

Network Layer

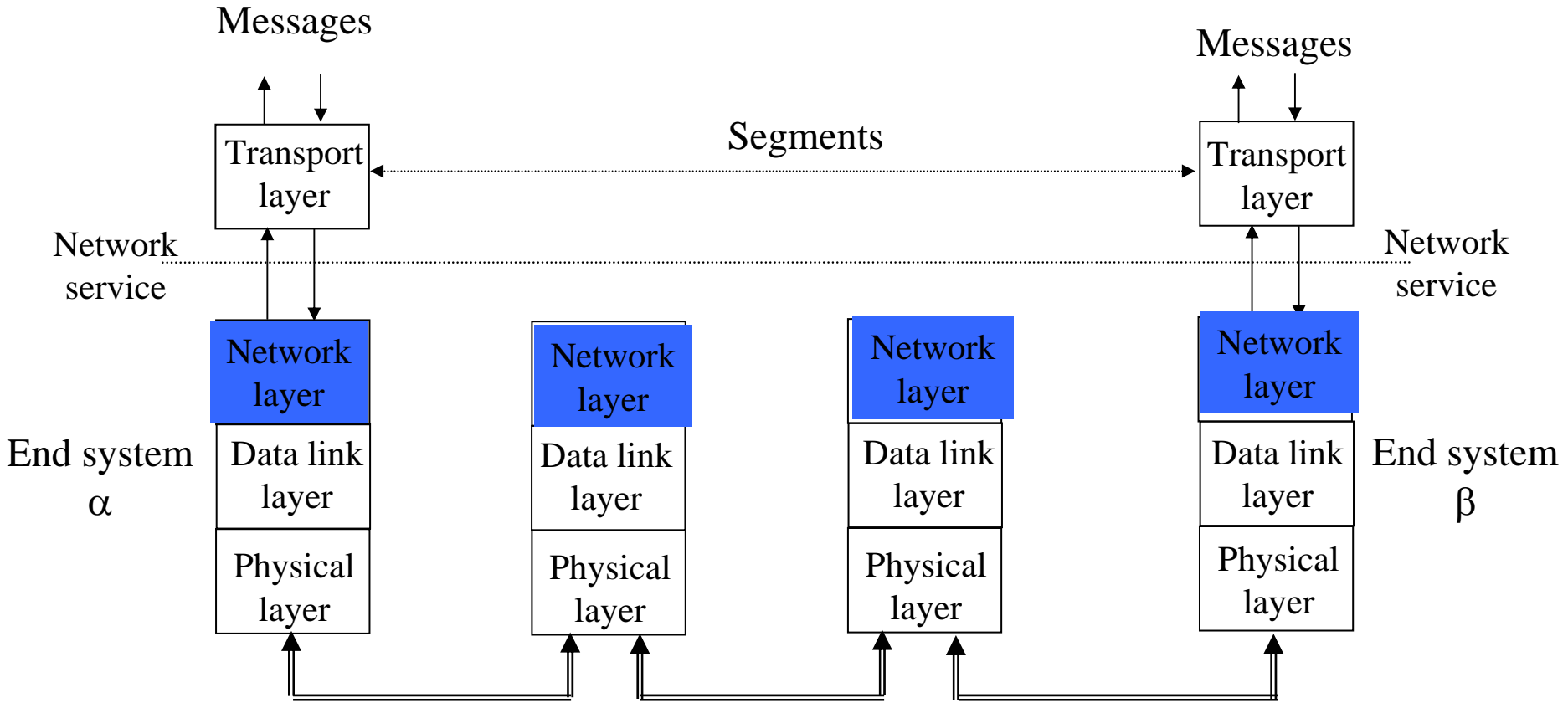
Routing

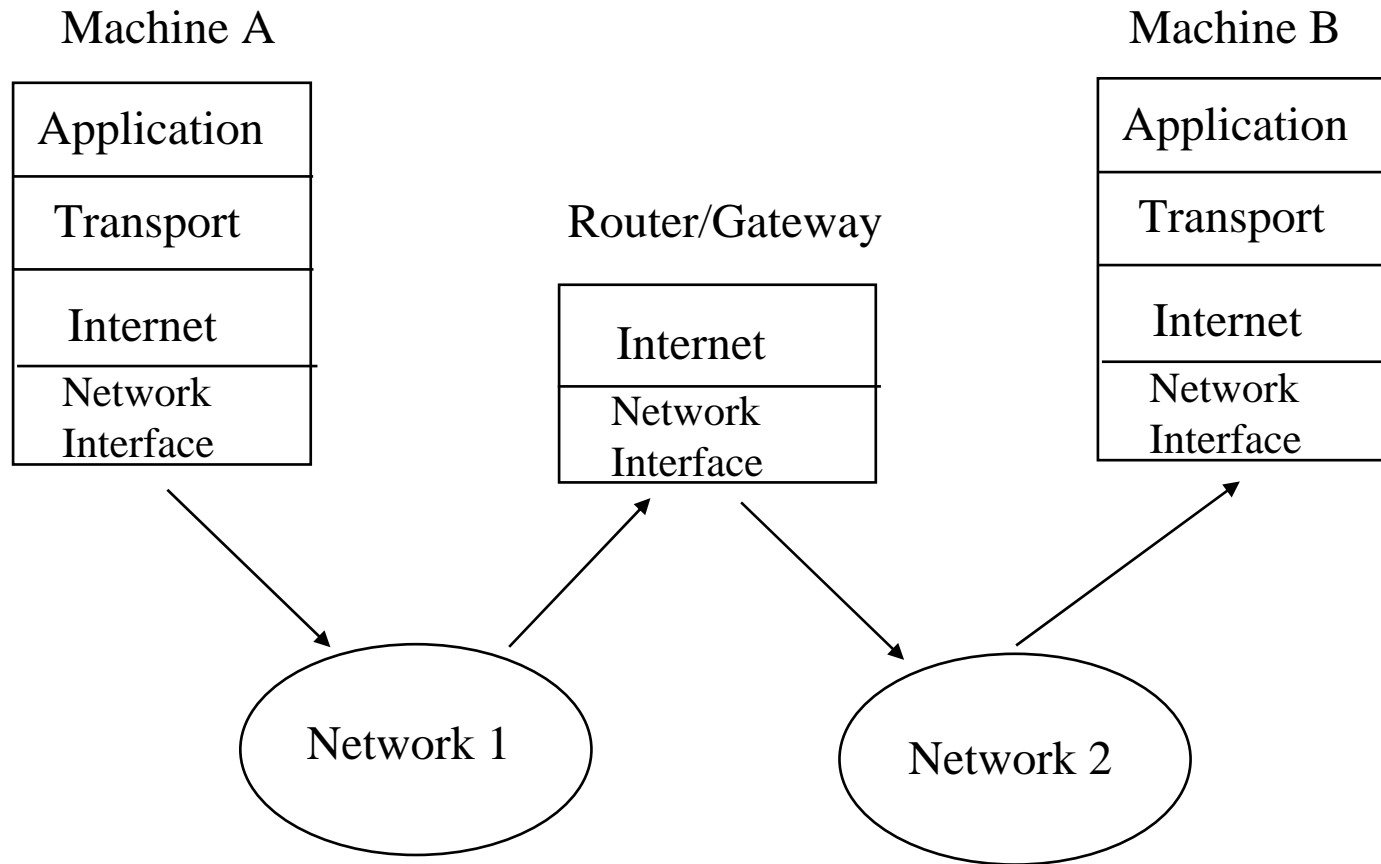
Network Layer

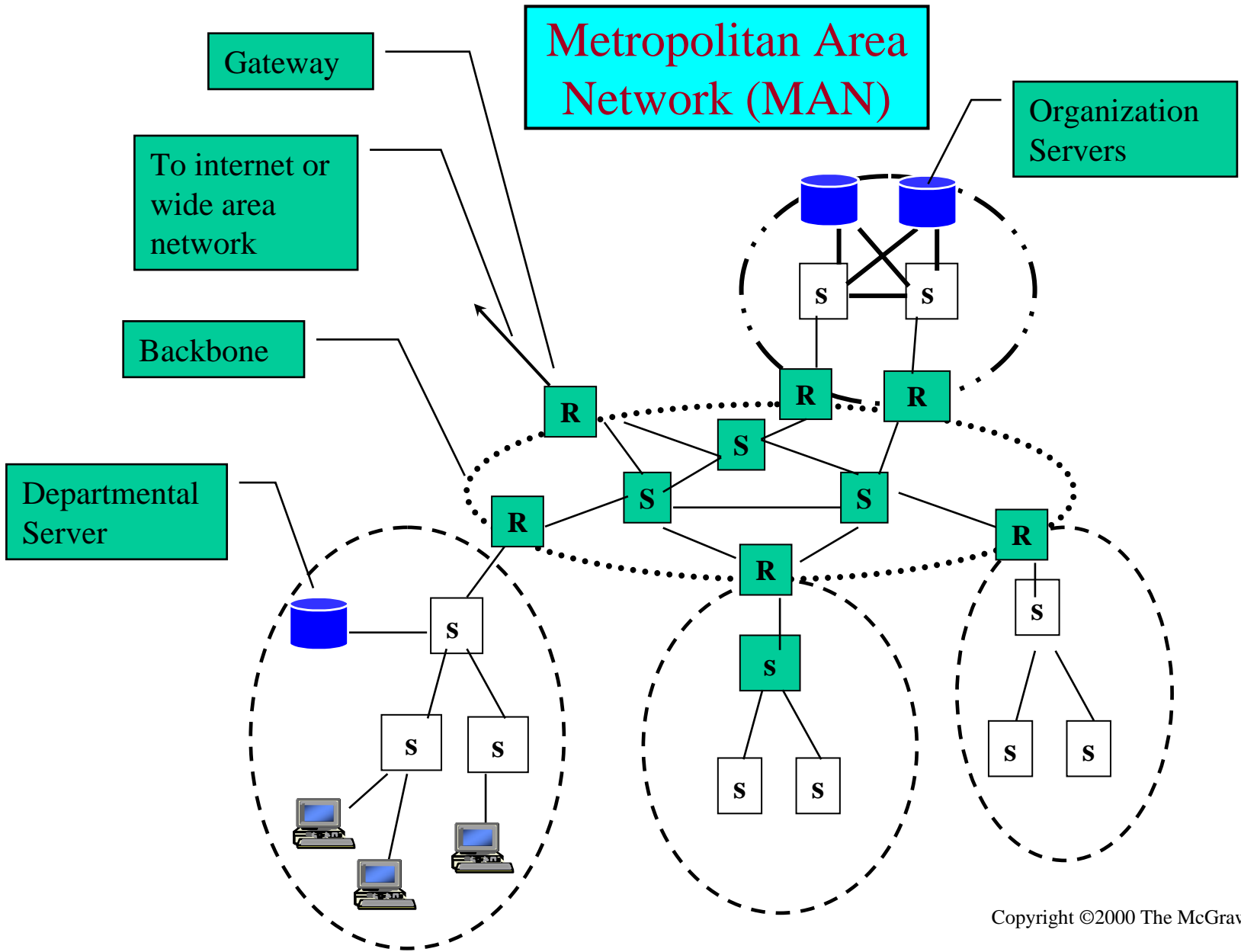
- Concerned with getting packets from source to destination.
- The network layer must know the topology of the subnet and choose appropriate paths through it.
- When source and destination are in *different networks*, the network layer (**IP**) must deal with these differences.
- * **Key issue:** *what service does the network layer provide to the transport layer (connection-oriented or connectionless).*

Network Layer Design Goals

1. The services provided by the network layer should be **independent** of the subnet topology.
2. The Transport Layer should be shielded from the number, type and topology of the subnets present.
3. The network addresses available to the Transport Layer should use a uniform numbering plan (even across LANs and WANs).







Copyright ©2000 The McGraw Hill Companies

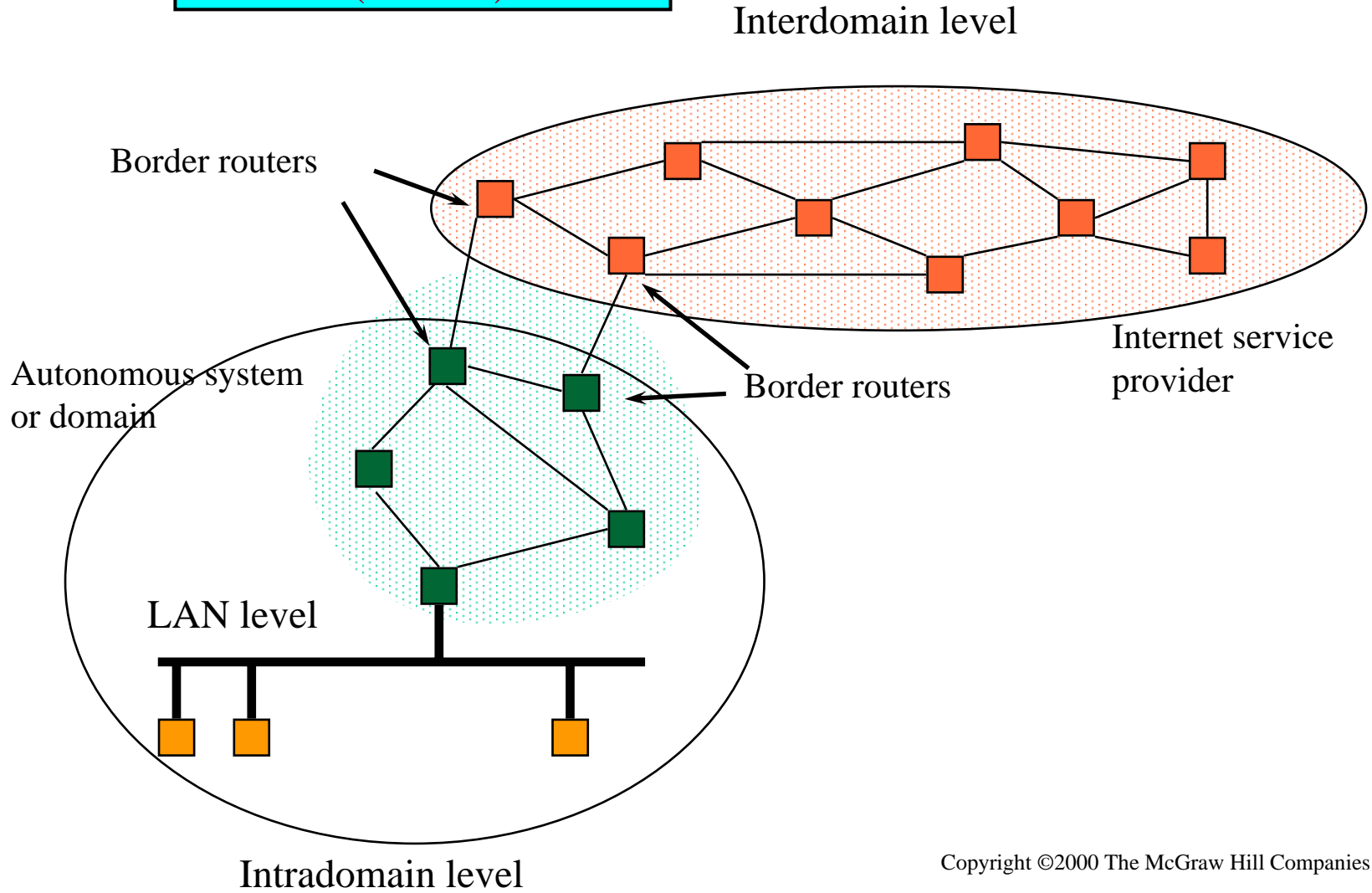
Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.6

Networks: Routing



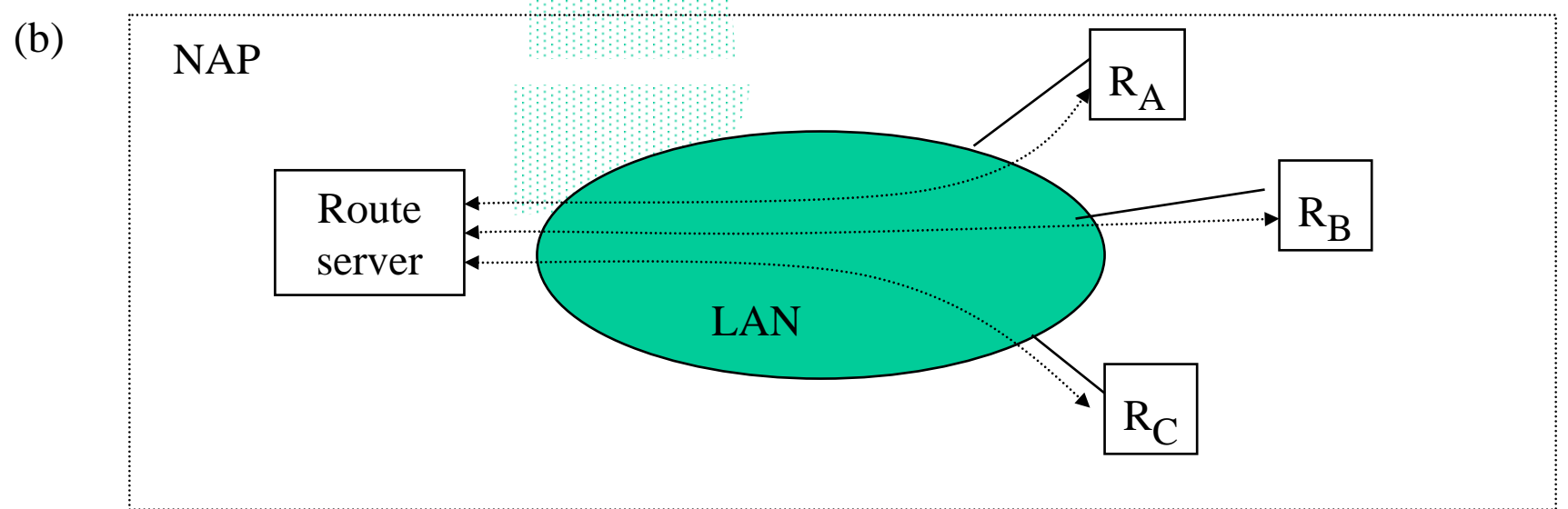
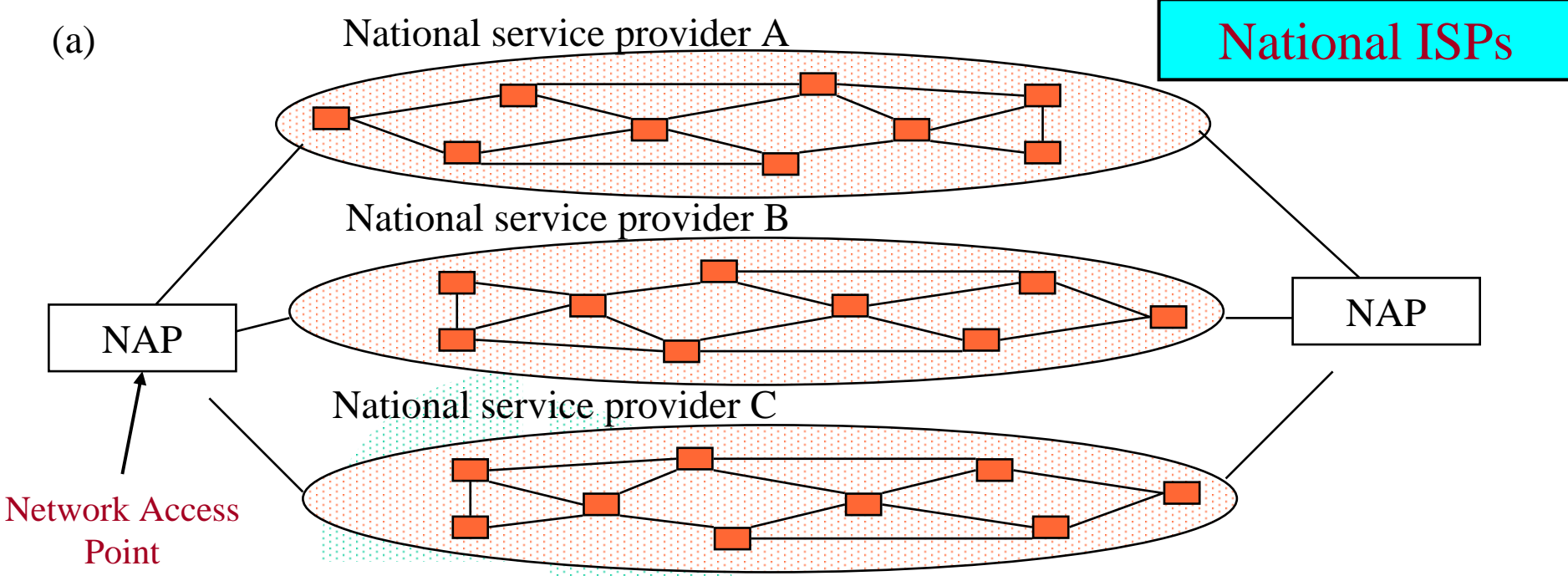
Wide Area Network (WAN)



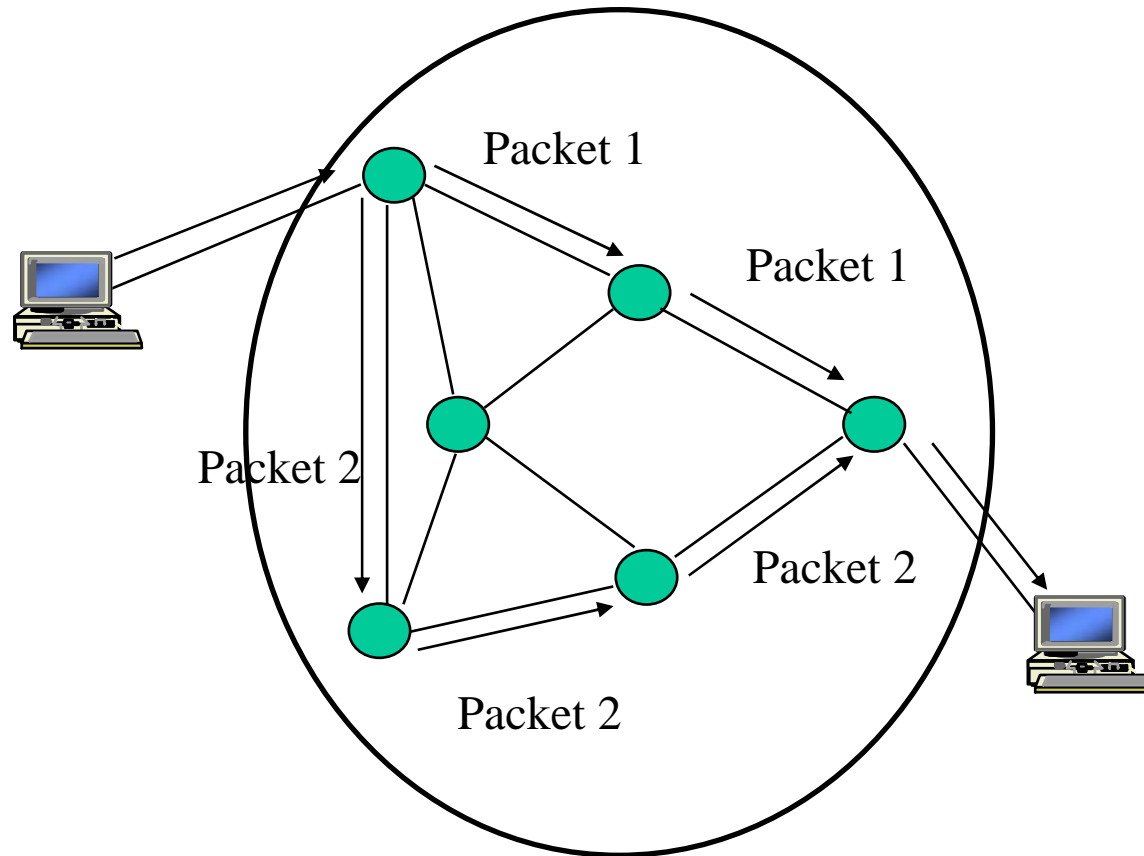
Copyright ©2000 The McGraw Hill Companies

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.7



Datagram Packet Switching



Routing Table in Datagram Network

Destination address	Output port
0785	7
1345	12
1566	6
2458	12

Copyright ©2000 The McGraw Hill Companies

Leon-Garcia & Widjaja: *Communication Networks*

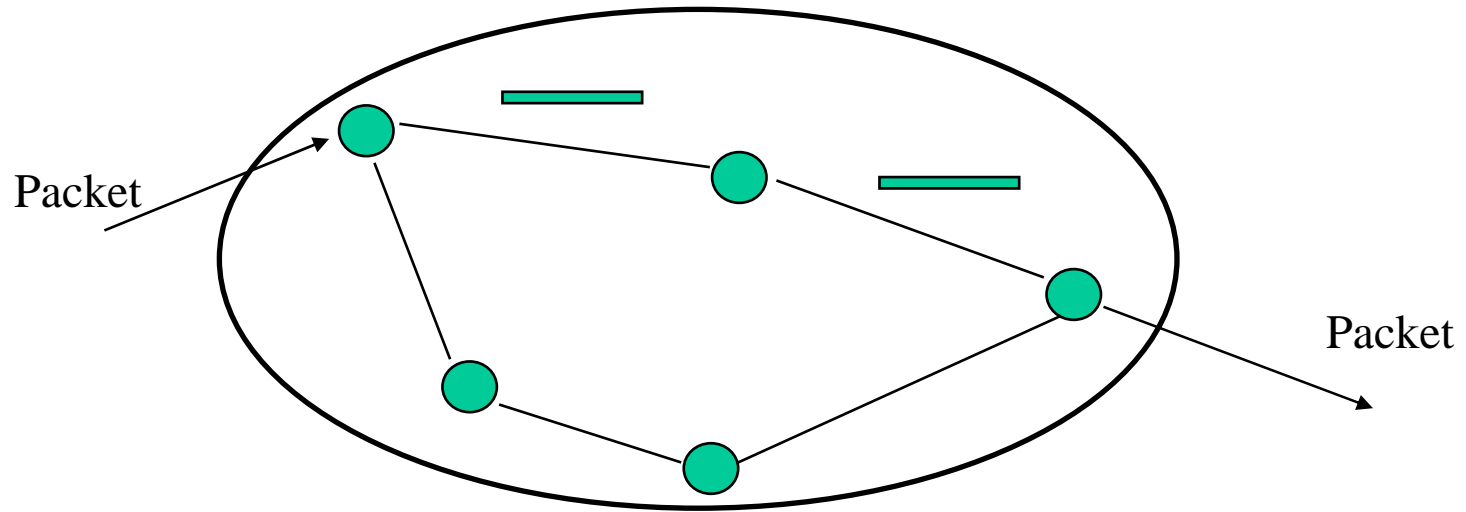
Figure 7.16

Networks: Routing

10



Virtual Circuit Packet Switching

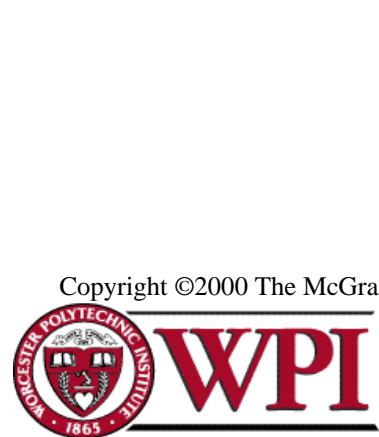


Routing Table in Virtual Circuit Network

Identifier Output
 port
 Next
 identifier

Identifier	Output port	Next identifier
12	13	44
15	15	23
27	13	16
58	7	34

Entry for packets
with identifier 15



Routing

Routing algorithm:: that part of the Network Layer responsible for deciding on which output line to transmit an incoming packet.

- Remember: For virtual circuit subnets the routing decision is made ONLY at set up.

Algorithm properties:: correctness, simplicity, robustness, stability, fairness, optimality, and scalability.

Routing Classification

Adaptive Routing

- based on current measurements of traffic and/or topology.
1. centralized
 2. isolated
 3. distributed

Non-Adaptive Routing

- routing computed in advance and off-line
1. flooding
 2. static routing using shortest path algorithms

Flooding

- *Pure flooding* :: every incoming packet to a node is sent out on **every** outgoing line.
 - Obvious adjustment – do not send out on arriving link (assuming full-duplex links).
 - The routing algorithm can use a hop counter (e.g., TTL) to **dampen the flooding**.
 - *Selective flooding* :: only send on those lines going “approximately” in the right direction.

Shortest Path Routing

1. Bellman-Ford Algorithm [Distance Vector]
2. Dijkstra's Algorithm [Link State]

What does it mean to be the shortest (or optimal) route?

- a. **Minimize mean packet delay**
- b. **Maximize the network throughput**
- c. **Minimize the number of hops along the path**

Dijkstra's Shortest Path Algorithm

Initially mark all nodes (except source) with infinite distance.

working node = source node

Sink node = destination node

While the working node is not equal to the sink

1. Mark the working node as permanent.
2. Examine all adjacent nodes in turn

If the sum of label on working node plus distance from working node to adjacent node is less than current labeled distance on the adjacent node, this implies a shorter path. Relabel the distance on the adjacent node and label it with the node from which the probe was made.

3. Examine all tentative nodes (not just adjacent nodes) and mark the node with the smallest labeled value as permanent. This node becomes the new working node.

Reconstruct the path backwards from sink to source.

Internetwork Routing [Halsall]

Adaptive Routing

Centralized
[RCC]

Distributed

Isolated

[IGP] Intradomain routing

Interdomain routing [EGP]

Interior
Gateway Protocols

[BGP, IDRP]

Exterior
Gateway Protocols

Distance Vector routing
[RIP]

Link State routing
[OSPF, IS-IS, PNNI]

Adaptive Routing

Basic functions:

1. Measurement of pertinent network data.
2. Forwarding of information to where the routing computation will be done.
3. Compute the routing tables.
4. Convert the routing table information into a **routing decision** and then *dispatching* the data packet.

Adaptive Routing

Design Issues:

1. How much **overhead** is incurred due to gathering the routing information and sending *routing packets*?
2. What is the time frame (i.e, the frequency) for sending *routing packets* in support of adaptive routing?
3. What is the **complexity** of the routing strategy?

Distance Vector Routing

- Historically known as the *old* ARPANET routing algorithm {or known as *Bellman-Ford algorithm*}.

Basic idea: each network node maintains a Distance Vector table containing the *distance* between itself and ALL possible destination nodes.

- Distances are based on a chosen metric and are computed using information from the **neighbors'** distance vectors.

Metric: *usually hops or delay*

Distance Vector Routing

Information kept by DV router

1. each router has an ID
2. associated with each link connected to a router, there is a link cost (static or dynamic) **the metric issue!**

Distance Vector Table Initialization

Distance to itself = 0

Distance to ALL other routers = infinity number

Distance Vector Algorithm

[Perlman]

1. Router transmits its *distance vector* to each of its neighbors.
 2. Each router receives and saves the most recently received *distance vector* from each of its neighbors.
 3. A router **recalculates** its distance vector when:
 - a. It receives a *distance vector* from a neighbor containing different information than before.
 - b. It discovers that a link to a neighbor has gone down (i.e., a topology change).
- The DV calculation is based on minimizing the cost to each destination.

Distance Vector Routing

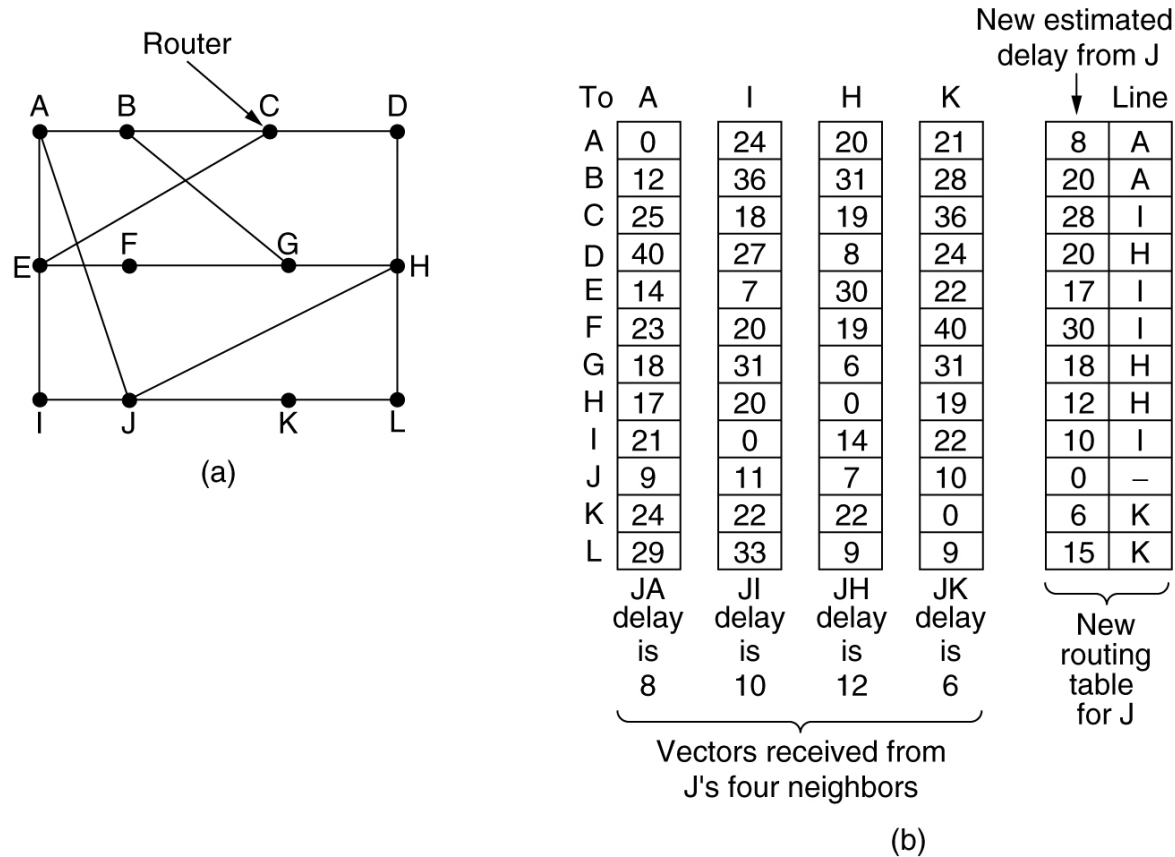


Figure 5-9.(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Routing Information Protocol (RIP)

- RIP had widespread use because it was distributed with BSD Unix in “*routed*”, a *router management daemon*.
- **RIP** is the most used Distance Vector protocol.
- RFC1058 in June 1988.
- Sends packets every 30 seconds or faster.
- Runs over UDP.
- Metric = hop count
- BIG problem is max. hop count =16
→ RIP limited to running on small networks!!
- Upgraded to RIPv2

Link State Algorithm

1. Each router is responsible for meeting its neighbors and learning their names.
2. Each router constructs a **link state packet (LSP)** which consists of a list of names and cost to reach each of its neighbors.
3. The **LSP** is transmitted to ***ALL other routers***. Each router stores the most recently generated **LSP** from each other router.
4. Each router uses complete information on the network topology to compute the ***shortest path route*** to each destination node.

Open Shortest Path First (OSPF)

- OSPF runs *on top of* IP, i.e., an OSPF packet is transmitted with IP data packet header.
- Uses Level 1 and Level 2 routers
- Has: backbone routers, area border routers, and AS boundary routers
- LSPs referred to as **LSAs (Link State Advertisements)**
- Complex algorithm due to **five** distinct LSA types.

OSPF Terminology

Internal router :: a level 1 router.

Backbone router :: a level 2 router.

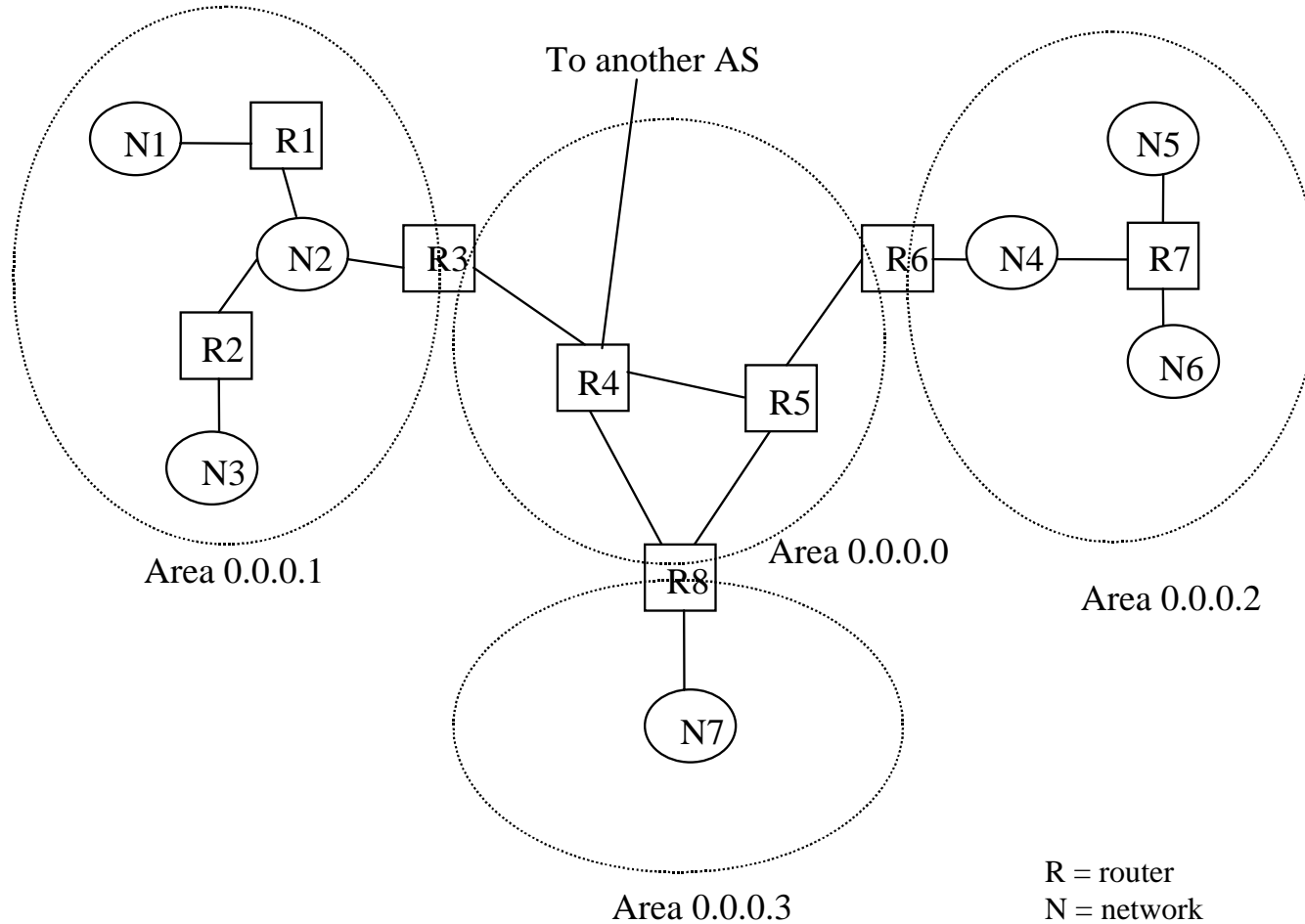
Area border router :: a backbone router that attaches to more than one area.

AS boundary router :: (an interdomain router), namely, a router that attaches to routers from other ASs.

OSPF LSA Types

1. Router link advertisement
2. Network link advertisement
3. Network summary link advertisement
4. AS boundary routers summary link advertisement
5. AS external link advertisement

OSPF Areas



OSPF

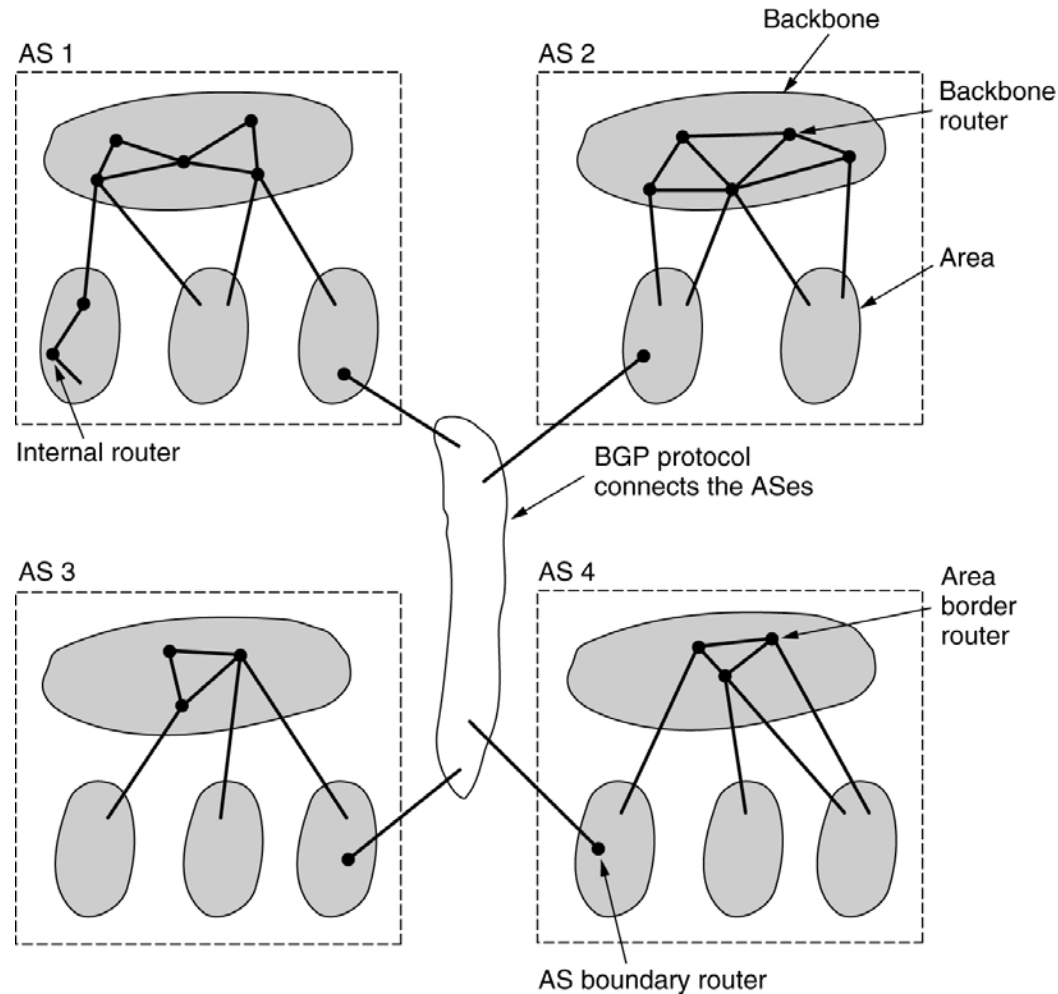


Figure 5-65. The relation between ASes, backbones, and areas in OSPF.

Border Gateway Protocol (BGP)

- The replacement for EGP is BGP. Current version is BGP-4.
- BGP assumes the Internet is an arbitrary interconnected set of AS's.
- In *interdomain routing* the goal is to find ANY path to the intended destination that is loop-free. The protocols are more concerned with **reachability** than optimality.