BLM6196 COMPUTER NETWORKS AND COMMUNICATION PROTOCOLS

Prof. Dr. Hasan Hüseyin BALIK

(7th Week)

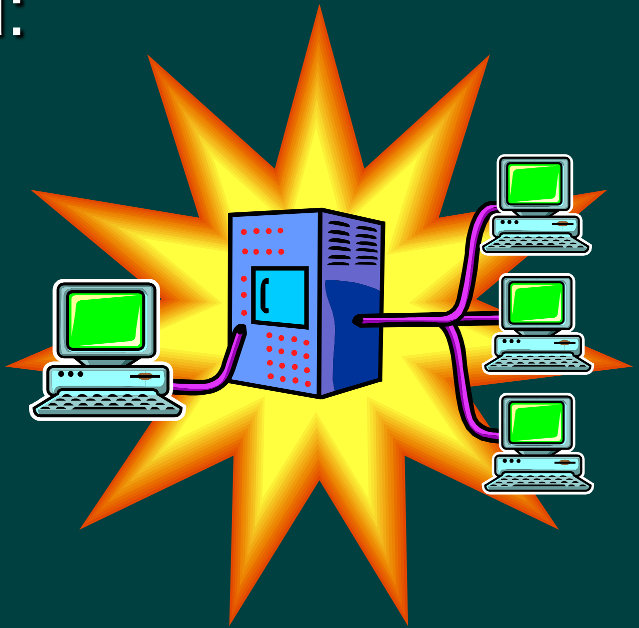
7. Routing

7.Outline

- Routing in Packet-Switching Networks
- Examples: Routing in ARPANET
- Internet Routing Protocols
- Least-Cost Algorithms

Routing in Packet Switching Networks

- Key design issue for (packet) switched networks
- Select route across network between end nodes
- Characteristics required:
  - Correctness
  - Simplicity
  - Robustness
  - Stability
  - Fairness
  - Optimality
  - Efficiency

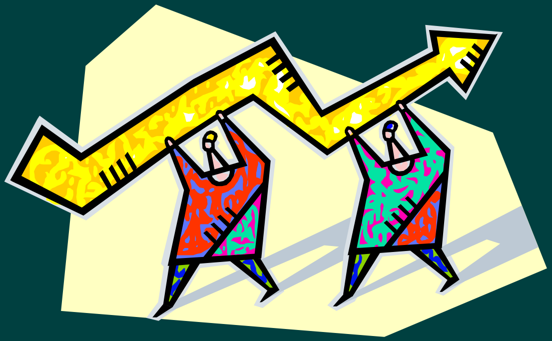


Elements of Routing Techniques for Packet-Switching Networks

Performance Criteria	Network Information Source
Number of hops	None
Cost	Local
Delay	Adjacent node
Throughput	Nodes along route
	All nodes
Decision Time	
Packet (datagram)	**Network Information Update Timing**
Session (virtual circuit)	Continuous
	Periodic
Decision Place	Major load change
Each node (distributed)	Topology change
Central node (centralized)	
Originating node (source)	

Performance Criteria

- Used for selection of route
- Simplest is to choose “minimum hop”
- Can be generalized as “least cost” routing
- Because “least cost” is more flexible it is more common than “minimum hop”



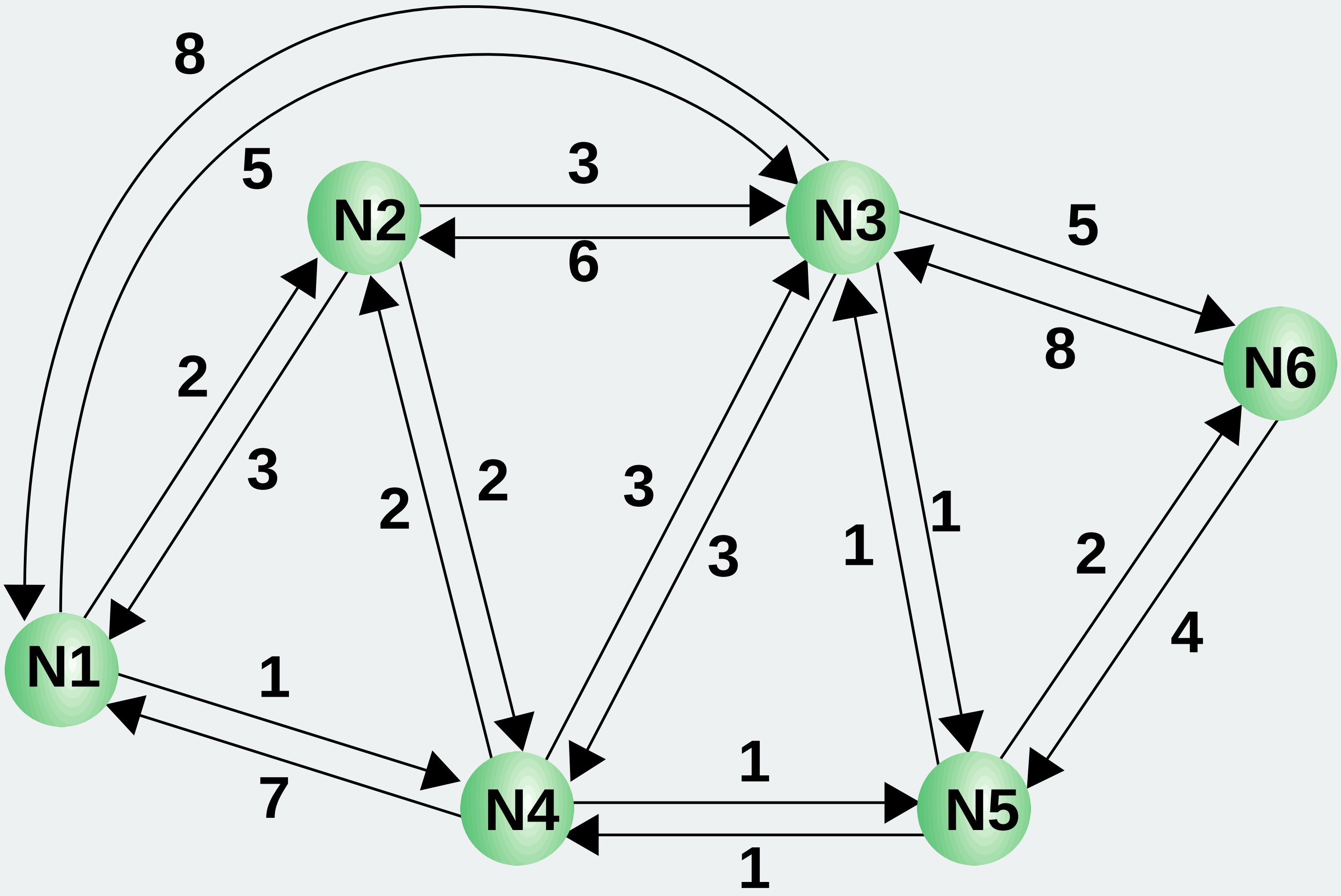


Figure 19.1 Example Network Configuration

Decision Time and Place

Decision time

- Packet or virtual circuit basis
- Fixed or dynamically changing

Decision place

- Distributed - made by each node
  - More complex, but more robust
- Centralized – made by a designated node
- Source – made by source station

Network Information Source and Update Timing

- Routing decisions usually based on knowledge of network, traffic load, and link cost
  - Distributed routing
    - Using local knowledge, information from adjacent nodes, information from all nodes on a potential route
  - Central routing
    - Collect information from all nodes

Issue of update timing

- Depends on routing strategy
- Fixed - never updated
- Adaptive - regular updates

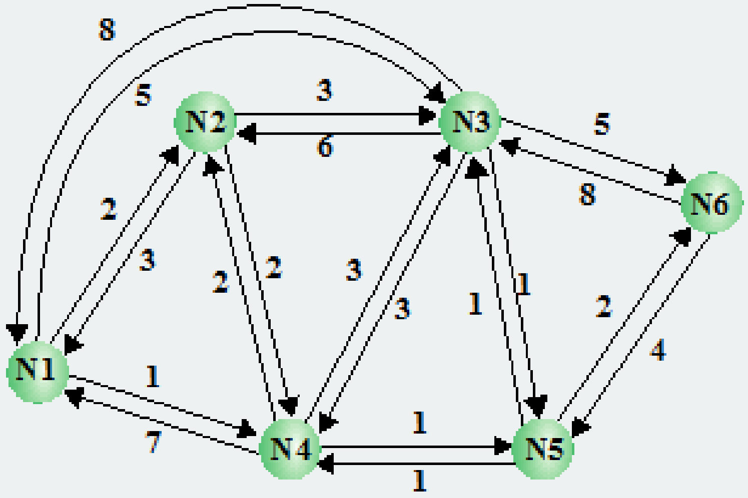
Routing Strategies - Fixed Routing

- Use a single permanent route for each source to destination pair of nodes
- Determined using a least cost algorithm
- Route is fixed
  - Until a change in network topology
  - Based on expected traffic or capacity
- Advantage is simplicity
- Disadvantage is lack of flexibility
  - Does not react to network failure or congestion

CENTRAL ROUTING DIRECTORY

From Node

To 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

Figure 19.2 Fixed Routing (using Figure 19.1)

Routing Strategies - Flooding

- Packet sent by node to every neighbor
- Eventually multiple copies arrive at destination
- No network information required
- Each packet is uniquely numbered so duplicates can be discarded
- Need to limit incessant retransmission of packets
  - Nodes can remember identity of packets retransmitted
  - Can include a hop count in packets

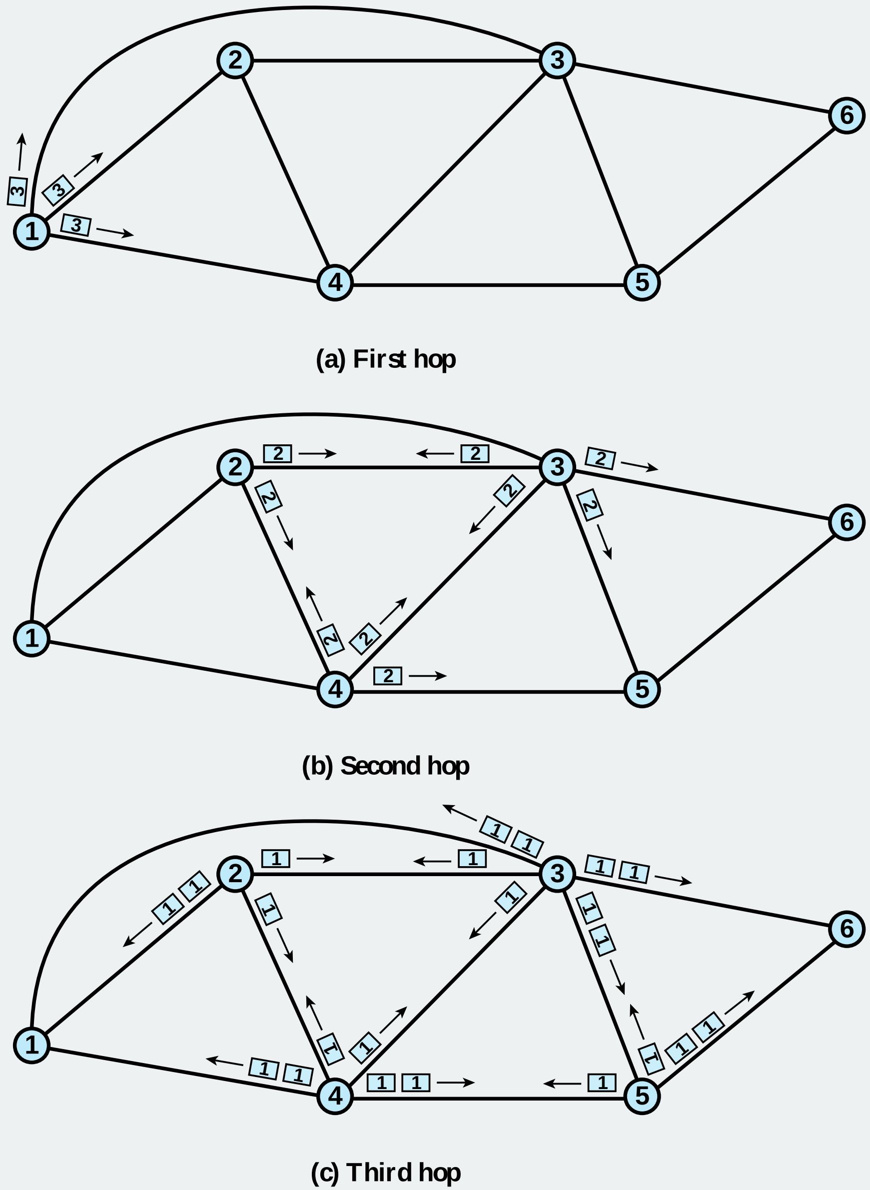


Figure 19.3 Flooding Example (hop count = 3)

Properties of Flooding

- All possible routes are tried
  - Highly robust
  - Can be used to send emergency messages
- At least one packet will have taken minimum hop route
- Nodes directly or indirectly connected to source are visited
- Disadvantages:
  - High traffic load generated
  - Security concerns

Routing Strategies - Random Routing

- Simplicity of flooding with much less traffic load
- Node selects one outgoing path for retransmission of incoming packet
- Selection can be random or round robin
- A refinement is to select outgoing path based on probability calculation
- No network information needed
- Random route is typically neither least cost nor minimum hop

Routing Strategies - Adaptive Routing

- Used by almost all packet switching networks
- Routing decisions change as conditions on the network change due to failure or congestion
- Requires information about network

Disadvantages:	Decisions more complex
	Tradeoff between quality of network information and overhead
	Reacting too quickly can cause oscillation
	Reacting too slowly means information may be irrelevant

Adaptive Routing Advantages

Improved performance

Aid in congestion control

These benefits depend on the soundness of the design and nature of the load

Classification of Adaptive Routing Strategies

- A convenient way to classify is on the basis of information source

Local (isolated)
- Route to outgoing link with shortest queue
- Can include bias for each destination
- Rarely used - does not make use of available information

Adjacent nodes
- Takes advantage of delay and outage information
- Distributed or centralized

All nodes
- Like adjacent

Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0

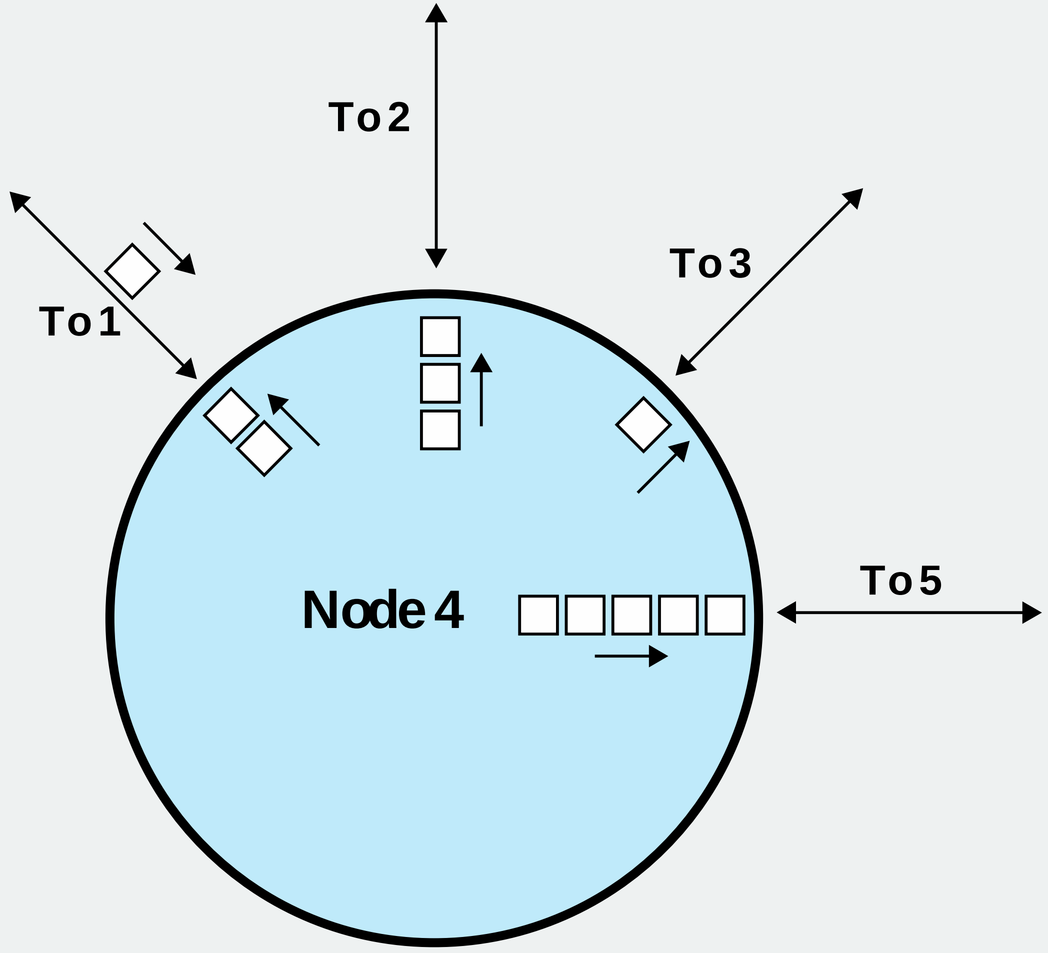


Figure 19.4 Example of Isolated Adaptive Routing

ARPANET Routing Strategies 1st Generation

Distance Vector Routing

- **1969**
- Distributed adaptive using estimated delay
  - Queue length used as estimate of delay
- Version of Bellman-Ford algorithm
- Node exchanges delay vector with neighbors
- Update routing table based on incoming information
- Doesn't consider line speed, just queue length and responds slowly to congestion

Destination	Delay	Next node
1	0	—
2	2	2
3	5	3
4	1	4
5	6	3
6	8	3
	D_1	S_1

(a) Node 1's Routing table before update

3	7	5
0	4	2
3	0	2
2	2	0
3	1	1
5	3	3
D_2	D_3	D_4

(b) Delay vectors sent to node 1 from neighbor nodes

Destination	Delay	Next node
1	0	—
2	2	2
3	3	4
4	1	4
5	2	4
6	4	4

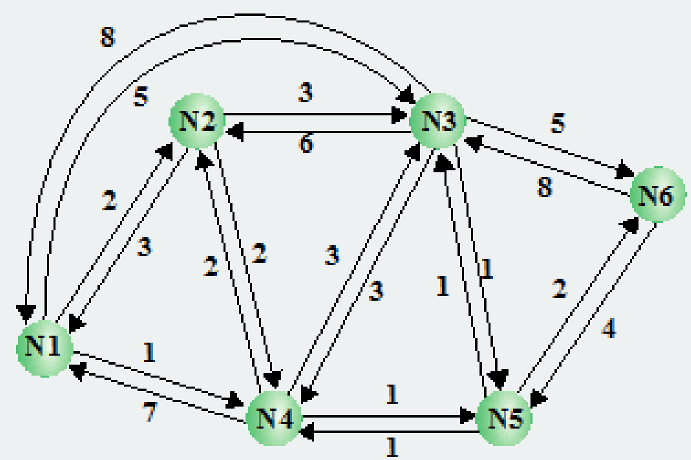
$l_{1,2} = 2$

$l_{1,3} = 5$

$l_{1,4} = 1$

(c) Node 1's routing table after update and link costs used in update

Figure 19.5 Original ARPANET Routing Algorithm



ARPANET Routing Strategies 2nd Generation

Link-State Routing

- **1979**
- Distributed adaptive using delay criterion
  - Using timestamps of arrival, departure and ACK times
- Re-computes average delays every 10 seconds
- Any changes are flooded to all other nodes
- Re-computes routing using Dijkstra’s algorithm
- Good under light and medium loads
- Under heavy loads, little correlation between reported delays and those experienced

ARPANET Routing Strategies 3rd Generation

- **1987**
- Link cost calculation changed
  - Damp routing oscillations
  - Reduce routing overhead
- Measure average delay over last 10 seconds and transform into link utilization estimate
- Normalize this based on current value and previous results
- Set link cost as function of average utilization

Internet Routing Protocols

- Routers are responsible for receiving and forwarding packets through the interconnected set of networks
  - Each router makes routing decisions based on knowledge of the topology and traffic/delay conditions of the internet
  - Routers exchange routing information using a special routing protocol
- Two concepts in considering the routing function:
  - Routing information
    - Information about the topology and delays of the internet
  - Routing algorithm
    - The algorithm used to make a routing decision for a particular datagram, based on current routing information

Autonomous Systems (AS)

- Exhibits the following characteristics:
  - An AS is a set of routers and networks managed by a single organization
  - An AS consists of a group of routers exchanging information via a common routing protocol
  - Except in times of failure, an AS is connected (in a graph-theoretic sense); That is there is a path between any pair of nodes

Interior Router Protocol (IRP)

- A shared routing protocol, which we shall refer to as an interior router protocol (IRP), passes routing information between routers within an AS.
- This flexibility allows IRPs to be custom tailored to specific applications and requirements

Exterior Router Protocol (ERP)

- Protocol used to pass routing information between routers in different ASs
- Will need to pass less information than an IRP for the following reason:
  - If a datagram is to be transferred from a host in one AS to a host in another AS, a router in the first system need only determine the target AS and devise a route to get into that target system
  - Once the datagram enters the target AS, the routers within that system can cooperate to deliver the datagram
  - The ERP is not concerned with, and does not know about, the details of the route

Examples

- Border Gateway Protocol (BGP)
- Open Shortest Path First (OSPF)

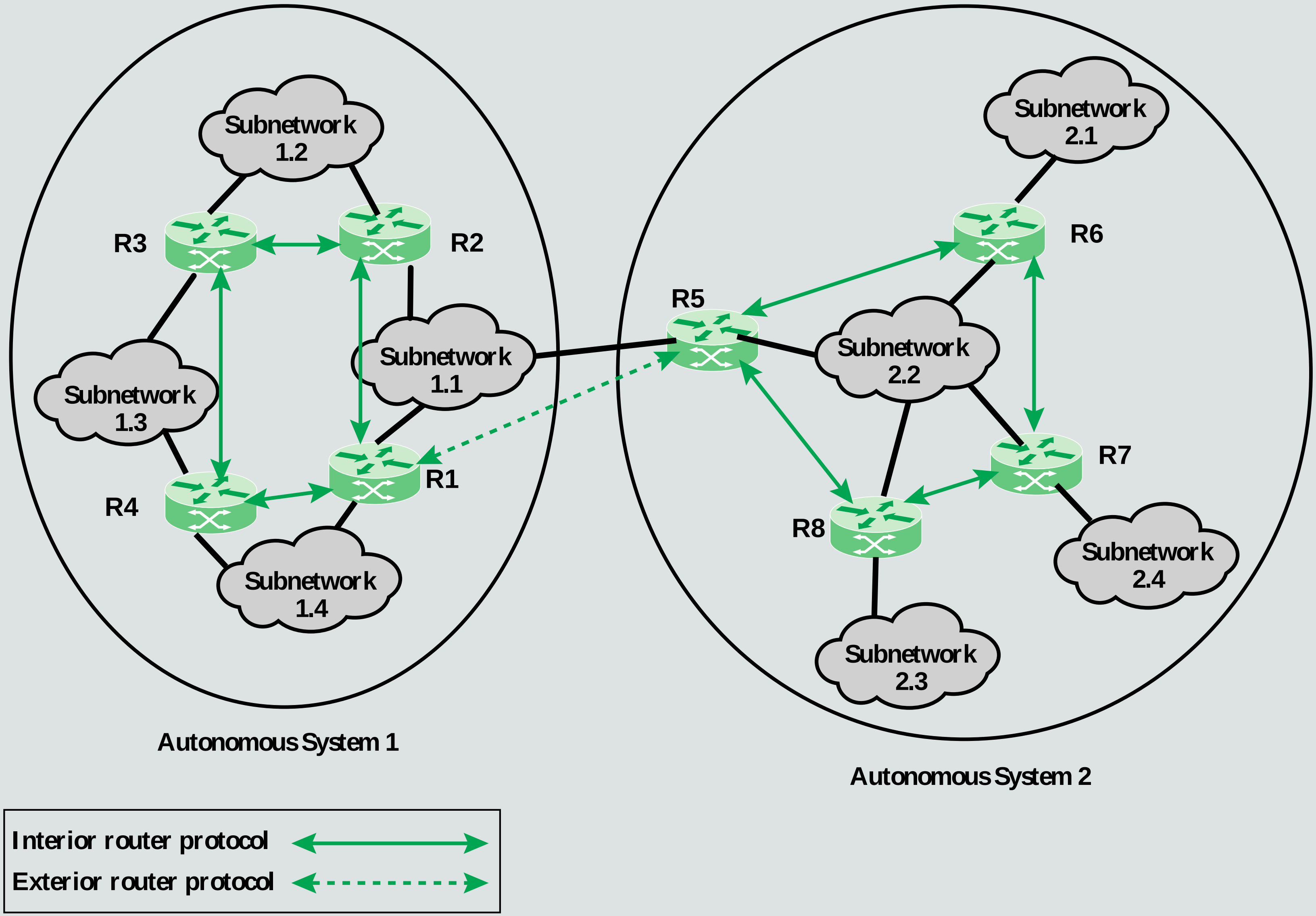


Figure 19.9 Application of Exterior and Interior Routing Protocols

Approaches to Routing

- Internet routing protocols employ one of three approaches to gathering and using routing information:

Distance-vector routing

Path-vector routing

Link-state routing

Distance-Vector Routing

- Requires that each node exchange information with its neighboring nodes
  - Two nodes are said to be neighbors if they are both directly connected to the same network
- Used in the first-generation routing algorithm for ARPANET
- Each node maintains a vector of link costs for each directly attached network and distance and next-hop vectors for each destination
- Routing Information Protocol (RIP) uses this approach

Path-Vector Routing

- Alternative to dispense with routing metrics and simply provide information about which networks can be reached by a given router and the ASs visited in order to reach the destination network by this route

- Differs from a distance-vector algorithm in two respects:
  - The path-vector approach does not include a distance or cost estimate
  - Each block of routing information lists all of the ASs visited in order to reach the destination network by this route

Link-State Routing

- Designed to overcome the drawbacks of distance-vector routing
- When a router is initialized, it determines the link cost on each of its network interfaces
- The router then advertises this set of link costs to all other routers in the internet topology, not just neighboring routers
- From then on, the router monitors its link costs
- Whenever there is a significant change the router again advertises its set of link costs to all other routers in the configuration
- The OSPF protocol is an example
- The second-generation routing algorithm for ARPANET also uses this approach

Border Gateway Protocol (BGP)

- Was developed for use in conjunction with internets that employ the TCP/IP suite
- Has become the preferred exterior router protocol for the Internet
- Designed to allow routers in different autonomous systems to cooperate in the exchange of routing information
- Protocol operates in terms of messages, which are sent over TCP connections
- Current version is known as BGP-4 (RFC 4271)

Three functional procedures:

Neighbor acquisition
Neighbor reachability
Network reachability

Neighbor Acquisition

- Occurs when two neighboring routers in different autonomous systems agree to exchange routing information regularly
- Two routers send Open messages to each other after a TCP connection is established
  - If each router accepts the request, it returns a Keepalive message in response
- Protocol does not address the issue of how one router knows the address or even the existence of another router nor how it decides that it needs to exchange routing information with that particular router

Open Shortest Path First (OSPF) Protocol

- RFC 2328
- Used as the interior router protocol in TCP/IP networks
- Computes a route through the internet that incurs the least cost based on a user-configurable metric of cost
- Is able to equalize loads over multiple equal-cost paths

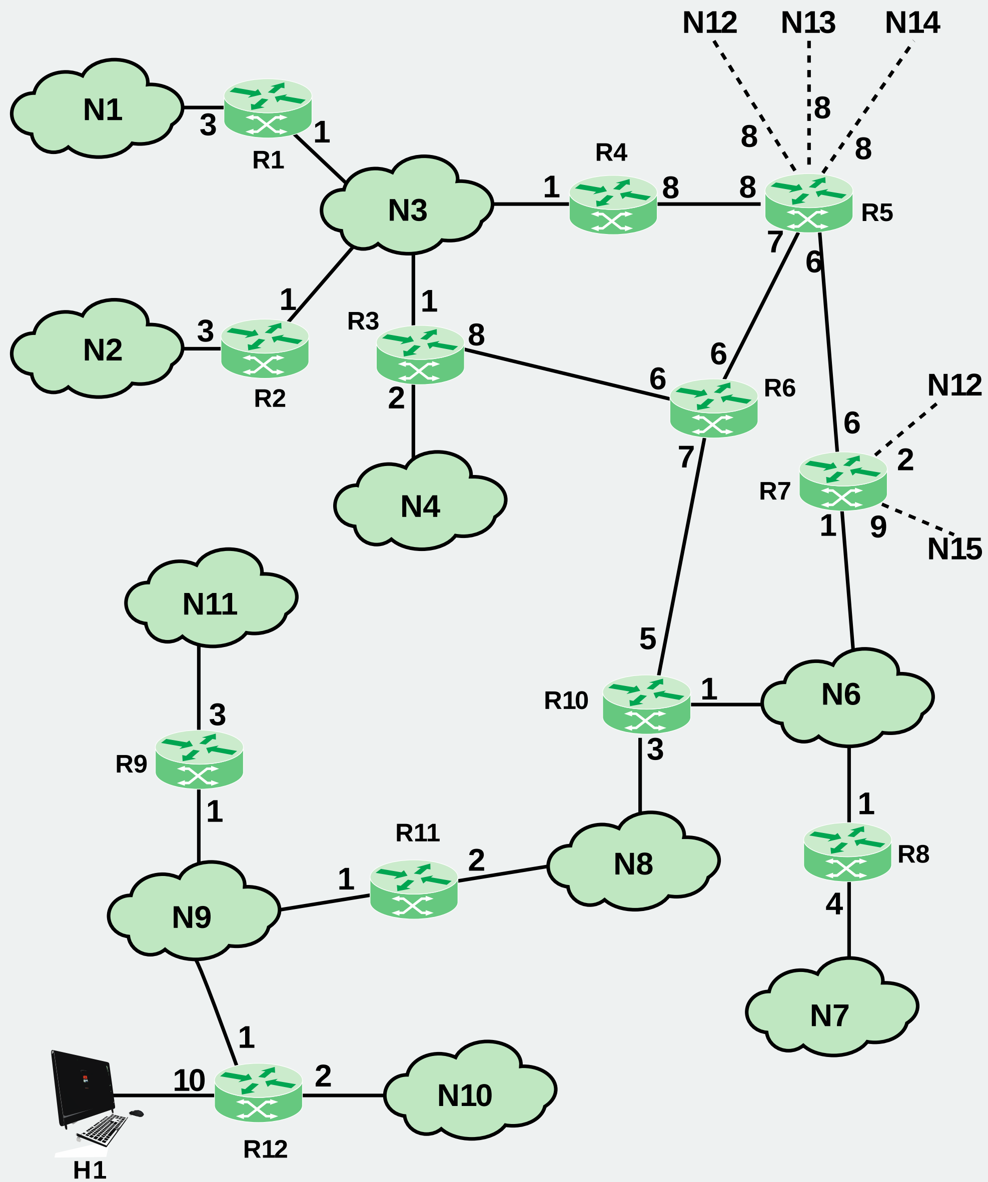


Figure 19.11 A Sample Autonomous System

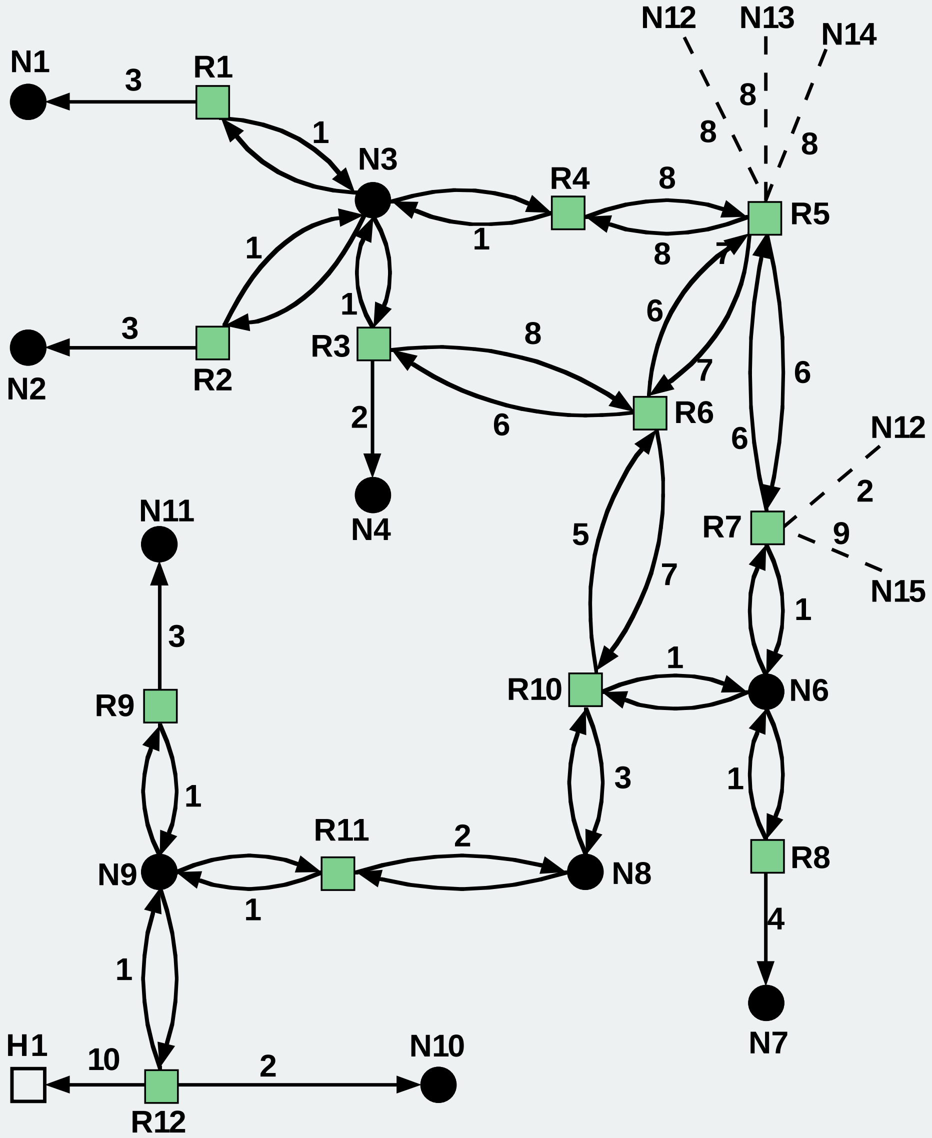


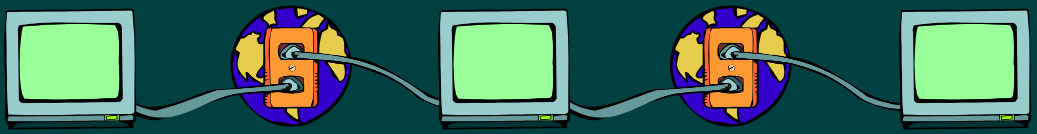
Figure 19.12 Directed Graph of Autonomous System of Figure 19.11

Destination	Next Hop	Distance
N1	R3	10
N2	R3	10
N3	R3	7
N4	R3	8
N6	R10	8
N7	R10	12
N8	R10	10
N9	R10	11
N10	R10	13
N11	R10	14
H1	R10	21
R5	R5	6
R7	R10	8
N12	R10	10
N13	R5	14
N14	R5	14
N15	R10	17

Routing Table for R6

Dijkstra's Algorithm

- Finds shortest paths from given source nodes to all other nodes
- Develop paths in order of increasing path length
- Algorithm runs in stages
  - Each time adding node with next shortest path
- Algorithm terminates when all nodes have been added to *T*



Dijkstra's Algorithm Method

Step 1 [Initialization]

$T = \{s\}$ Set of nodes so far incorporated	$L(n) = w(s, n)$ for $n \neq s$	Initial path costs to neighboring nodes are simply link costs

Step 2 [Get Next Node]

Find neighboring node not in T with least-cost path from s	Incorporate node into T	Also incorporate the edge that is incident on that node and a node in T that contributes to the path

Step 3 [Update Least-Cost Paths]

$L(n) = \min[L(n), L(x) + w(x, n)]$ for all $n \notin T$	If latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n

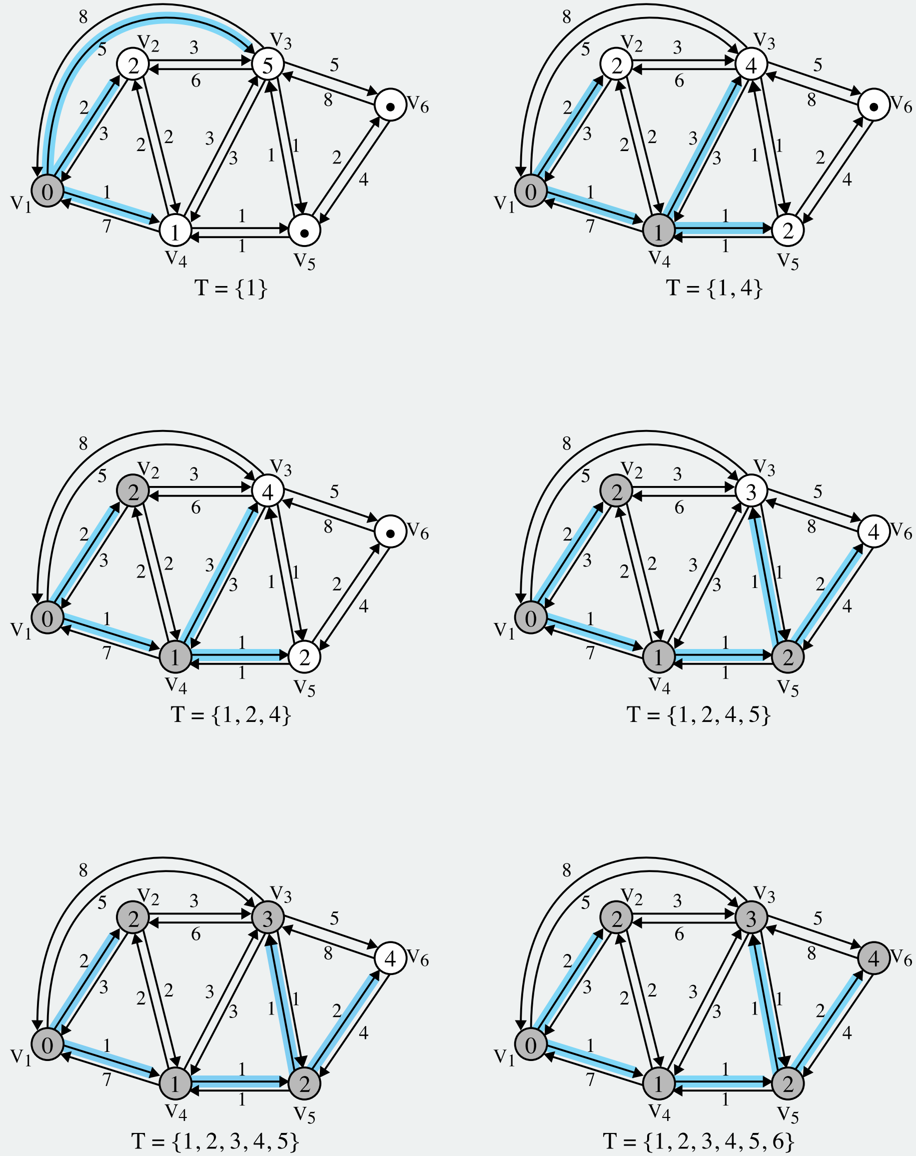
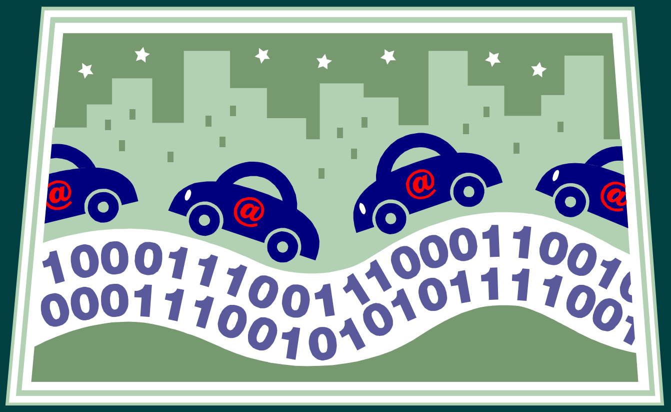


Figure 19.14 Dijkstra's Algorithm Applied to Graph of Figure 19.1

Bellman-Ford Algorithm

- Find shortest paths from given node subject to constraint that paths contain at most one link
- Find the shortest paths with a constraint of paths of at most two links
- Proceeds in stages



Bellman-Ford Algorithm

Step1 [Initialization]

$L_0(n) = \infty$, for all $n \neq s$

$L_h(s) = 0$, for all h

Step 2 [Update]

For each successive $h \geq 0$

For each $n \neq s$, compute:

$L_{h+1}(n) = \min_j [L_h(j) + w(j,n)]$

Connect n with predecessor node j that gives min

Eliminate any connection of n with different predecessor node formed during an earlier iteration

Path from s to n terminates with link from j to n

Example of Least-Cost Routing Algorithms (using Figure 19.1)

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	¥	—	¥	—	¥	—	¥	—	¥	—
1	2	1 - 2	5	1 - 3	1	1 - 4	¥	—	¥	—
2	2	1 - 2	4	1 - 4 - 3	1	1 - 4	2	1 - 4 - 5	10	1- 3 - 6
3	2	1 - 2	3	1 - 4 - 5 - 3	1	1 - 4	2	1 - 4 - 5	4	1 - 4 - 5 - 6
4	2	1 - 2	3	1 - 4 - 5 - 3	1	1 - 4	2	1 - 4 - 5	4	1 - 4 - 5 - 6

Bellman-Ford Algorithm (s = 1)

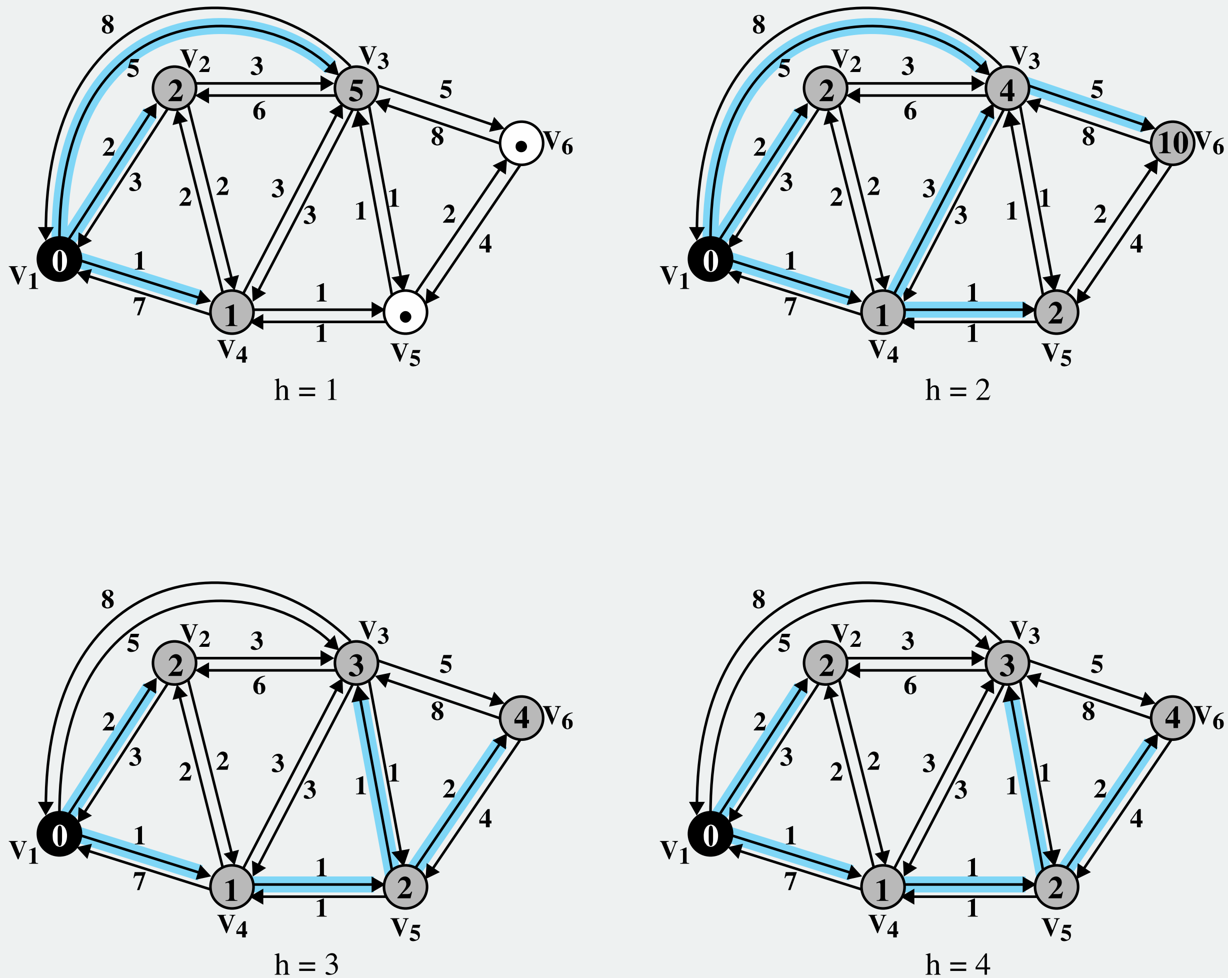


Figure 19.15 Bellman-Ford Algorithm Applied to Graph of Figure 19.1

Comparison

- Bellman-Ford
  - Calculation for node n needs link cost to neighboring nodes plus total cost to each neighbor from s
  - Each node can maintain set of costs and paths for every other node
  - Can exchange information with direct neighbors
  - Can update costs and paths based on information from neighbors and knowledge of link costs
- Dijkstra
  - Each node needs complete topology
  - Must know link costs of all links in network
  - Must exchange information with all other nodes

Evaluation

Dependent on

- Processing time of algorithms
- Amount of information required from other nodes

Implementation specific

Both converge under static topology and costs

Both converge to same solution

If link costs change, algorithms attempt to catch up

If link costs depend on traffic, which depends on routes chosen, may have feedback instability