MODULE 11: The Internet Protocol Suite

Lecture 11.5
Network Layer and Routing

Prepared By:

- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering

Virginia Tech

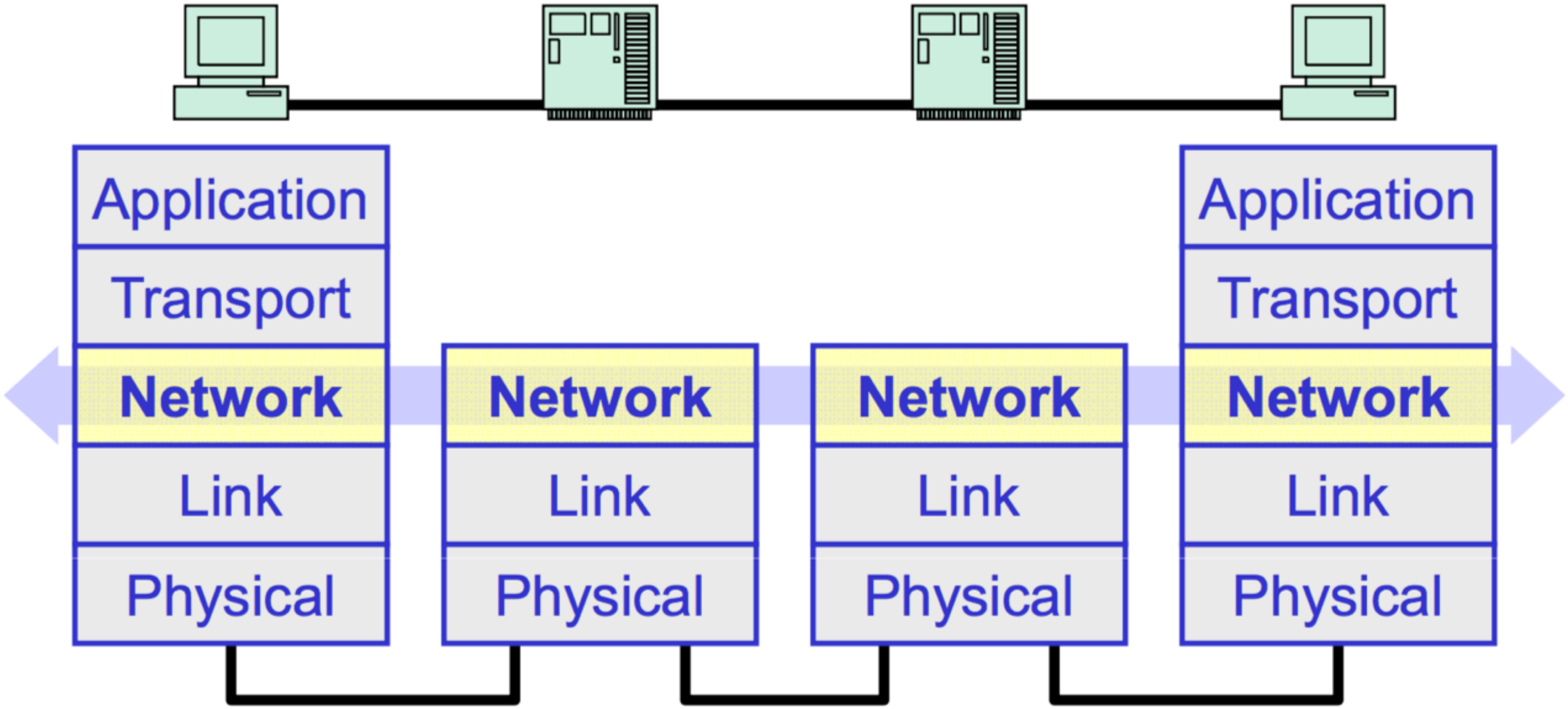
Lecture 11.5 Objectives

- Describe the services provided by the network layer
- Compare and contrast circuit switching, datagram, and virtual circuit switching
- Given a network topology and datagram switching, build a routing table at each router
- Discuss the basic objectives of routing and of least-cost routing
- Describe the key features of distance vector and link state routing algorithms

Lecture Objectives (cont'd.)

- Describe the basic operation of the asynchronous distributed Bellman-Ford algorithm as used for distance vector (DV) routing
- Describe the operation of Dijkstra's algorithm as used for link state (LS) routing
- Discuss the motivation for and concept of hierarchical routing

Routing and the Network Layer

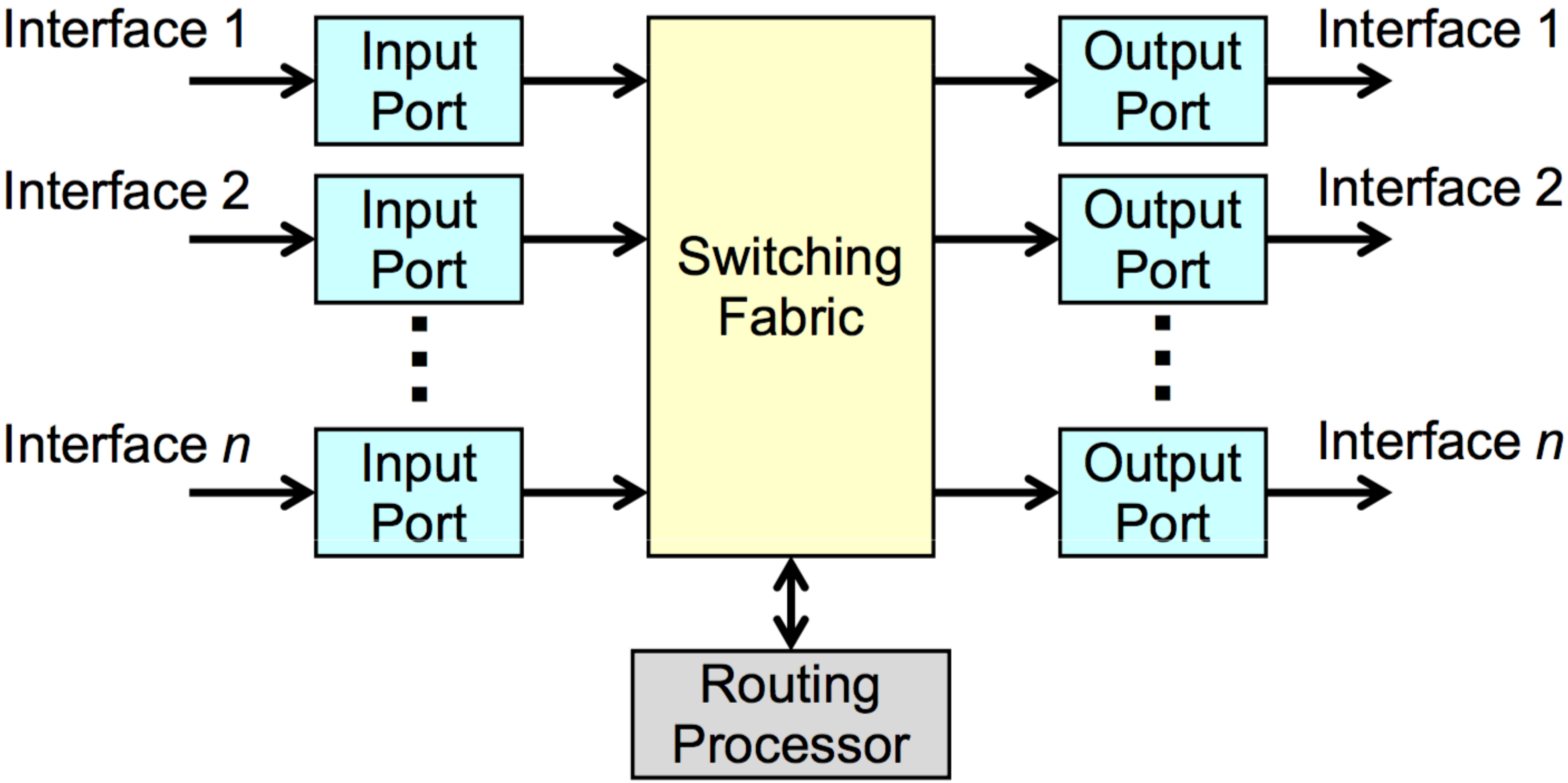


The network layer is responsible for moving packets through the entire network – routing is a key function

Network Layer Services

- Routing
  - Determining the route for a packet to take from source to destination
  - Relatively hard to do well – it is a large-scale, asynchronous, distributed systems problem
- Forwarding
  - Moving a packet from an input port to an output port
  - Relatively easy – it is a local problem, although achieving high speed requires a good design

Basic Router Components



Three Basic Forms of Routing

Circuit Switching	Datagram	Virtual Circuits
Dedicated path	No dedicated path	No dedicated path
Continuous transmission of data	Data encapsulated in packets	Data encapsulated in packets
Path established for entire call	Route established for each datagram	Path established for entire call
Call set-up delay	No call set-up delay	Call set-up delay

Three Basic Forms of Routing (cont'd.)

Circuit Switching	Datagram	Virtual Circuits
No store-and-forward delay	Store-and-forward delay	Store-and-forward delay
Call blocking; overload should not affect existing calls	Congestion increases packet delay	Congestion may cause call blocking, increased packet delay
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call set-up	Overhead bits in each packet	Overhead bits in each packet

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

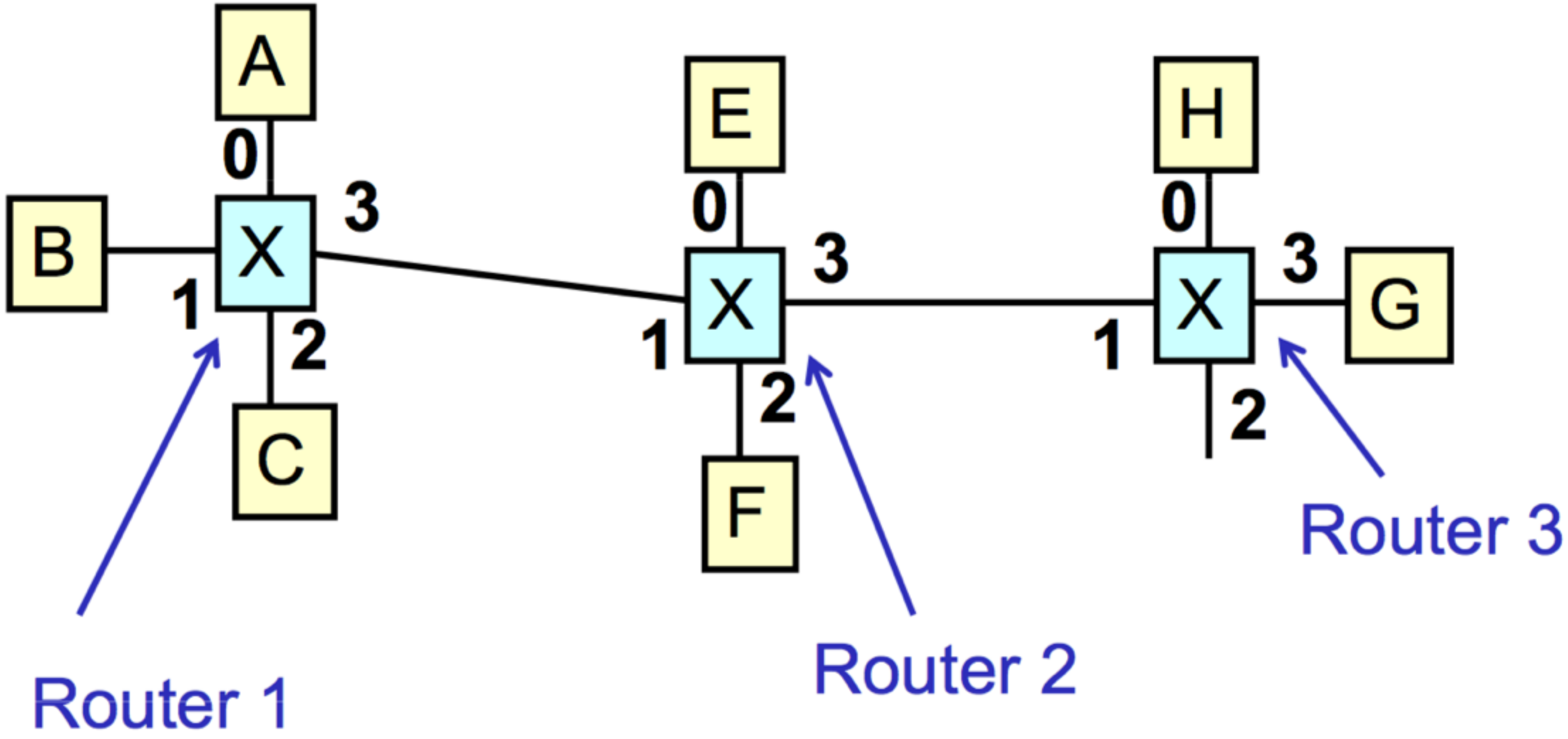
- Describe the services provided by the network layer
- Compare and contrast circuit switching, datagram, and virtual circuit switching

If you have any difficulties, please review the lecture video before continuing.

Datagram Routing

- Connectionless method
  - No set-up phase is needed prior to data transmission
- Each packet contains complete destination address
  - Address usually has a hierarchical structure
- Routing table is consulted to decide where to forward each packet
  - Routing table maps destination addresses to outbound links (creating the routing table is the hard part!)
- Datagram networks do not maintain connection-state information
- Used for routing Internet Protocol (IP) packets

Datagram Routing Example



- A-H indicate hosts (or other routers) connected to this network
- Numbers indicate ports on Routers 1-3

Example: Routing Table at Router 2

Destination	Port #
A	1 (to Router 1)
B	1 (to Router 1)
C	1 (to Router 1)
E	0 (direct)
F	2 (direct)
G	3 (to Router 3)
H	3 (to Router 3)

Note that a “destination” could be a node or a network

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Given a network topology and datagram switching, build a routing table at each router

If you have any difficulties, please review the lecture video before continuing.

Comments on Datagram Routing

- Each packet is forwarded independently and, therefore, may follow a different route
- Enables flexible response to changes in network topology, including node and link failures
- This was an important reason for the selection of this approach the ARPANET (and its persistence in the Internet today)
- Forming the routing table is hard
- Routing tables built through a distributed algorithm
- Changes in network topology and state leads to updates in the routing tables at some or all nodes

Routing Algorithms

- Routing algorithms determine the path that a packet should follow through the network (build the routing tables)
- Location of the routing algorithm
  - Centralized – at some central control center
  - Decentralized or distributed – at each router
- Information used in a distributed algorithm
  - Global – full network topology
  - Local or decentralized – just local information
- Timing
  - Static – fixed or very slowly changing network
  - Dynamic – changing network which requires updates

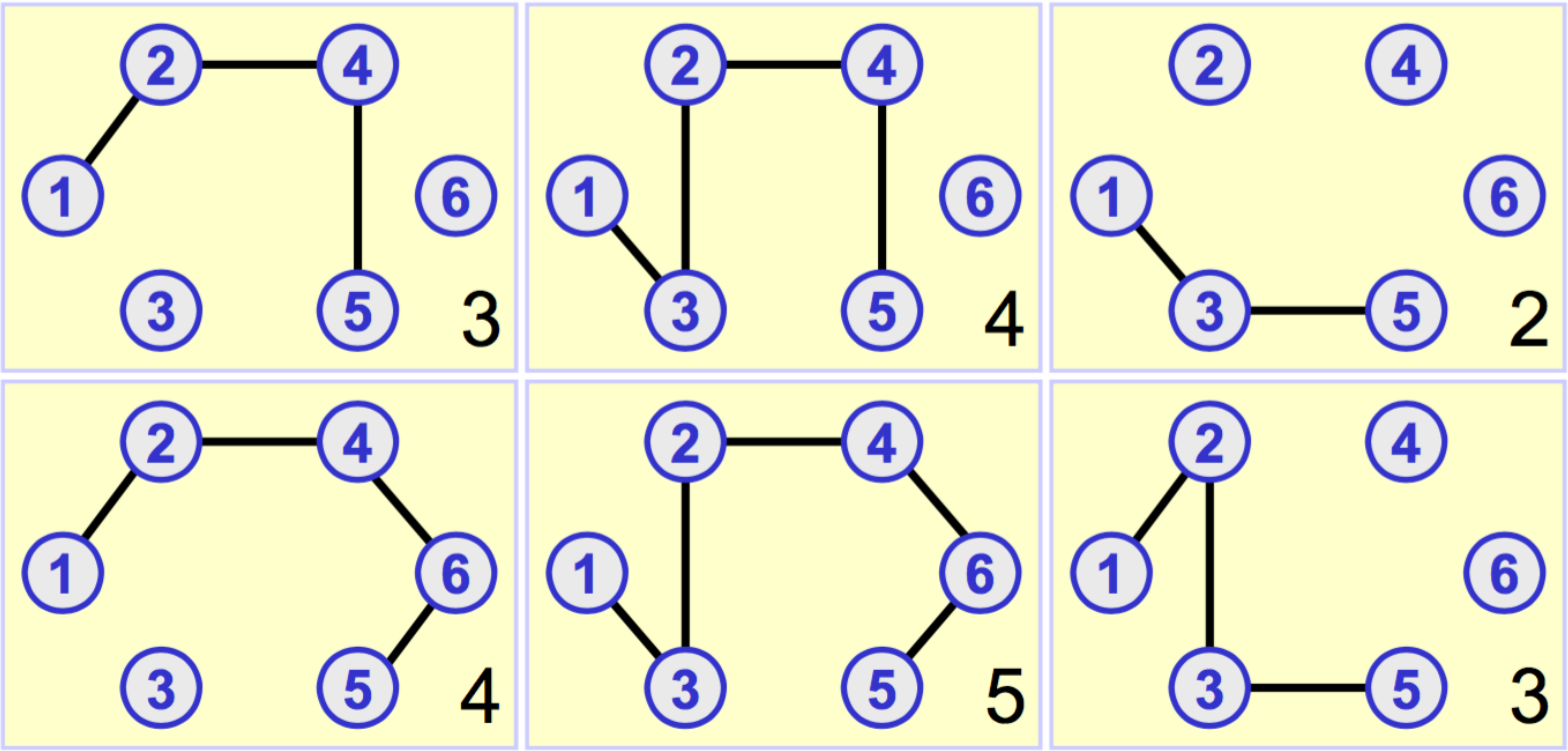
Least-Cost Routing

- A routing algorithm usually should do more than just find a path – it should find a path that is “good”
  - Minimum number of hops
  - Lowest delay on links in the path
  - Lowest cost for transmission
  - Some other metric that is to be optimized
- We can assign a “distance” or “cost” (some metric) to each link and find a path that minimizes the aggregate path “distance” or “cost”
- Policies may also be considered in practice

Variations in Path Cost

Which is best for a path from Node 1 to Node 5?

Consider number of hops (cost = 1 for each link)



Two Common Routing Schemes

- Distance vector (DV) routing
  - Relies on local or decentralized information
  - Often based on distributed, asynchronous version of the Bellman-Ford algorithm
- Link state (LS) routing
  - Relies on global information
  - Paths found by a shortest path algorithm, such as Djikstra's algorithm
- Both algorithms run in a distributed manner and can accommodate a dynamic network

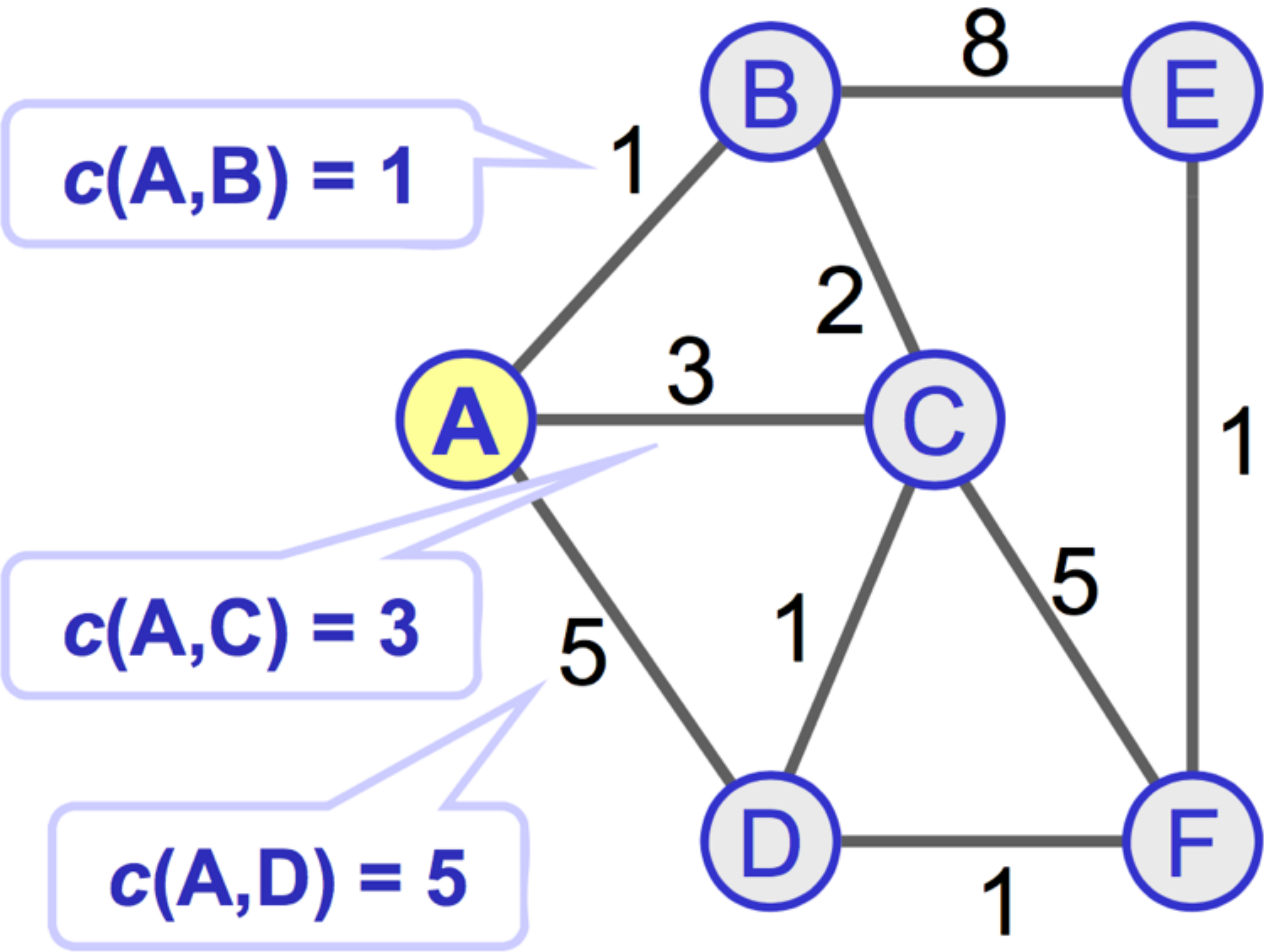
CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Discuss the basic objectives of routing and of least-cost routing

If you have any difficulties, please review the lecture video before continuing.

DV Example: Initialization



		cost via		
	$D^A()$	B	C	D
destination	B	1	∞	∞
	C	∞	3	∞
	D	∞	∞	5
	E	∞	∞	∞
	F	∞	∞	∞

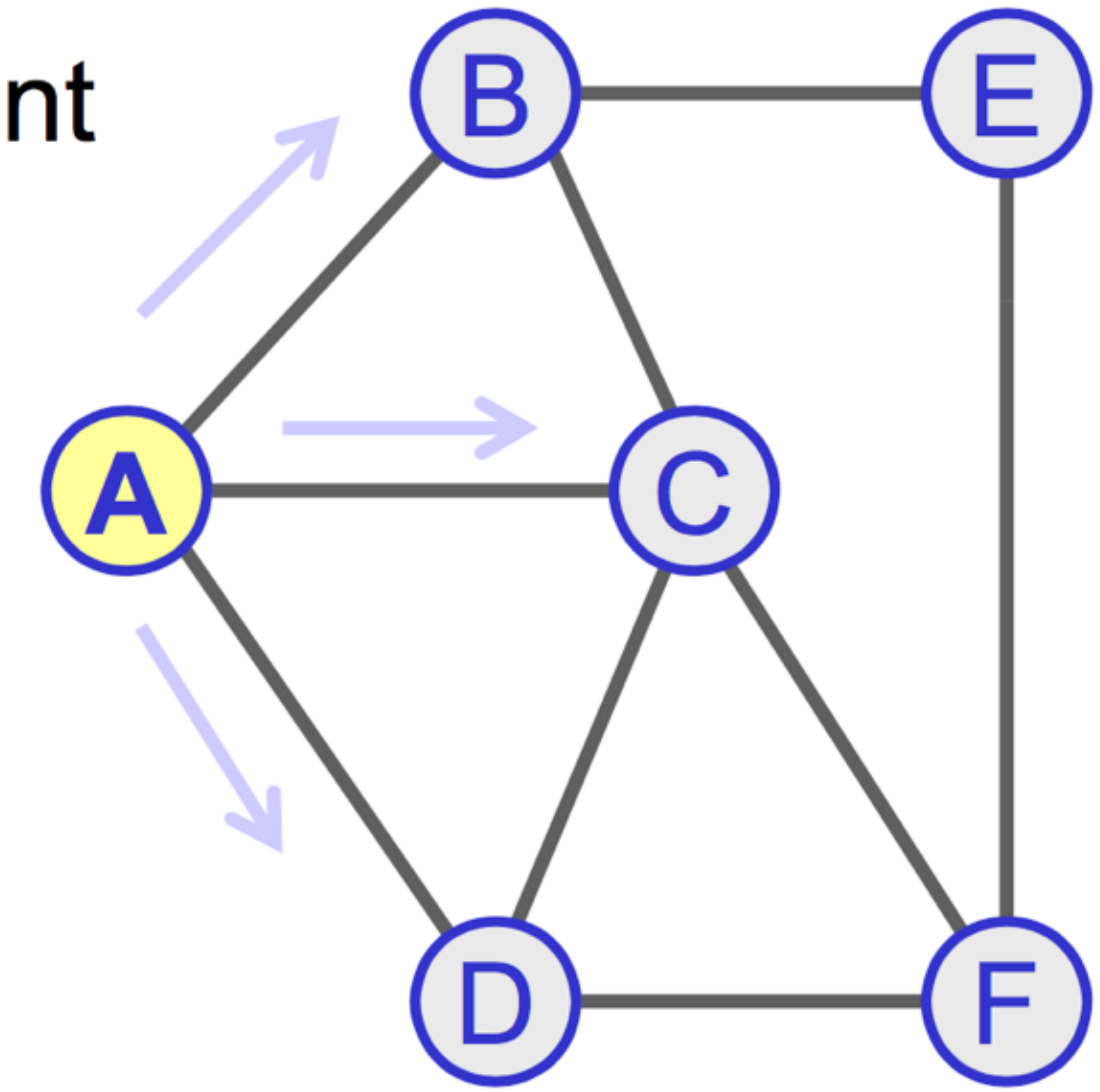
Distance Table at A

- Initially there are only paths from node A to each of its neighbors
- All other paths have infinite cost

DV Example: Updates from Node A

update sent

dest	cost
B	1
C	3
D	5



	$D^A()$	cost via B	cost via C	cost via D
destination	B	1	∞	∞
	C	∞	3	∞
	D	∞	∞	5
	E	∞	∞	∞
	F	∞	∞	∞

Distance Table at A

- Update message sent from node A to its three neighbors
- Similarly, node A will, at some time, receive update messages from its three neighbors

DV Example: Updates to A from B and D

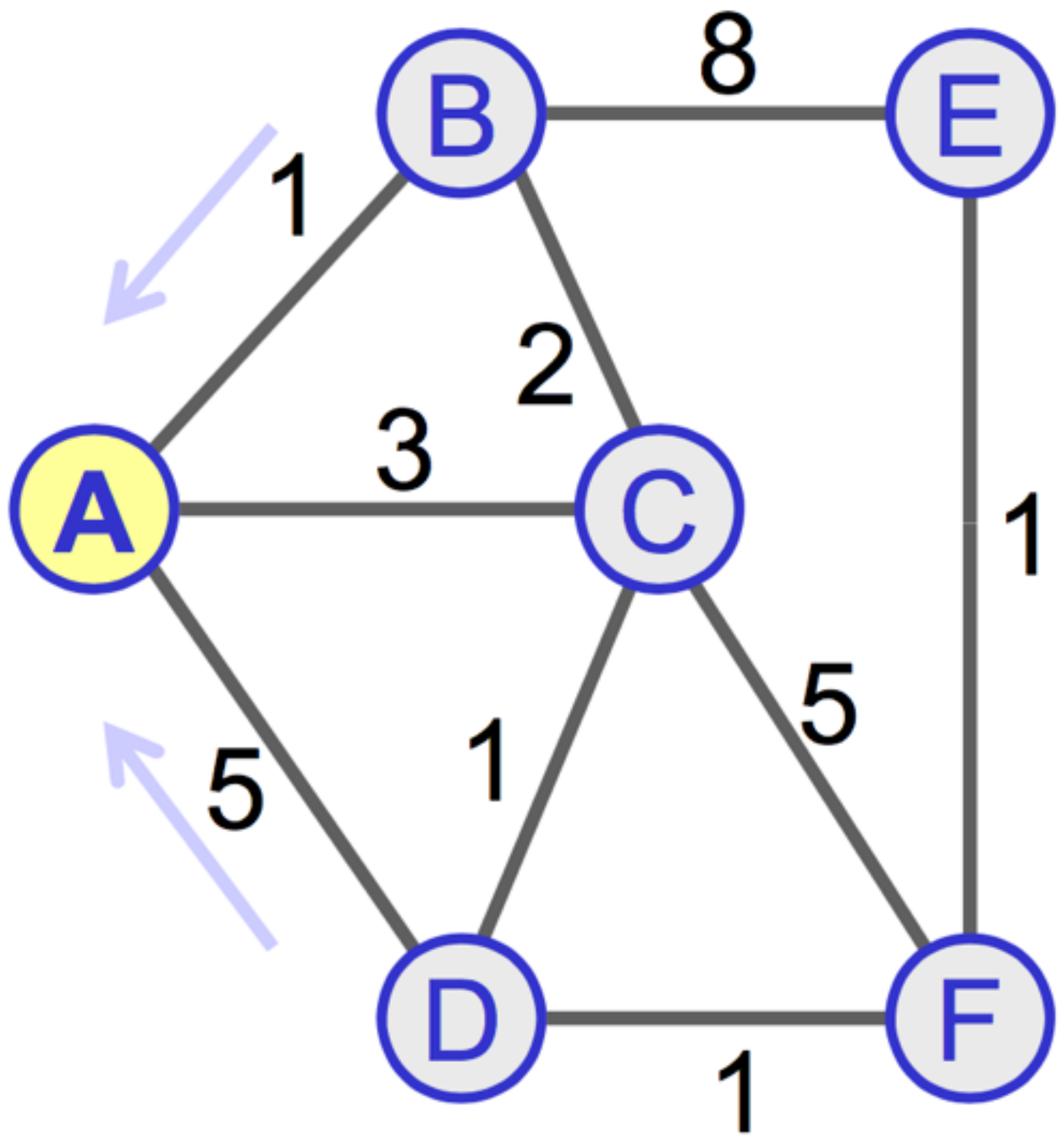
- Suppose nodes B, C, and D provide node A with update messages about their direct (one-hop) routes to other nodes

update from B

dest	cost
C	2
E	8

update from D

dest	cost
C	1
F	1



		cost via		
	$D^A()$	B	C	D
destination	B	1	5	∞
	C	3	3	6
	D	∞	4	5
	E	9	∞	∞
	F	∞	8	6

Distance Table at A

DV Example: So Far ...

- At this point we have a route to all nodes, but routes may not be the best due to:
  - Incomplete information (the case here)
  - Changes in link costs and topology

		cost via		
	$D^A()$	B	C	D
destination	B	1	5	∞
	C	3	3	6
	D	∞	4	5
	E	9	∞	∞
	F	∞	8	6

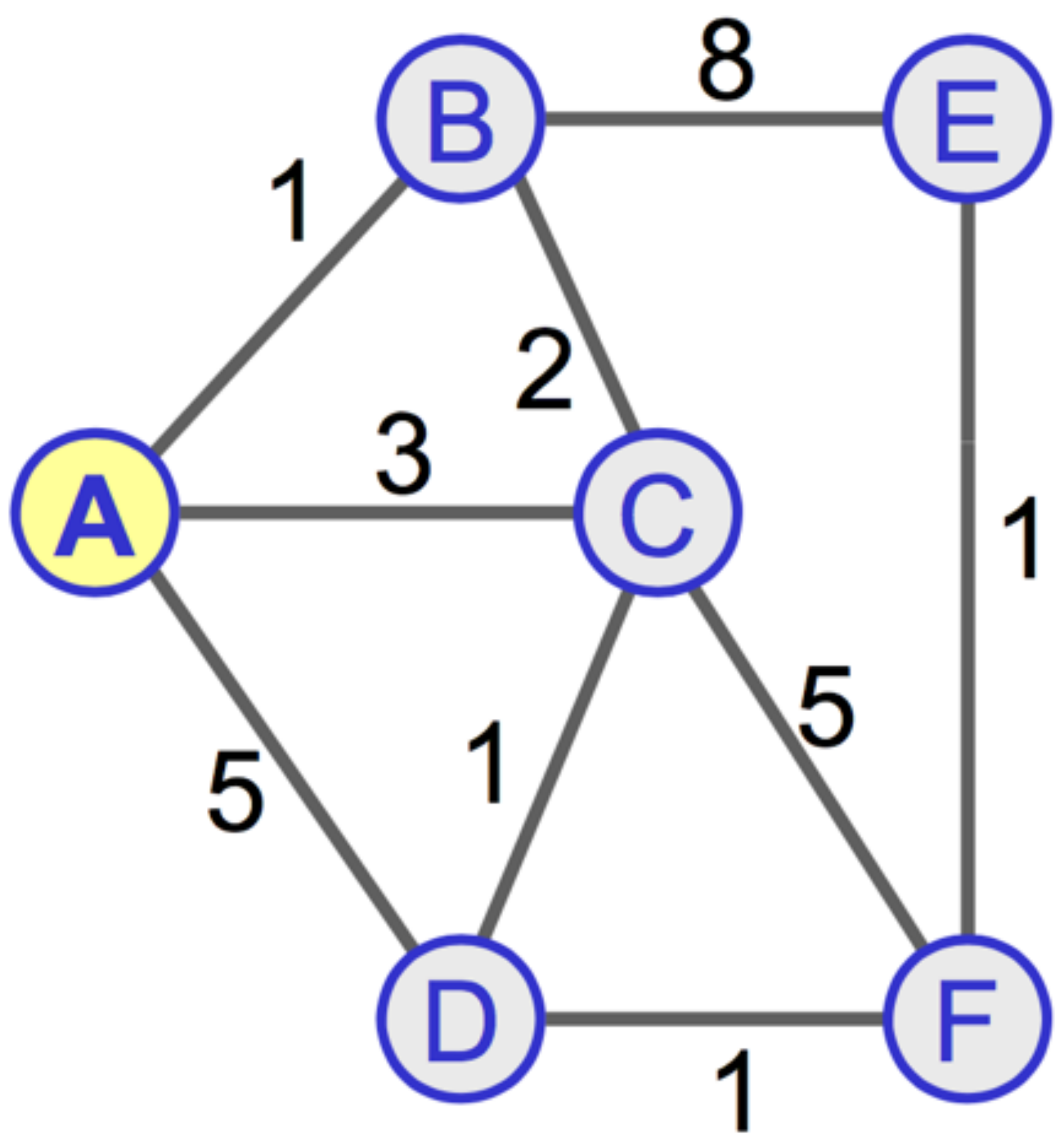
Distance Table at A

dest	next
B	B
C	C
D	C
E	B
F	D

Forwarding Table at A

DV Example: More Updates

- Suppose that nodes B, C, and D provide updates about “final” best routes, including
  - C has a better route to E with cost 3
  - C has a better route to F with cost 2



	$D^A()$	B	C	D
destination	B	1	5	8
	C	3	3	6
	D	4	4	5
	E	9	6	7
	F	5	5	6

(cost via: B, C, D)

Distance Table at A

DV Example: Best Routes

- The routing table is now as shown below
  - Remains the same as long as the topology and costs do not change
- Note that there are other possible least-cost routes

		cost via		
	$D^A()$	B	C	D
destination	B	1	5	8
	C	3	3	6
	D	4	4	5
	E	9	6	7
	F	5	5	6

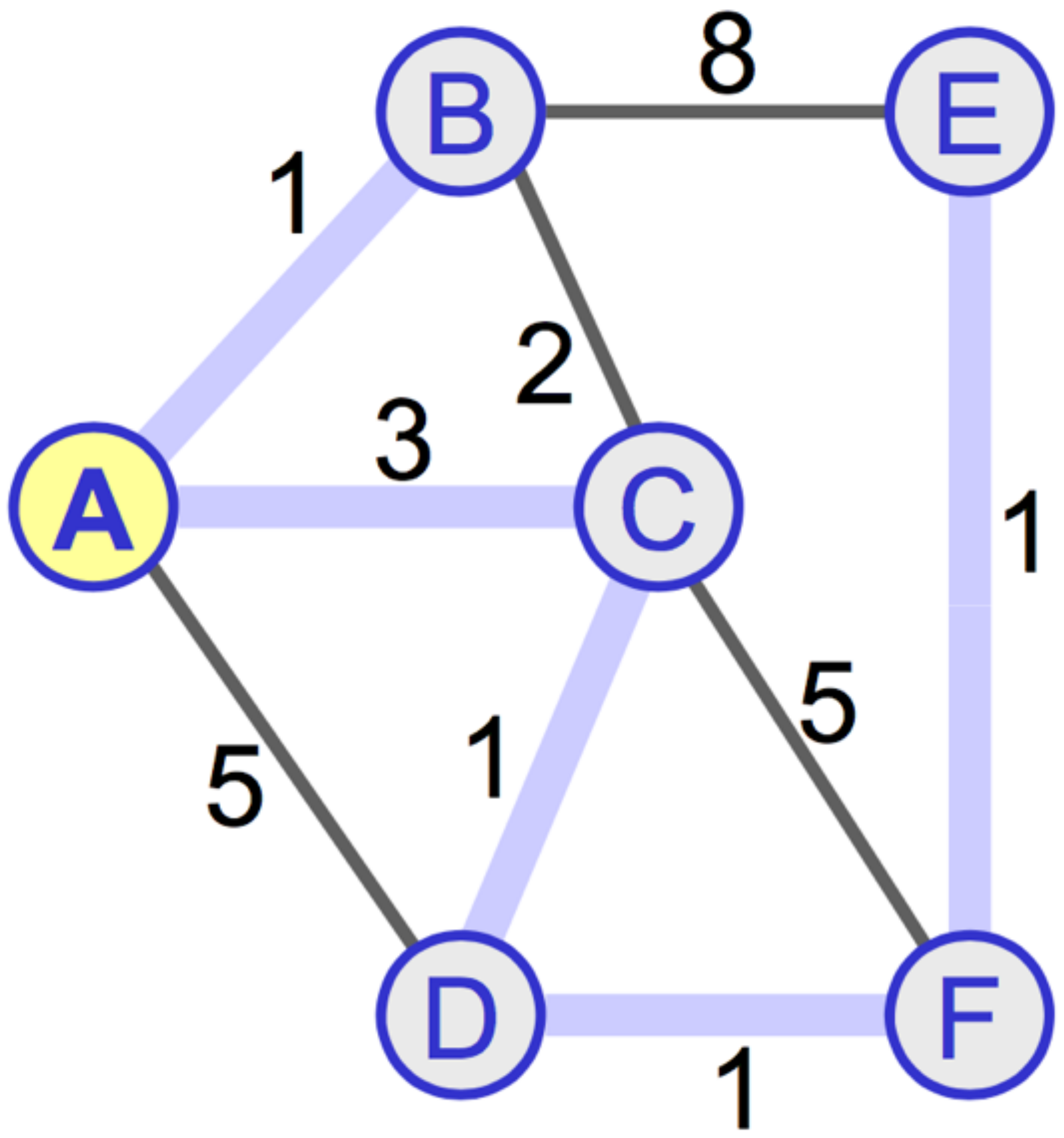
Distance Table at A

dest	next
B	B
C	C
D	C
E	C
F	C

Forwarding Table at A

DV Example: Resulting Paths from A

- The resulting least-cost paths created by Bellman-Ford in this example are shown below
- Note that there are other least-cost spanning trees



		cost via		
	$D^A()$	B	C	D
destination	B	1	5	8
	C	3	3	6
	D	4	4	5
	E	9	6	7
	F	5	5	6

Distance Table at A

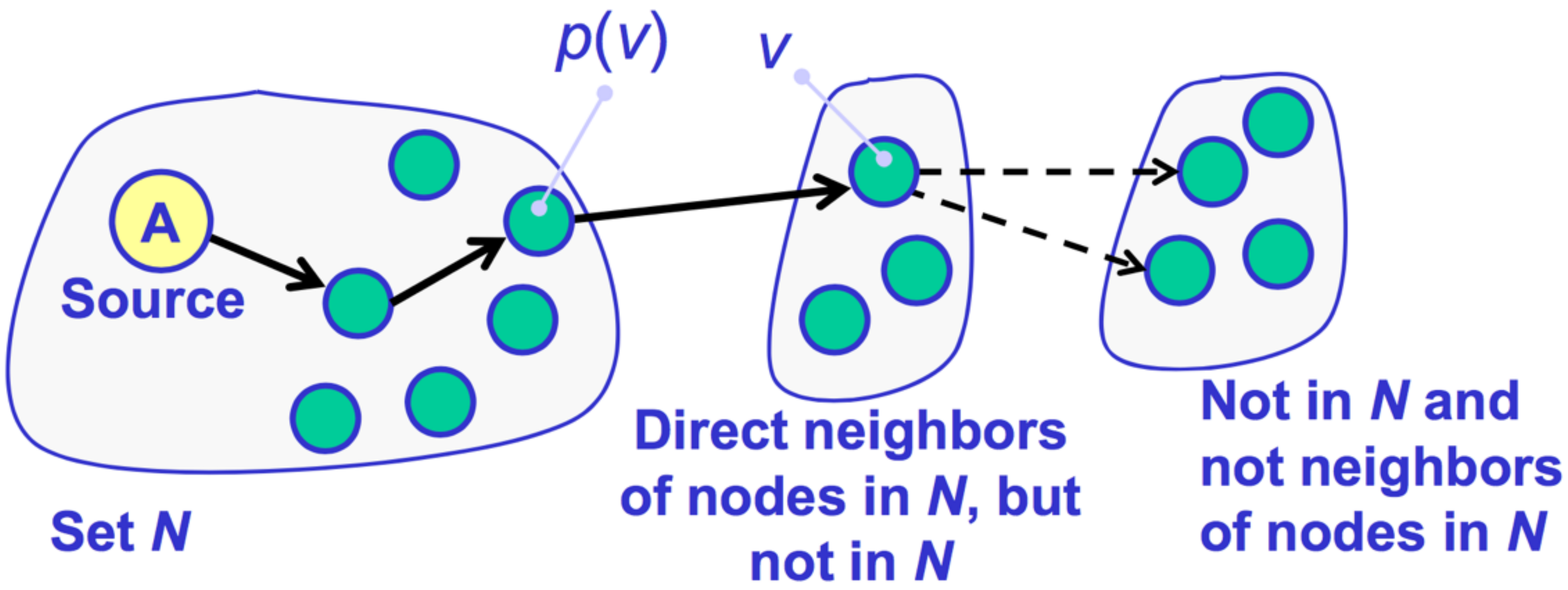
CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe the key features of distance vector algorithms
- Describe the basic operation of the asynchronous distributed Bellman-Ford algorithm as used for distance vector (DV) routing

If you have any difficulties, please review the lecture video before continuing.

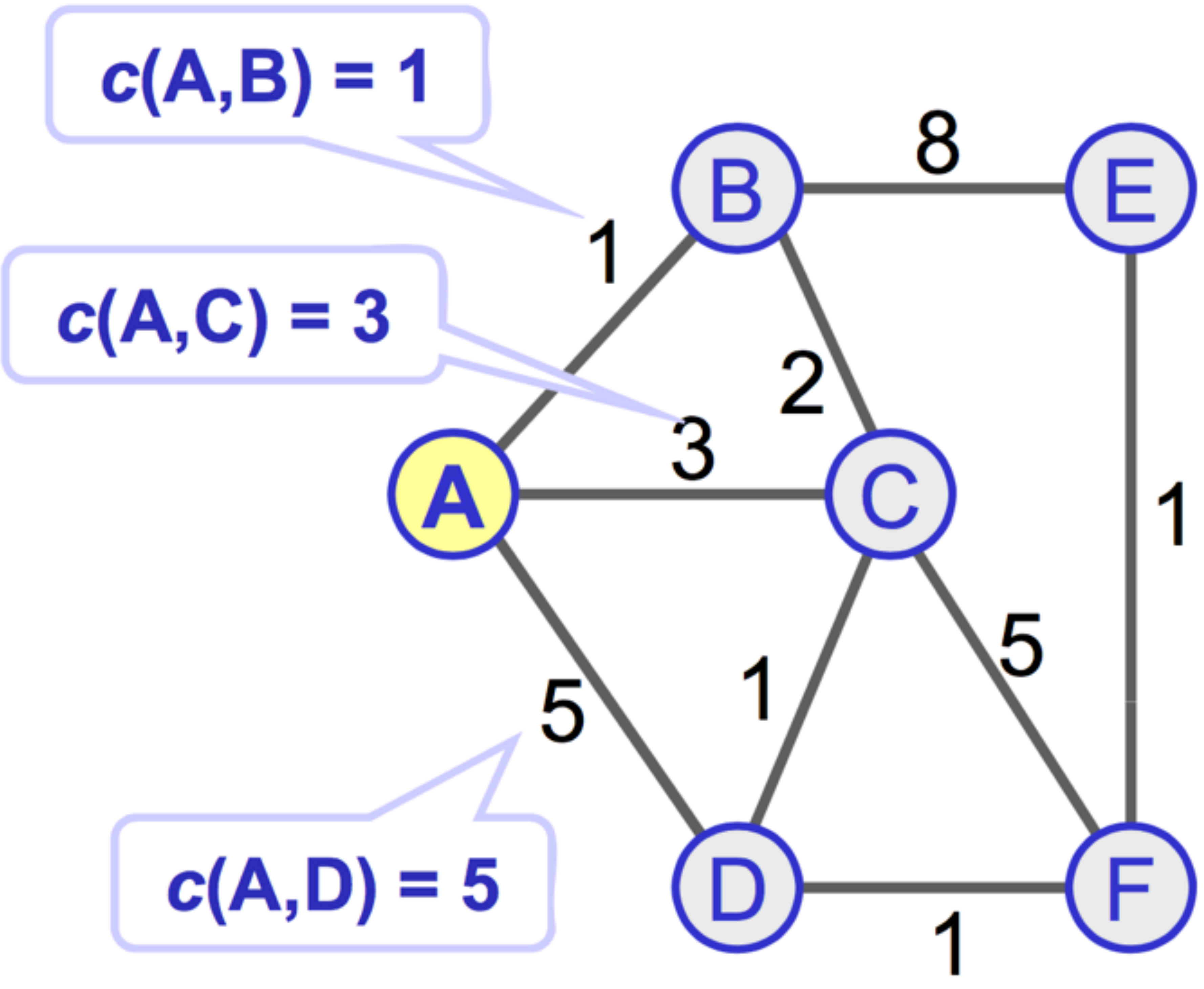
General Operation of Dijkstra's Algorithm



- Will find one new lowest cost path at each iteration
- With current set N, find lowest cost path to a neighbor of a node in N that is not yet in N
- That node now joins N and may add new neighbors

LS Example: Iteration 0 (Initialization)

- At iteration 0, set N and initial D(v) values are set



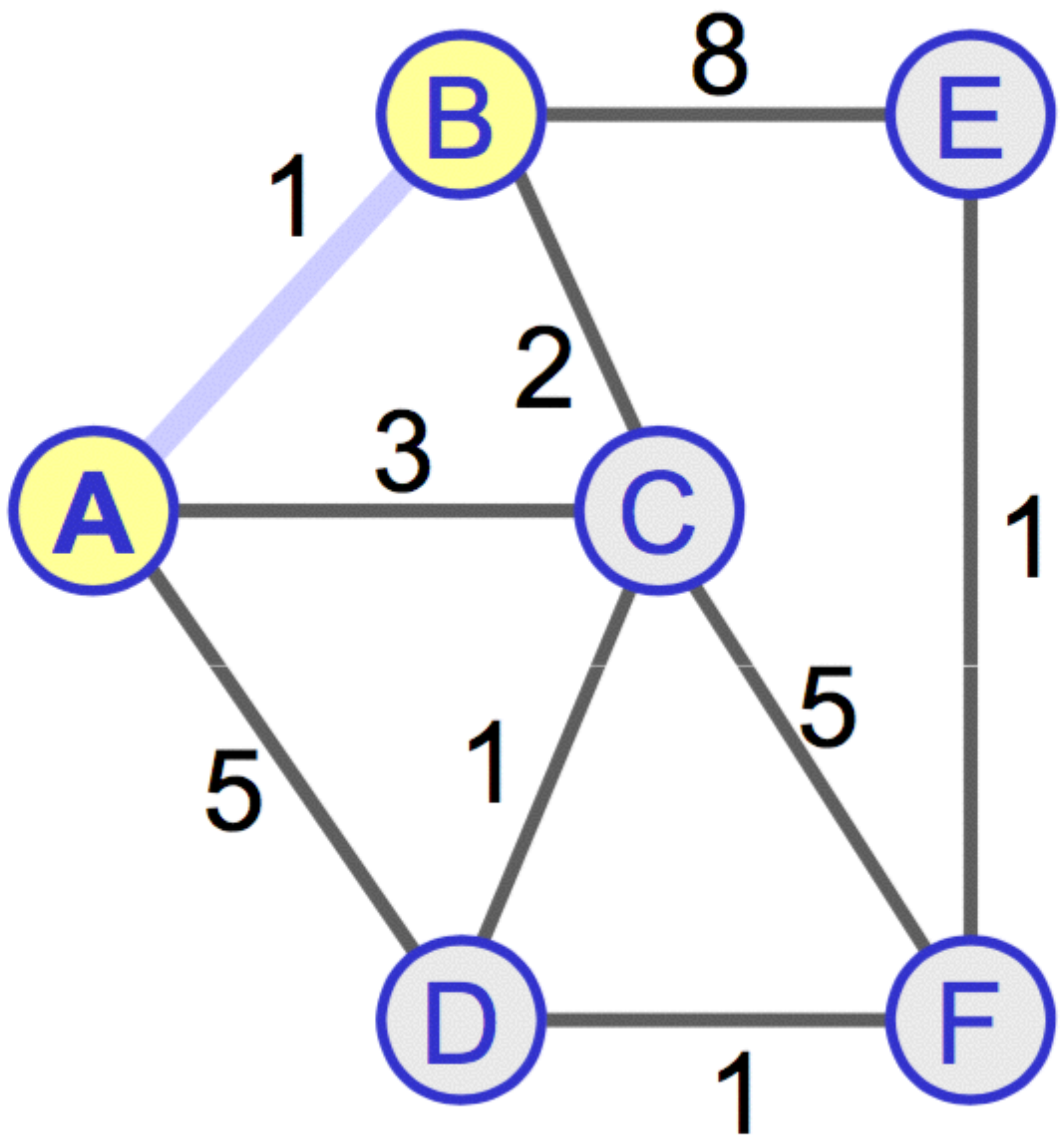
N = { A }

iteration: 0

v	*D*(*v*)
B	1
C	3
D	5
E	∞
F	∞

LS Example: Iteration 1

- In the first pass through the loop, the path to B is the lowest cost one not in N
- B is added to the set N and distances are updated



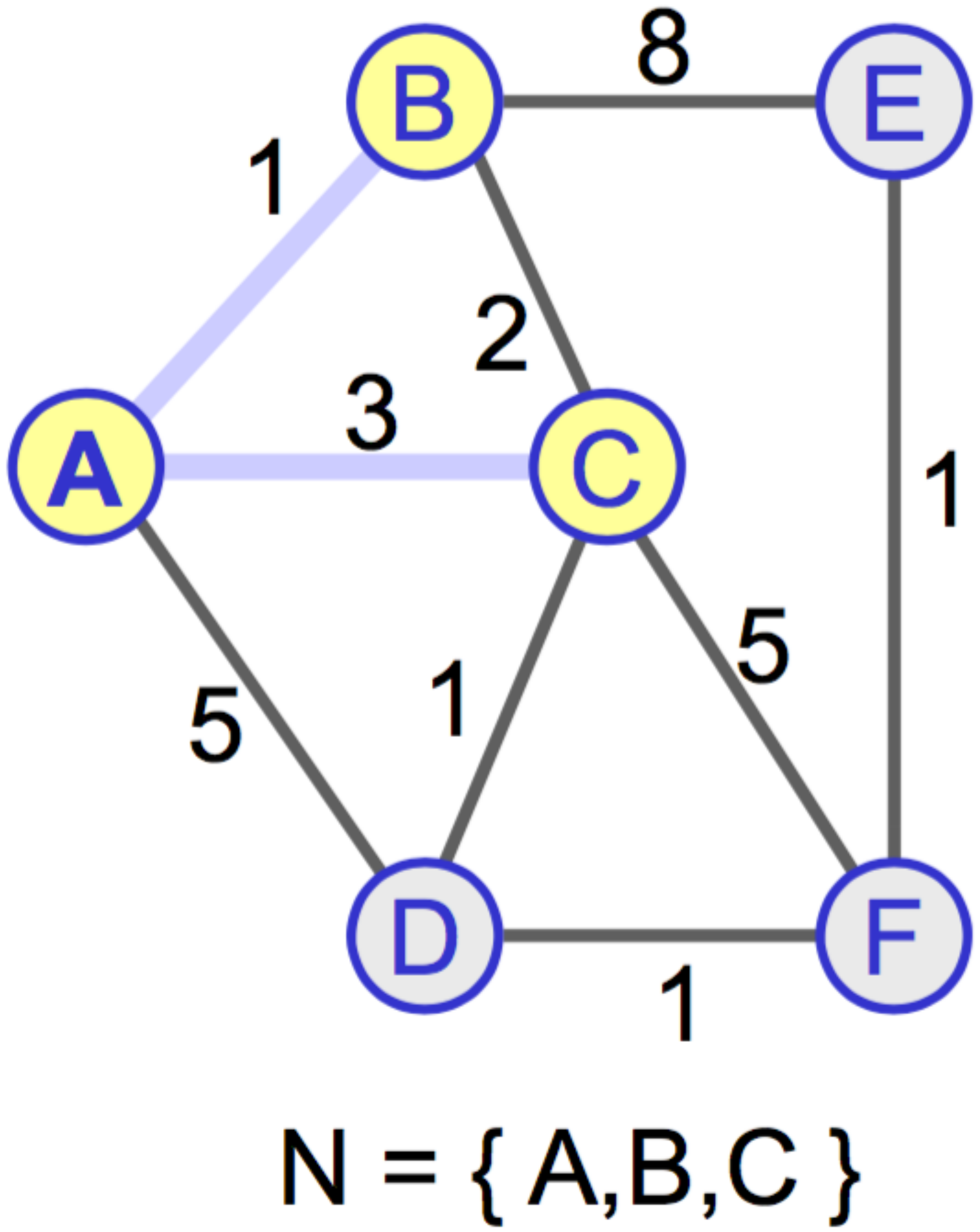
N = { A,B }

iteration: 0 1

v	$D(v)$	$D(v)$
B	1	1
C	3	3
D	5	5
E	∞	9
F	∞	∞

LS Example: Iteration 2

- At iteration 2, node C is added to N

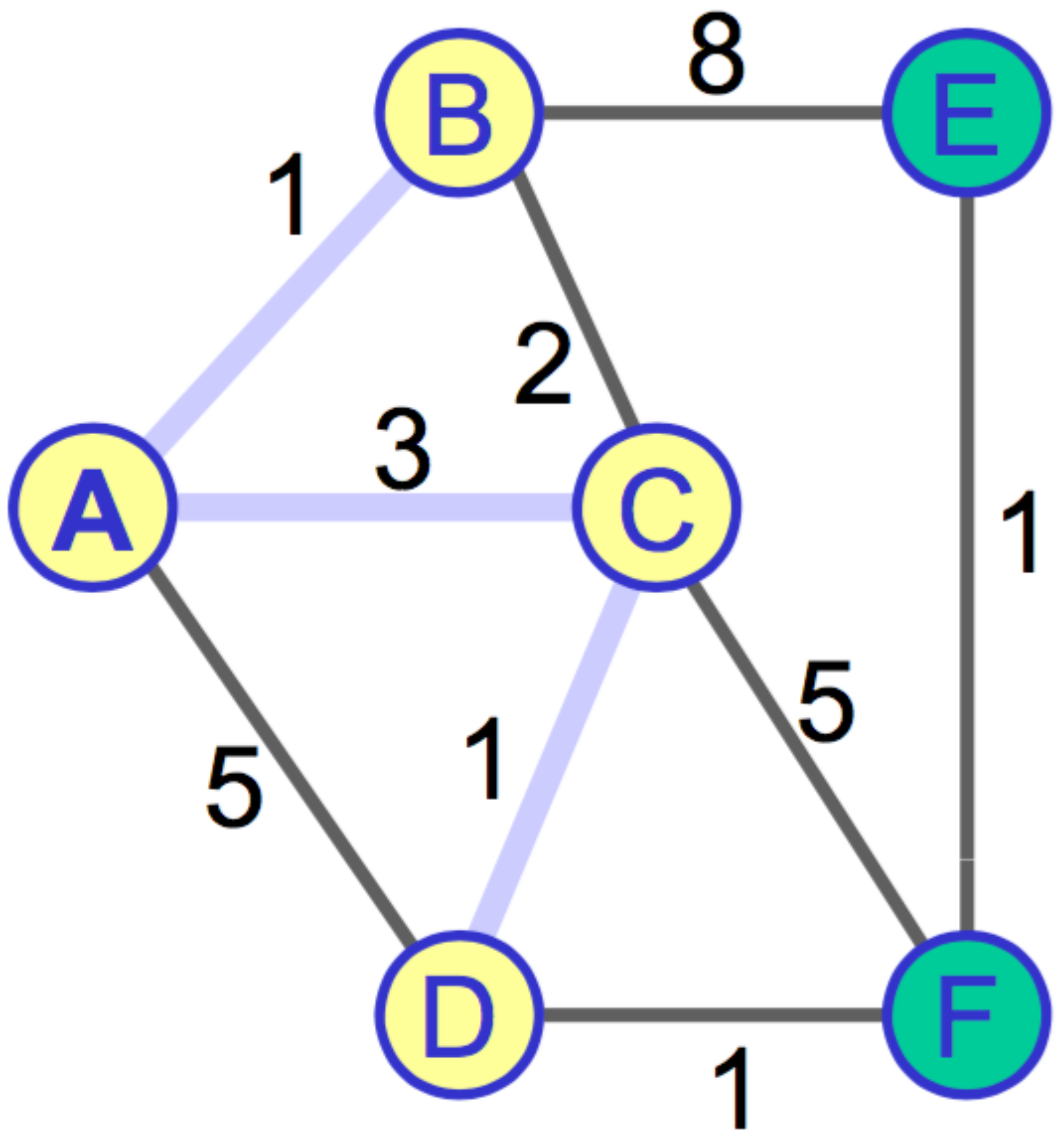


iteration: 0 1 2

v	*D*(*v*)	*D*(*v*)	*D*(*v*)
B	1	1	1
C	3	3	3
D	5	5	4
E	∞	9	9
F	∞	∞	8

LS Example: Iteration 3

- Node D is added to set N at iteration 3



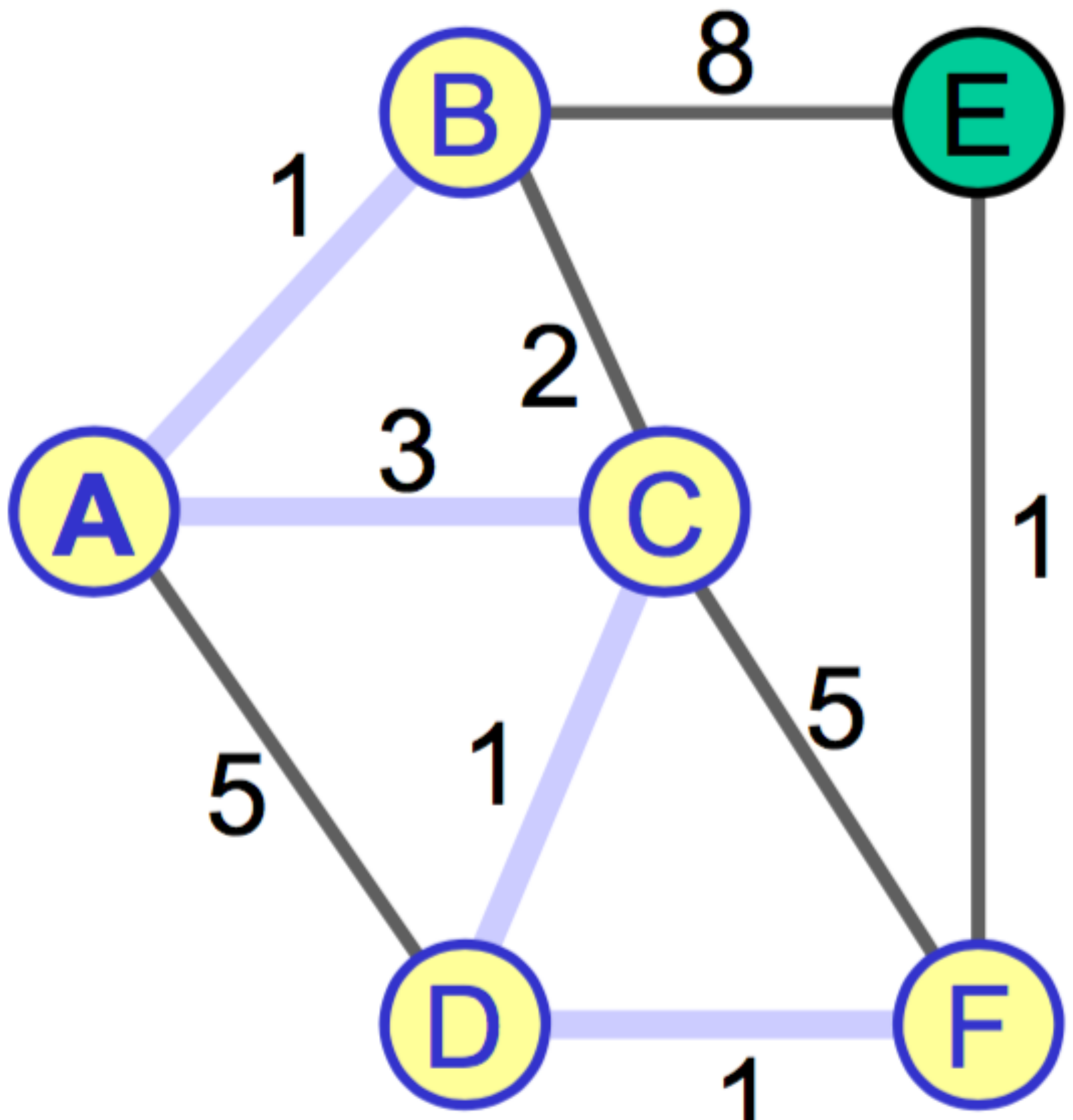
N = { A,B,C,D }

iteration:	0	1	2	3
v	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)
B	1	1	1	1
C	3	3	3	3
D	5	5	4	4
E	∞	9	9	9
F	∞	∞	8	5

VirginiaTech
Invent the Future®

LS Example: Iteration 4

- Node F is added at iteration 4

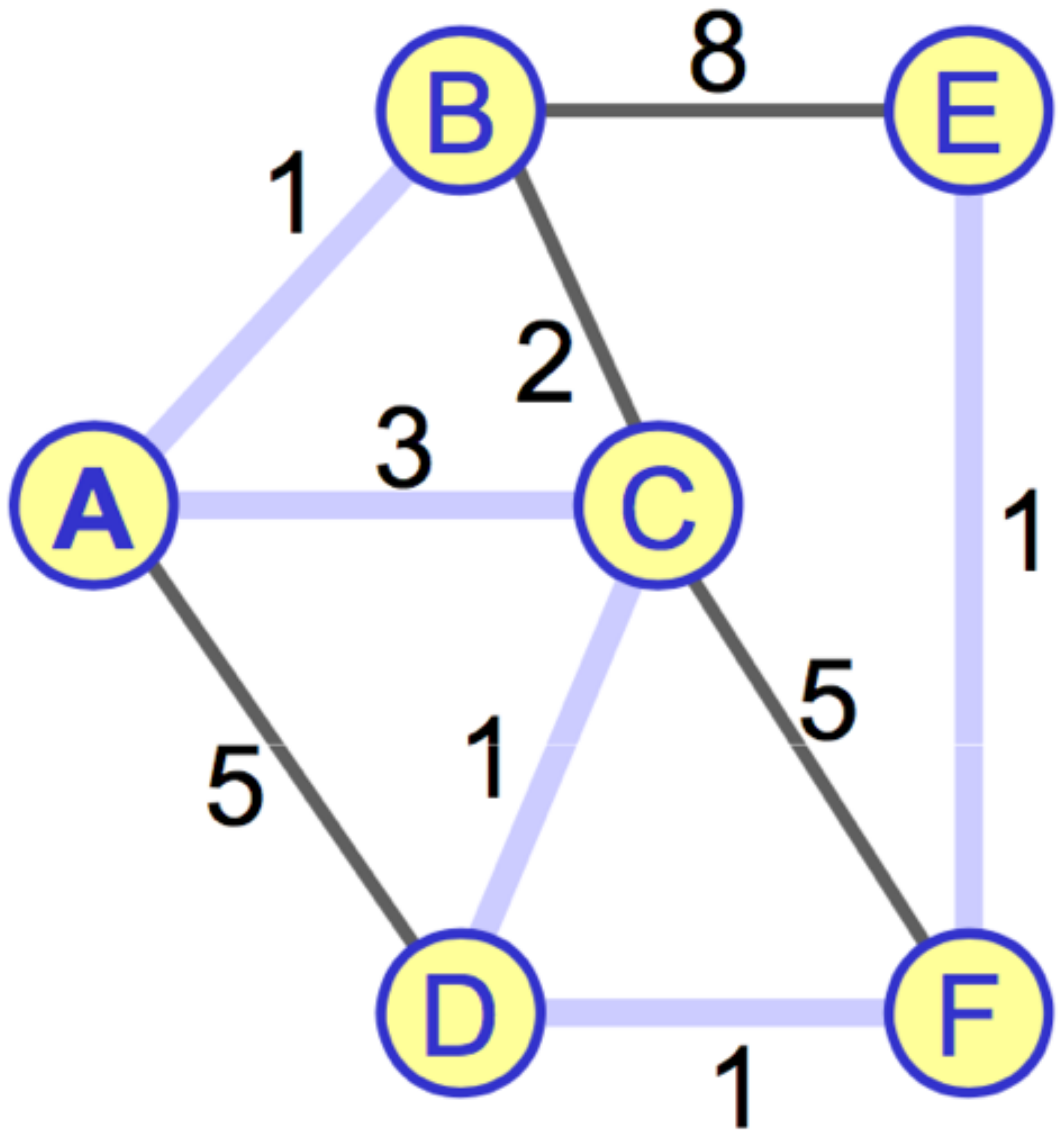


N = { A,B,C,D,F }

iteration:	0	1	2	3	4
v	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)
B	1	1	1	1	1
C	3	3	3	3	3
D	5	5	4	4	4
E	∞	9	9	9	6
F	∞	∞	8	5	5

LS Example: Iteration 5

- Node E is added at the 5th and final iteration
- This defines the set of least-cost paths from Node A



N = { A,B,C,D,E,F }

iteration:	0	1	2	3	4	5
v	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)	*D*(*v*)
B	1	1	1	1	1	1
C	3	3	3	3	3	3
D	5	5	4	4	4	4
E	∞	9	9	9	6	6
F	∞	∞	8	5	5	5

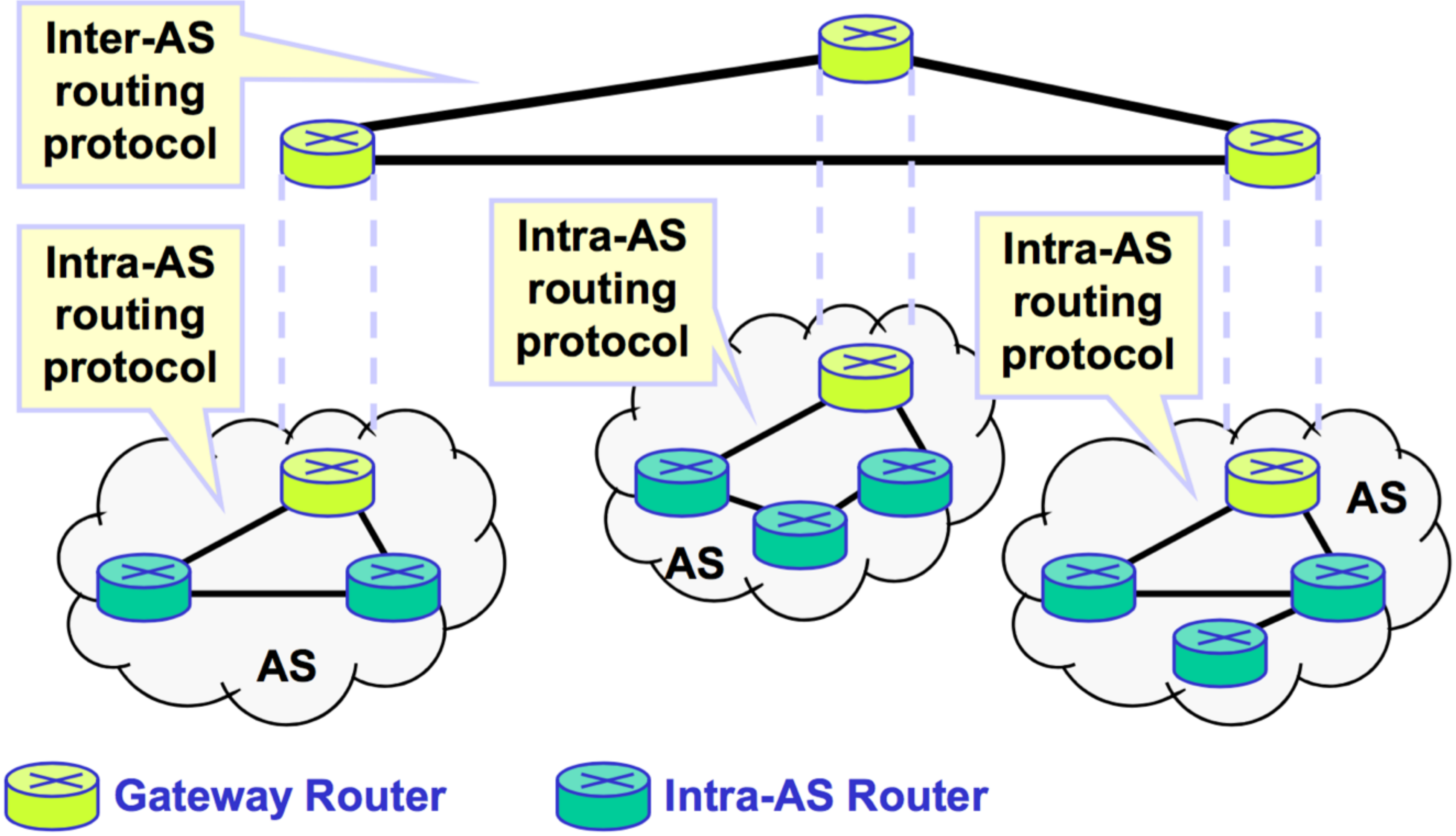
Link State Versus Distance Vector

- Distance vector routing uses local information
    - Uses distance to destinations as seen by neighbors
    - There are inefficiencies due to the dynamic nature of the network
    - There are methods to overcome problems with distance vector routing
- Link state routing uses global information
    - Topology information flooded to the network
    - Each node then runs its own instance of a shortest or least-cost path algorithm
    - Performance is good, but at the cost of complexity

Need for Hierarchical Routing

- Problems with simple "flat" routing
  - Scalability – how do routing algorithms deal with numbers of routers in the millions to hundreds of millions?
  - Administration – how can separate organizations (e.g., different Internet Service Providers and corporations) administer their own internal network, yet allow it to operate as part of the Internet?
- Solution is to decompose the global network into a set of autonomous systems (ASs)

Autonomous Systems



CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe the key features of link state routing algorithms
- Describe the operation of Dijkstra's algorithm as used for link state (LS) routing
- Discuss the motivation for and concept of hierarchical routing

If you have any difficulties, please review the lecture video before continuing.

Summary

- Network layer routing enables the end-to-end movement of packets through a network
- The routing may use circuit switching, datagram routing, or virtual circuits
- Datagram routing is a connectionless method where each packet (a datagram) carries full address information for the receiver and a routing table is consulted prior to forwarding the datagram

Summary (cont'd)

- Key characteristics of routing schemes
  - Network topology: static or dynamic
  - Execution location: centralized or distributed
  - Information requirements: local or global
- Bellman-Ford algorithm (for distance vector routing) and Dijkstra's algorithm (for link state routing) are commonly used in routing
- Hierarchical routing provides scalability and autonomy

MODULE 11: The Internet Protocol Suite

Lecture 11.5
Network Layer and Routing

Prepared By:

- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering

Virginia Tech