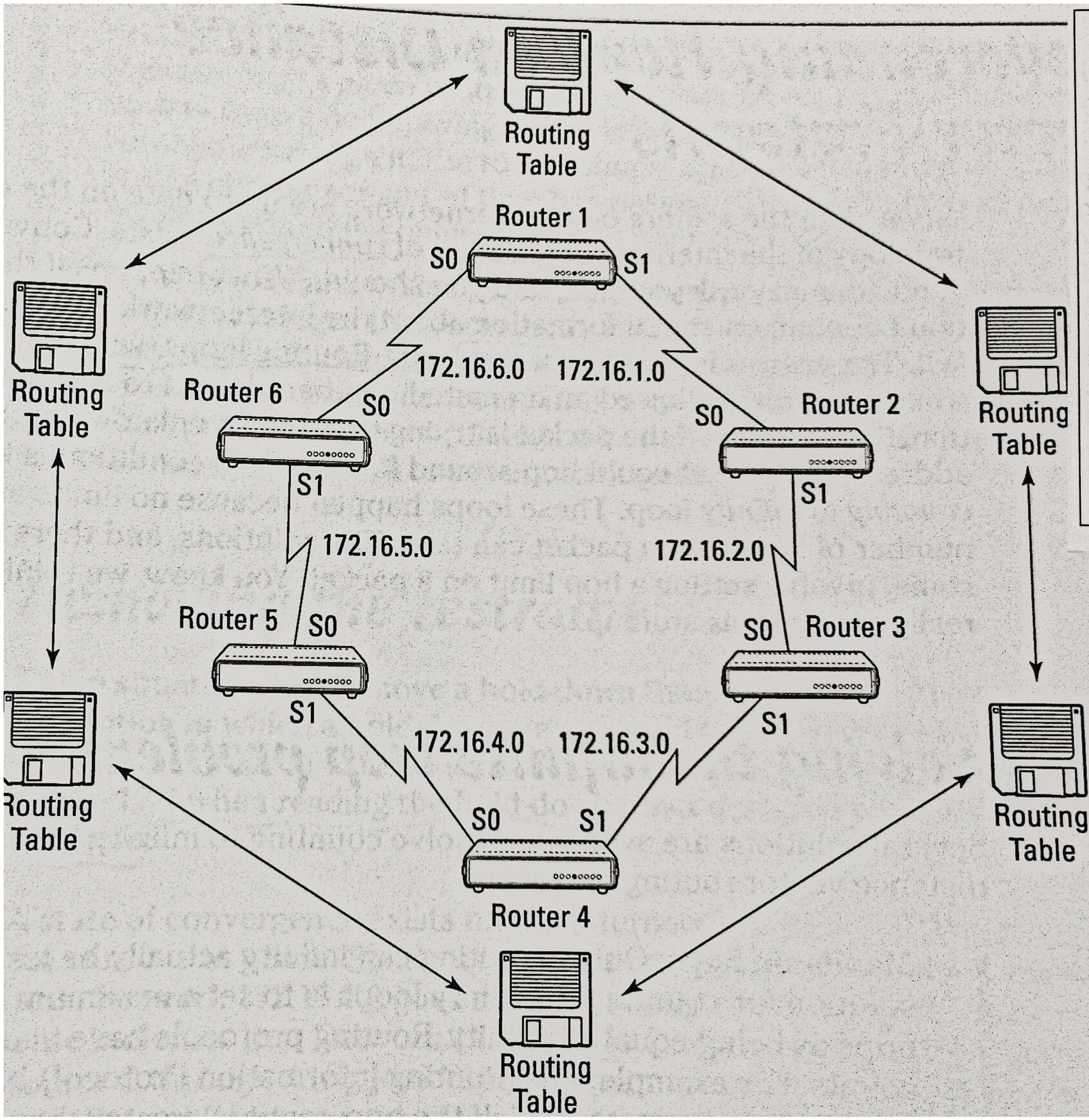
T=3

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

No new information is received.
A calculates all the shortest paths
→Forwarding (routing) table is finalized

Distributed Route Computation

Distance Vector Routing (DVR) – Sample Routing Table



Router 1

Network	Port	Disc
172.16.6.0	S0	0
172.16.1.0	S1	0
172.16.5.0	S0	1
172.16.2.0	S1	1
172.16.4.0	S0	2
172.16.3.0	S1	2

Router 2

Network	Port	Disc
172.16.1.0	S0	0
172.16.2.0	S1	0
172.16.6.0	S0	1
172.16.3.0	S1	1
172.16.5.0	S0	2
172.16.4.0	S1	2

Router 3

Network	Port	Disc
172.16.2.0	S0	0
172.16.3.0	S1	0
172.16.1.0	S0	1
172.16.4.0	S1	1
172.16.6.0	S0	2
172.16.5.0	S1	2

Router 4

Network	Port	Disc
172.16.4.0	S0	0
172.16.3.0	S1	0
172.16.5.0	S0	1
172.16.2.0	S1	1
172.16.6.0	S0	2
172.16.1.0	S1	2

Router 5

Network	Port	Disc
172.16.5.0	S0	0
172.16.4.0	S1	0
172.16.6.0	S0	1
172.16.3.0	S1	1
172.16.1.0	S0	2
172.16.2.0	S1	2

Router 6

Network	Port	Disc
172.16.6.0	S0	0
172.16.5.0	S1	0
172.16.1.0	S0	1
172.16.4.0	S1	1
172.16.2.0	S0	2
172.16.3 .0	S1	2

Algorithm 18.3

Distance-vector algorithm for route computation

Algorithm 18.3

Given:

A local forwarding table with a distance for each entry, a distance to reach each neighbor, and an incoming DV message from a neighbor

Compute:

An updated forwarding table

Method:

```
Maintain a distance field in each forwarding table entry;
Initialize forwarding table with a single entry that has the
  destination equal to the local packet switch, the
  next-hop unused, and the distance set to zero;

Repeat forever {
  Wait for a routing message to arrive over the network
    from a neighbor;  let the sender be switch N;
  for each entry in the message {
    Let V be the destination in the entry and let D
      be the distance;
    Compute C as D plus the weight assigned to the
      link over which the message arrived;
    Examine and update the local routing table:
    if (no route exists to V) {
      add an entry to the local routing table for destination
       V with next-hop N and distance C;
    } else if (a route exists that has next-hop N) {
       replace the distance in existing route with C;
    } else if (a route exists with distance greater than C) {
       change the next-hop to N and distance to C;
    }
}
```

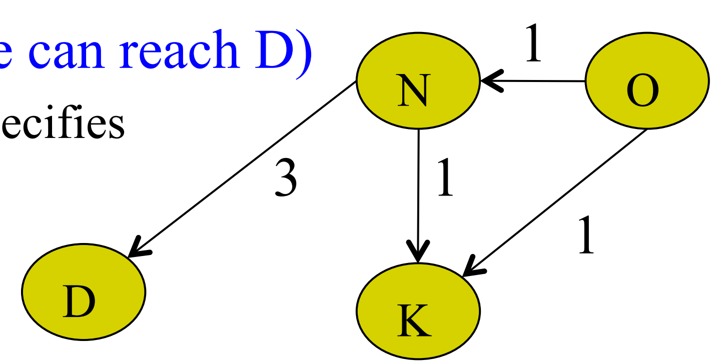
Routing Issues & Problems

- Each approach will eventually converge
  - meaning that the forwarding tables in all packet switches agree
- Shortest path calculation:
  - In theory, either LSR or DVR routing will compute shortest paths
- LSR and DVR problems:
  - LSR: For example, if LSR messages are lost, two packet switches can disagree about the shortest path
  - DVR: because a link failure can cause two or more packet switches to create a routing loop
    - in which each packet switch thinks the next packet switch in the set is the shortest path to a particular destination
    - As a result, a packet can circulate among the switches indefinitely

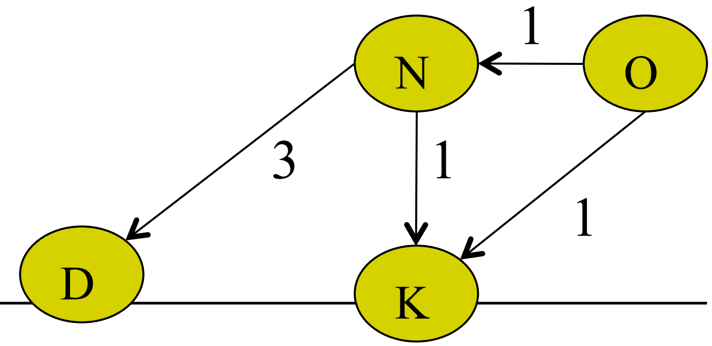
Routing Issues & Problems
- Delay & Looping

- DVR protocols can suffer from backwash (resulting in routing loop)
  - (i.e., a packet switch receives information that itself sent)
- Example of backwash:
- suppose a switch tells its neighbors (N tells O)
  - *"I can reach destination D at cost 3"*
  - If the connection leading to destination D fails
    - the switch will remove the entry for D from its forwarding table (N removes D) (or mark the entry invalid)
  - But the switch has told neighbors that a route exists (O thinks N can reach D)
- Imagine that just after the link fails (K tells N he can reach D)
  - one of the neighbors sends a DVR message that specifies
  - *"I can reach destination D at cost 5"*
- →Unfortunately
  - the message will be believed
  - and a routing loop will be created

Basic problem:
N says she can reach D
K tells N that he can reach D



Routing Problems - Possible Solutions



- Two Posible Solutions:
  - Split Horizon and Hysteresis
- Most practical routing mechanisms contain constraints and heuristics to prevent problems like routing loops
  - For example, DVR schemes employ split horizon
    - which specifies that a switch does not send information back to its origin
    - In previous example K will not tell N he can reach D!
- Furthermore, most practical routing systems introduce hysteresis
  - that prevents the software from making many changes in a short time
  - However, in a large network where many links fail and recover frequently, routing problems can occur

Shortest Path Computation in a Graph

- Once a graph has been created that corresponds to a network
  - software uses a method known as Dijkstra's Algorithm
- To find the shortest path from a source node to each of the other nodes in the graph:
  - a next-hop forwarding table is constructed during the computation of shortest paths
  - The algorithm must be run once for each source node in the graph

Shortest Path Computation in a Graph

- Dijkstra's algorithm is popular
  - because it can be used with various definitions of shortest path
  - In particular, the algorithm does not require edges in the graph to represent geographic distance. Instead, the algorithm
    - allows each edge to be assigned a nonnegative value called a weight
    - defines the distance between two nodes to be the sum of the weights along a path between the nodes

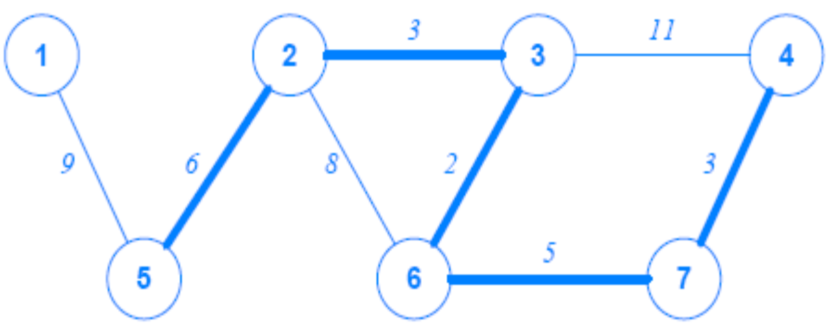
Dijkstra's Algorithm has many applications……

Dijkstra’s Algorithm

- finds shortest paths from given source node `s` to all other nodes
- by developing paths in order of increasing path length
- algorithm runs in stages (next slide)
  - each time adding node with next shortest path
- algorithm terminates when all nodes processed by algorithm (in set T)

Read this: http://www.cacr.caltech.edu/~sean/projects/stlib/html/shortest_paths/shortest_paths_dijkstra.html

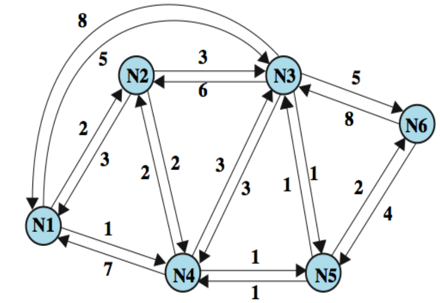
A version of Dijkstra's algorithm that computes *R, a nexthop* forwarding table, and *D, the distance to each node from* the specified source node




```
Given:
   A graph with a nonnegative weight assigned to each edge
   and a designated source node
Compute:
   The shortest distance from the source node to each other
   node and a next-hop routing table
Method:
  Initialize set S to contain all nodes except the source node;
  Initialize array D so that D[v] is the weight of the edge from the
     source to v if such an edge exists, and infinity otherwise;
  Initialize entries of R so that R[v] is assigned v if an
     edge exists from the source to v, and zero otherwise;

  while (set S is not empty) {
     choose a node u from S such that D[u] is minimum;
     if (D[u] is infinity) {
        error: no path exists to nodes in S; quit;
     }
     delete u from set S;
     for each node v such that (u,v) is an edge {
        if (v is still in S) {
           c = D[u] + weight(u,v);
           if (c < D[v]) {
              R[v] = R[u];
              D[v] = c;
           }
        }
     }
  }
```

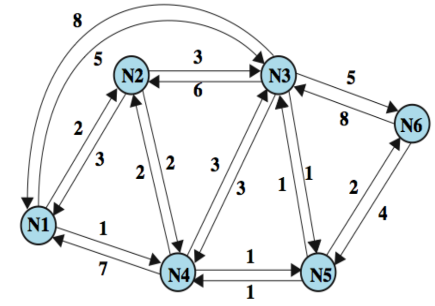
http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/dijkstra/DijkstraApp.shtml?demo1



Dijkstra's Algorithm Example

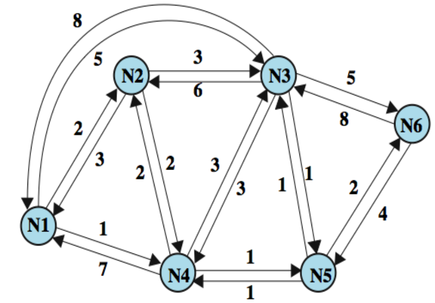
Iter	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path

Dijkstra’s Algorithm Example



Iter	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
2	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
4	{1, 2, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

Dijkstra's Algorithm Example



We want to know the distances from Node 1

Iter	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
2	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
4	{1, 2, 4, 5}					1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

We picked Node 2 because L(2) has not changed! Otherwise it will be random

Shortest path from Node 1 to all other nodes

It is possible some nodes are not reachable

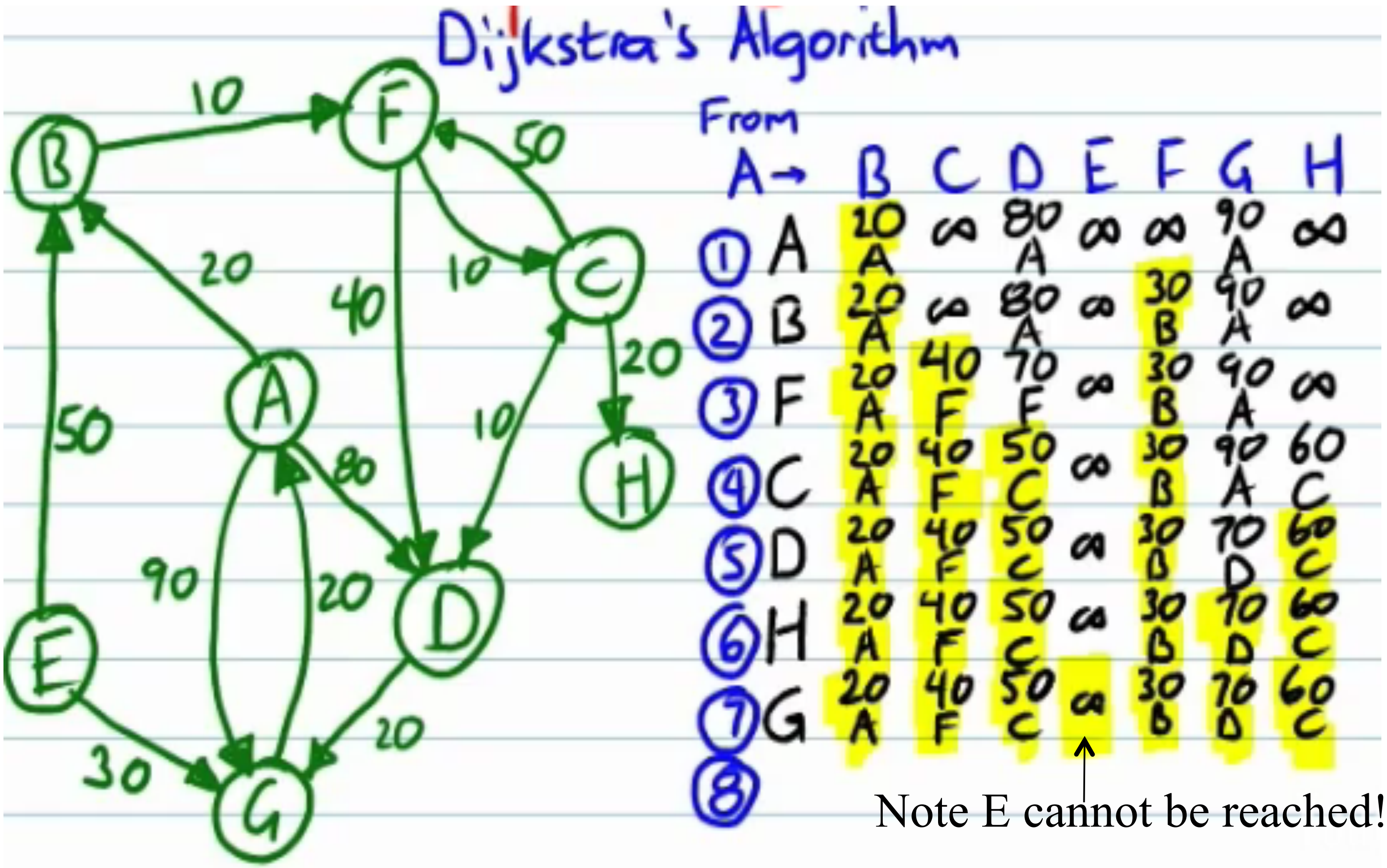
1→3 (1-4-5-3)

1→4 (1-4)

1→5 (1-4-5)

Note: We found the shortest paths from node 1 to all other nodes. Note that we performed the algorithm nx(n-1) times. To get all the shortest paths the complexity of the algorithm will be n^3

Another Example (Watch YouTube)



Note E cannot be reached!

Watch:
http://wn.com/Dijkstra%27s_algorithm - **Note: Assume A→ is 10**

Netstat – r and ipconfig

```
Windows IP Configuration


Ethernet adapter Local Area Connection:

        Media State . . . . . . . . . . . : Media disconnected

Ethernet adapter Wireless Network Connection:

        Connection-specific DNS Suffix  . : sbx10339.cotatca.wayport.net
        IP Address. . . . . . . . . . . . : 192.168.5.177
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.5.1

C:\Documents and Settings\farid>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 21 70 d9 4a 3a ...... Intel(R) 82567LM Gigabit Network Connection - P
cket Scheduler Miniport
0x10004 ...00 21 5d da 7a 26 ...... Intel(R) WiFi Link 5100 AGN - Packet Schedu
er Miniport
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.5.1   192.168.5.177     25
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1      1
      192.168.5.0    255.255.255.0    192.168.5.177   192.168.5.177     25
    192.168.5.177  255.255.255.255        127.0.0.1       127.0.0.1     25
    192.168.5.255  255.255.255.255    192.168.5.177   192.168.5.177     25
        224.0.0.0        240.0.0.0    192.168.5.177   192.168.5.177     25
  255.255.255.255  255.255.255.255    192.168.5.177               2      1
  255.255.255.255  255.255.255.255    192.168.5.177   192.168.5.177      1
Default Gateway:       192.168.5.1
===========================================================================
Persistent Routes:
  None
```

Read about NetStat:
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/netstat.mspx?mfr=true

References

- A very nice applet to try shortest path algorithms: http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/dijkstra/DijkstraApp.shtml?demo1
- A good resource for code download: http://en.literateprograms.org/Dijkstra's_algorithm_(Java)
- Good online slides to learn more about routing and routing algorithms: http://www.cs.umd.edu/~shankar/417-F01/Slides/chapter4a-aus/sld010.htm
- Fishnet: http://cseweb.ucsd.edu/classes/fa09/cse123/123f09_Proj2.pdf

References

- Chapter 12 of Stalling
- Chapter 18 Comer
- Chapter 11 Cisco Networking (for Dummies)