Network Layer

Routing

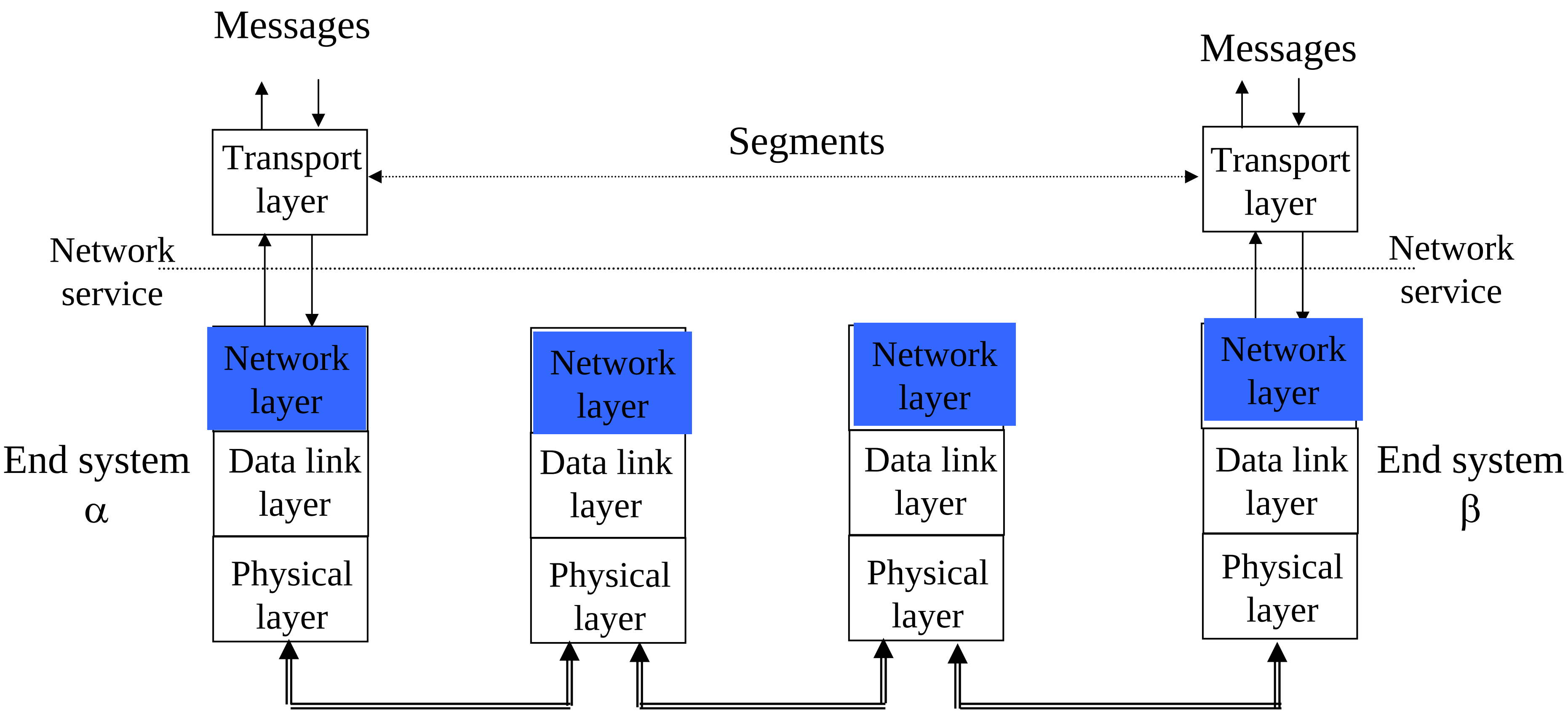
Network Layer

- 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).

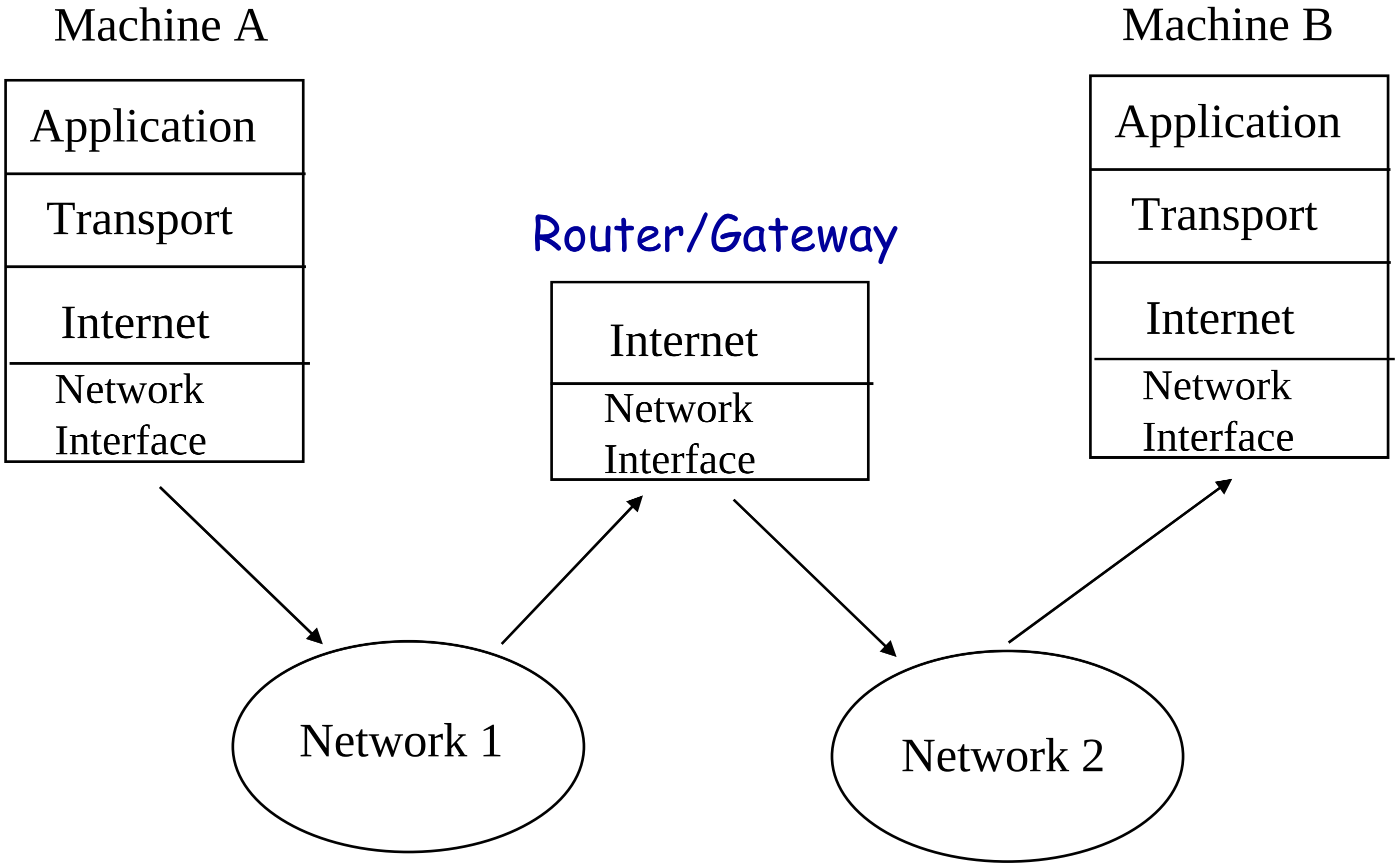
Network Layer Design Goals

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).



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.2



Machine A

- Application
- Transport
- Internet
- Network Interface

Router/Gateway

- Internet
- Network Interface

Machine B

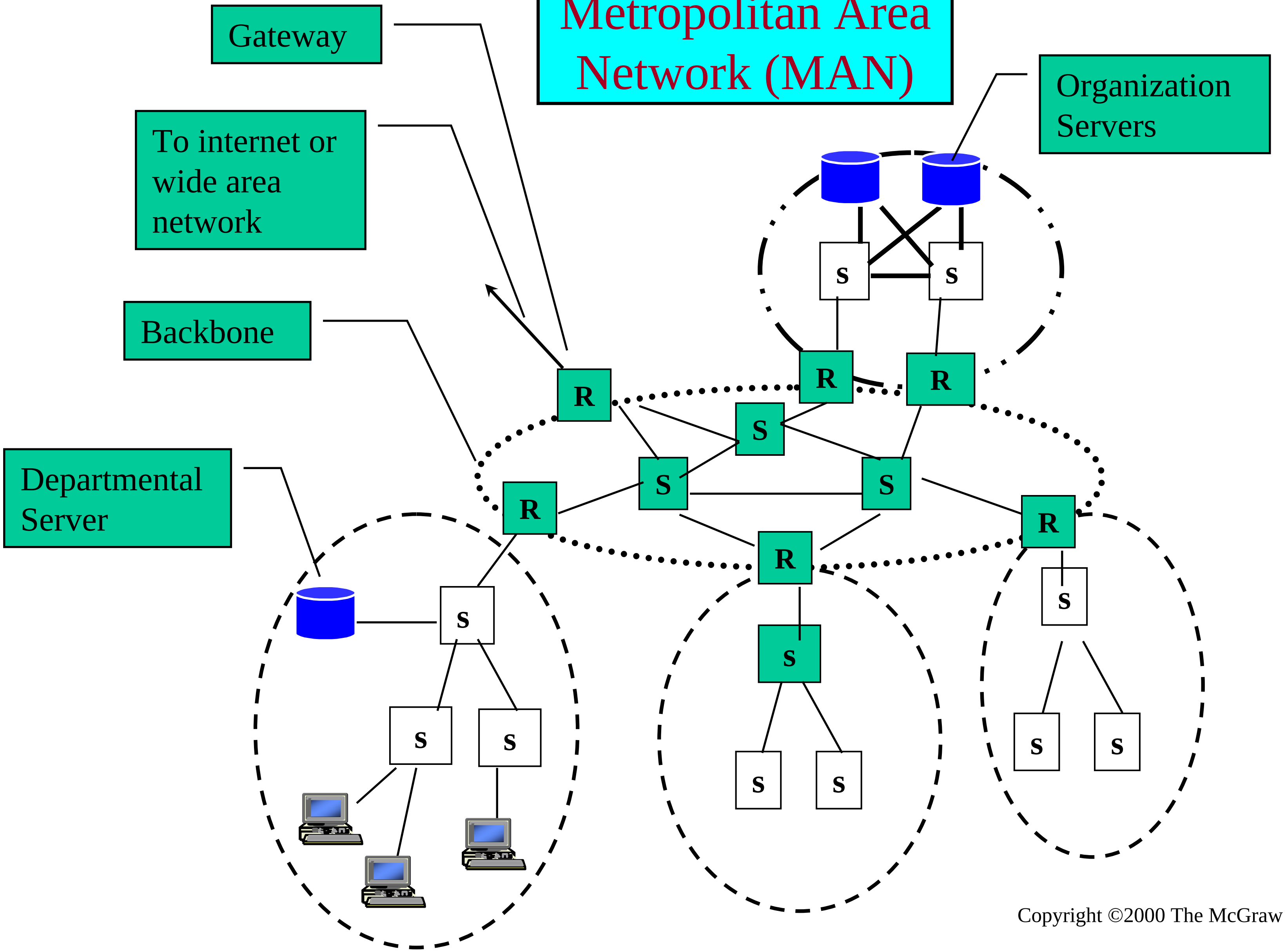
- Application
- Transport
- Internet
- Network Interface

Network 1

Network 2

Figure 8.3

Metropolitan Area Network (MAN)



Gateway

To internet or wide area network

Backbone

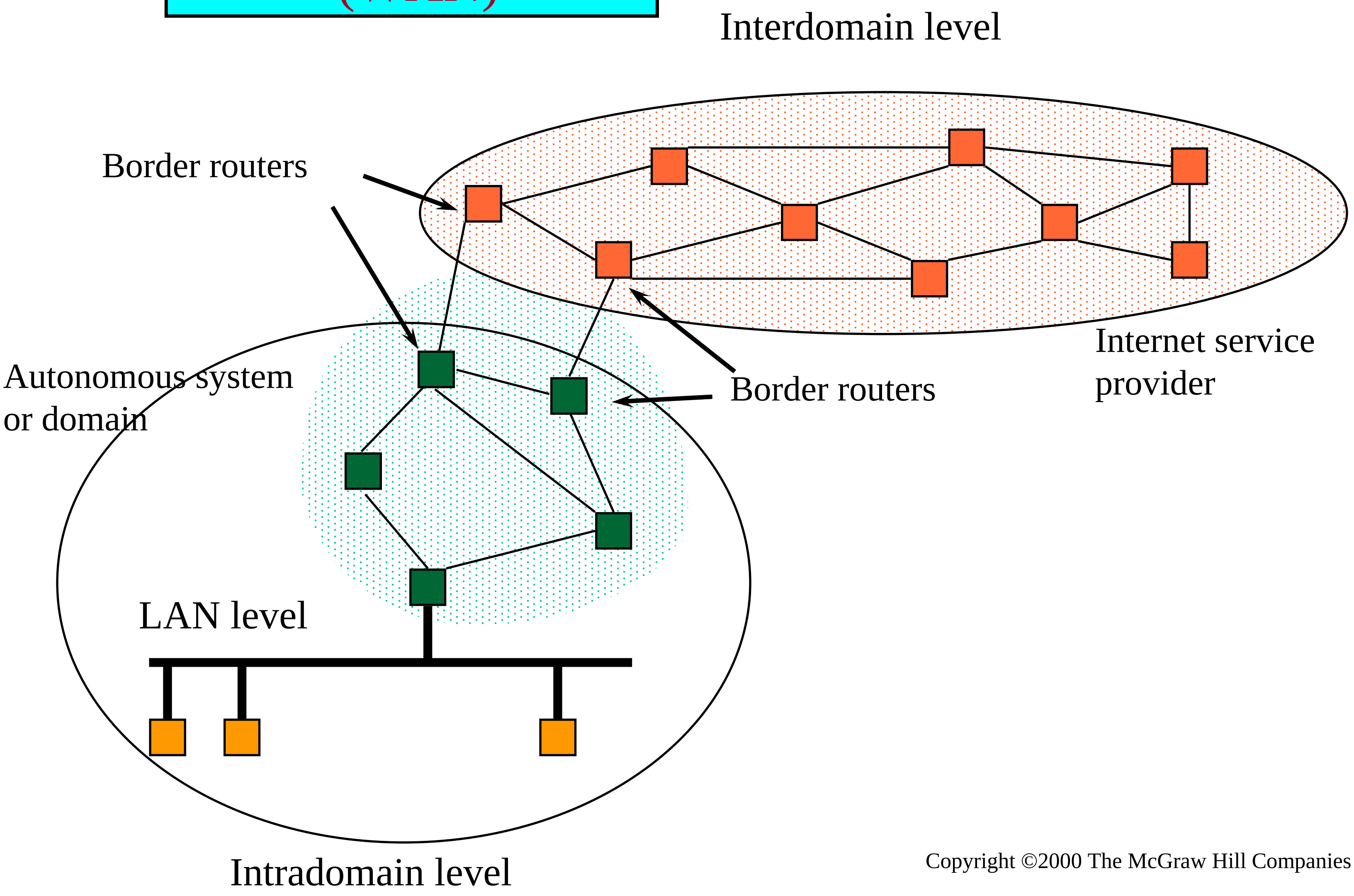
Departmental Server

Organization Servers

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.6

Wide Area Network (WAN)



Interdomain level

Border routers

Autonomous system or domain

Border routers

Internet service provider

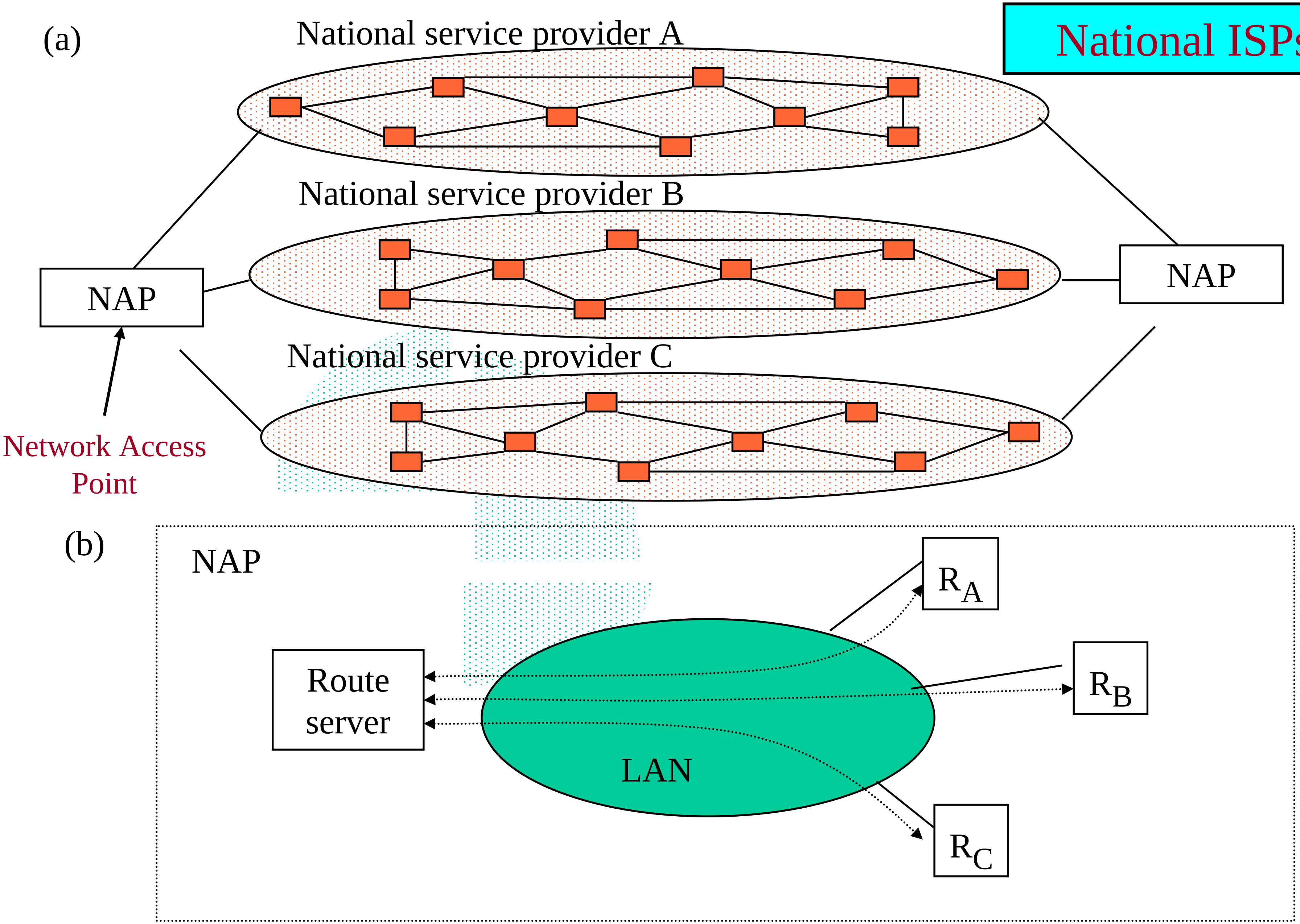
LAN level

Intradomain level

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.7

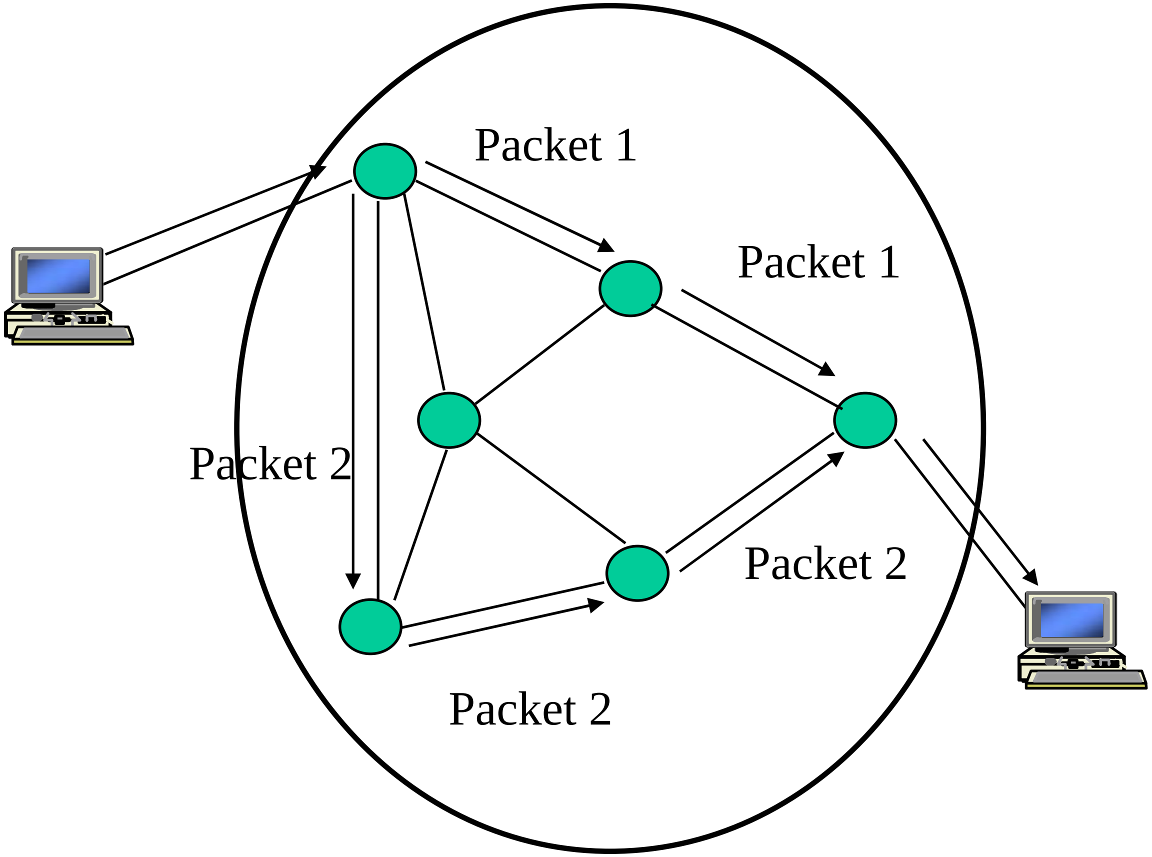
National ISPs



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.8

Datagram Packet Switching



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.15

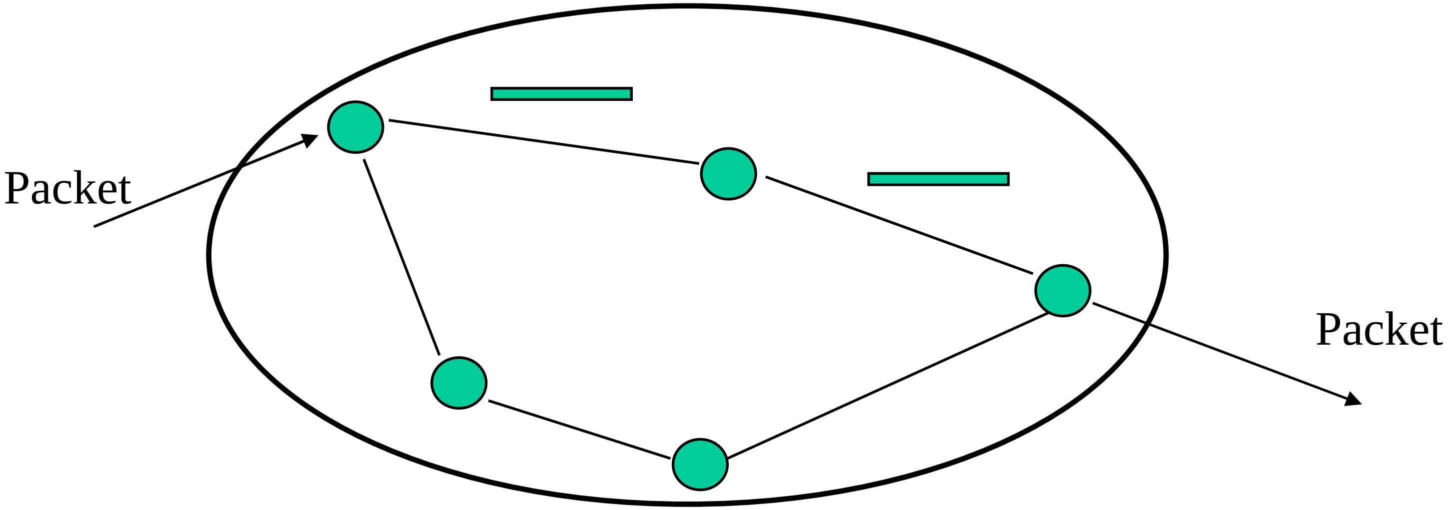
Routing Table in Datagram Network

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

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.16

Virtual Circuit Packet Switching



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.17

Routing Table
in Virtual Circuit Network

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

Entry for packets with identifier 15 →

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.21

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?

Choices:

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

Metrics

- Set all link costs to 1.
  - Shortest hop routing.
  - Disregards delay.
- Measure the number of packets queued.
  - Did not work well.
- Timestamp ArrivalTime and DepartTime* and use link-level ACK to compute:

  Delay = (DepartTime - ArrivalTime) + TransmissionTime + Latency

* Reset after retransmission

Metrics

  - Unstable under heavy link load.
  - Difficulty with granularity of the links.
- Revised ARPANET routing metric:
  - Compress dynamic range of the metric
  - Account for link type
  - Smooth variation of metric with time:
    - Delay transformed into link utilization
    - Utilization was averaged with last reported utilization.
    - Hard limit set on how much the metric could change per measurement cyle.

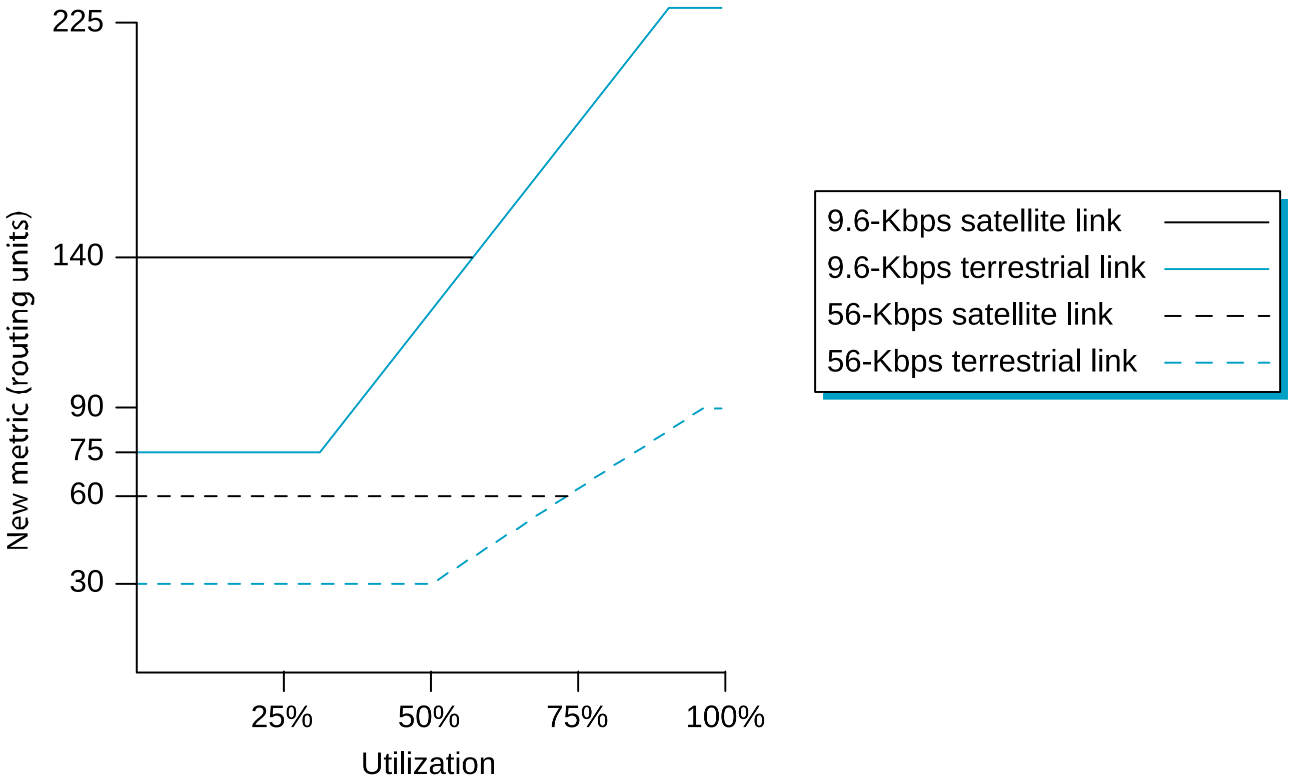


Figure 4.22 Revised ARPANET routing metric versus link utilization

P&D slide

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

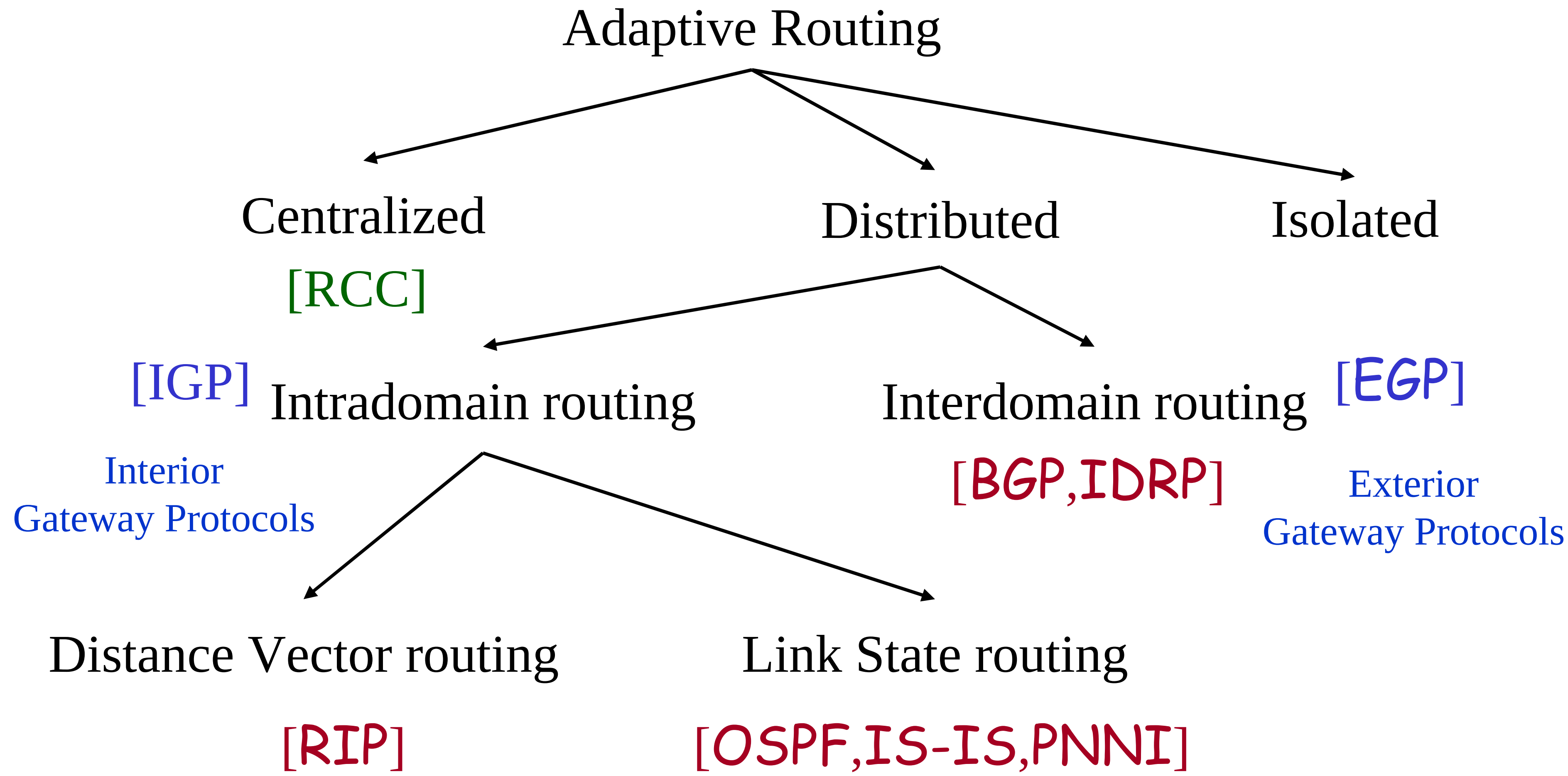
1. Mark the working node as permanent.
2. Examine all adjacent nodes in turn

   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

Interior Gateway Protocols

Interdomain routing [EGP]

[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 dispatch 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

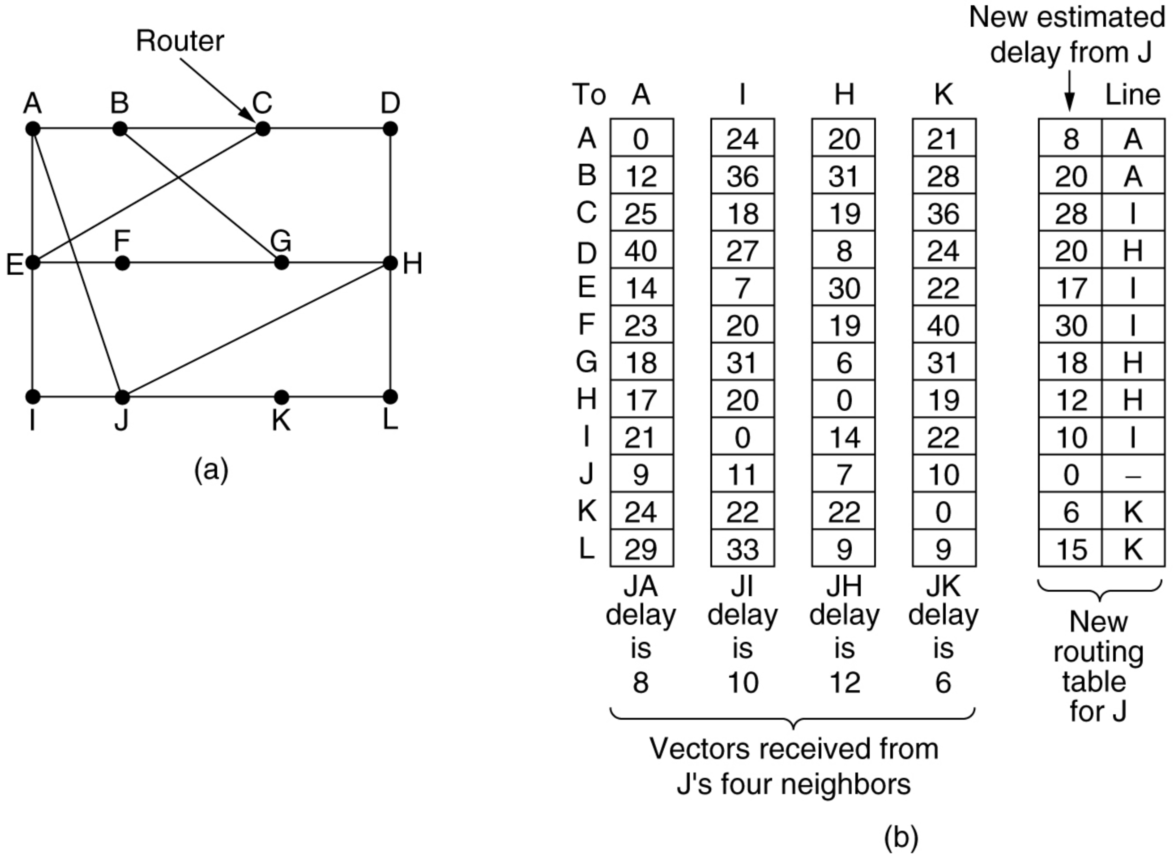
Distance to ALL other routers = infinity number

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



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

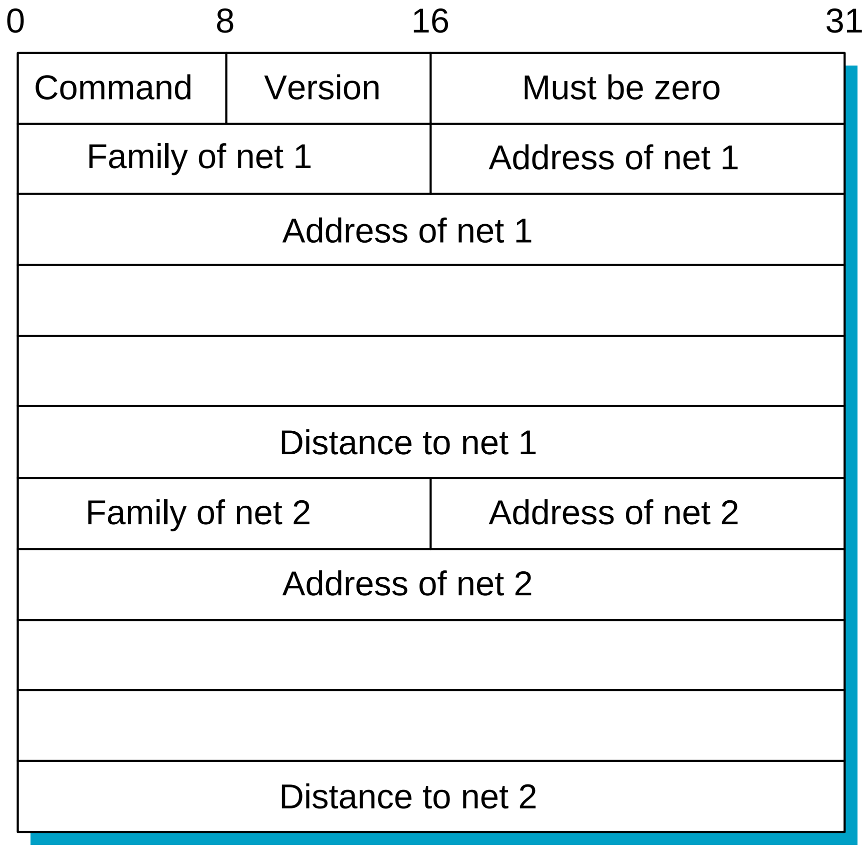
Figure 5-9.(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

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

(network_address, distance) pairs



Command	Version	Must be zero
Family of net 1		Address of net 1
Address of net 1		
Distance to net 1		
Family of net 2		Address of net 2
Address of net 2		
Distance to net 2		

Figure 4.17 RIP Packet Format

P&D slide

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.

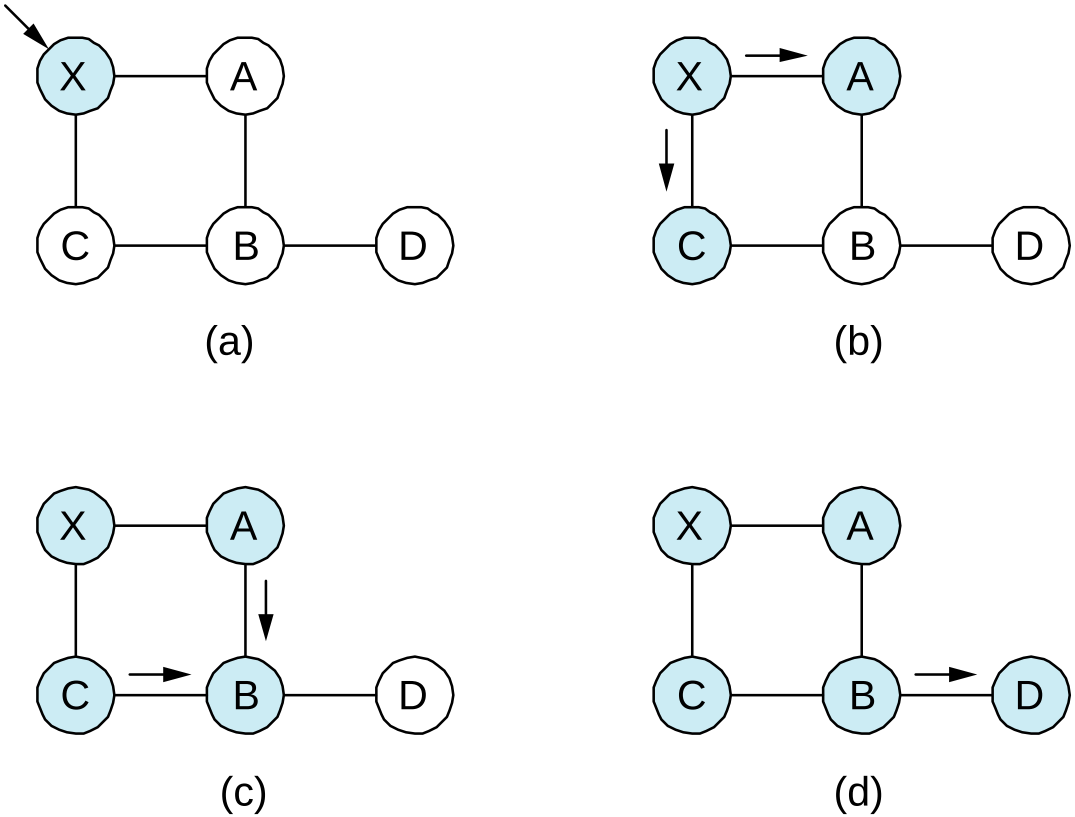


Figure 4.18 Reliable LSP Flooding

P&D slide

Reliable Flooding

- The process of making sure all the nodes participating in the routing protocol get a copy of the link-state information from all the other nodes.
- LSP contains:
  - Sending router's node ID
  - List connected neighbors with the associated link cost to each neighbor
  - Sequence number
  - Time-to-live

Reliable Flooding

- First two items enable route calculation
- Last two items make process reliable
  - ACKs and checking for duplicates is needed.
- Periodic Hello packets used to determine the demise of a negihbor
- The sequence numbers are not expected to wrap around.
  - => field needs to be large (64 bits)

Open Shortest Path First (OSPF)

- Provides for authentication of routing messages.
  - 8-byte password designed to avoid misconfiguration.
- Provides additional hierarchy
  - Domains are partitioned into **areas.**
  - This reduces the amount of information transmitted in packet.
- Provides load-balancing via multiple routes.

Open Shortest Path First (OSPF)

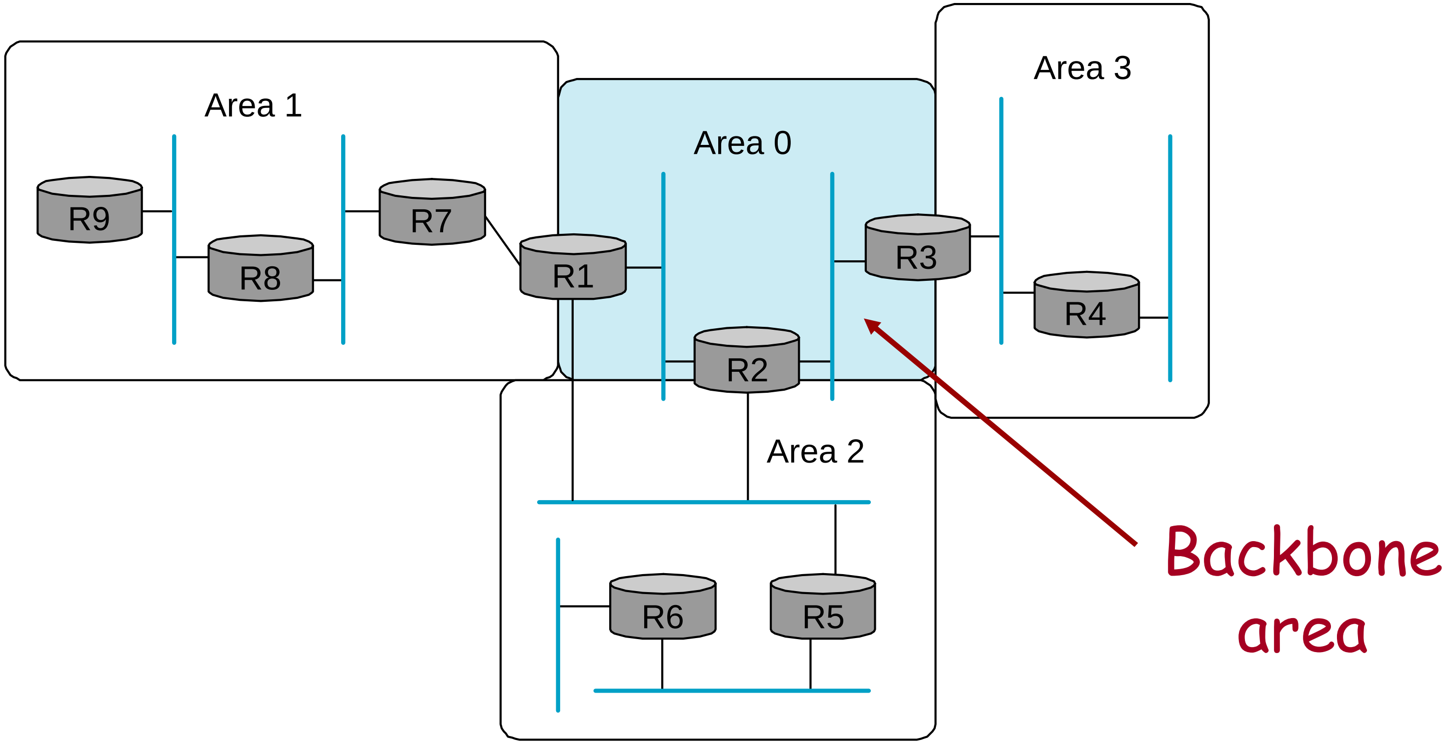


Figure 4.32 A Domain divided into Areas

P&D slide

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.

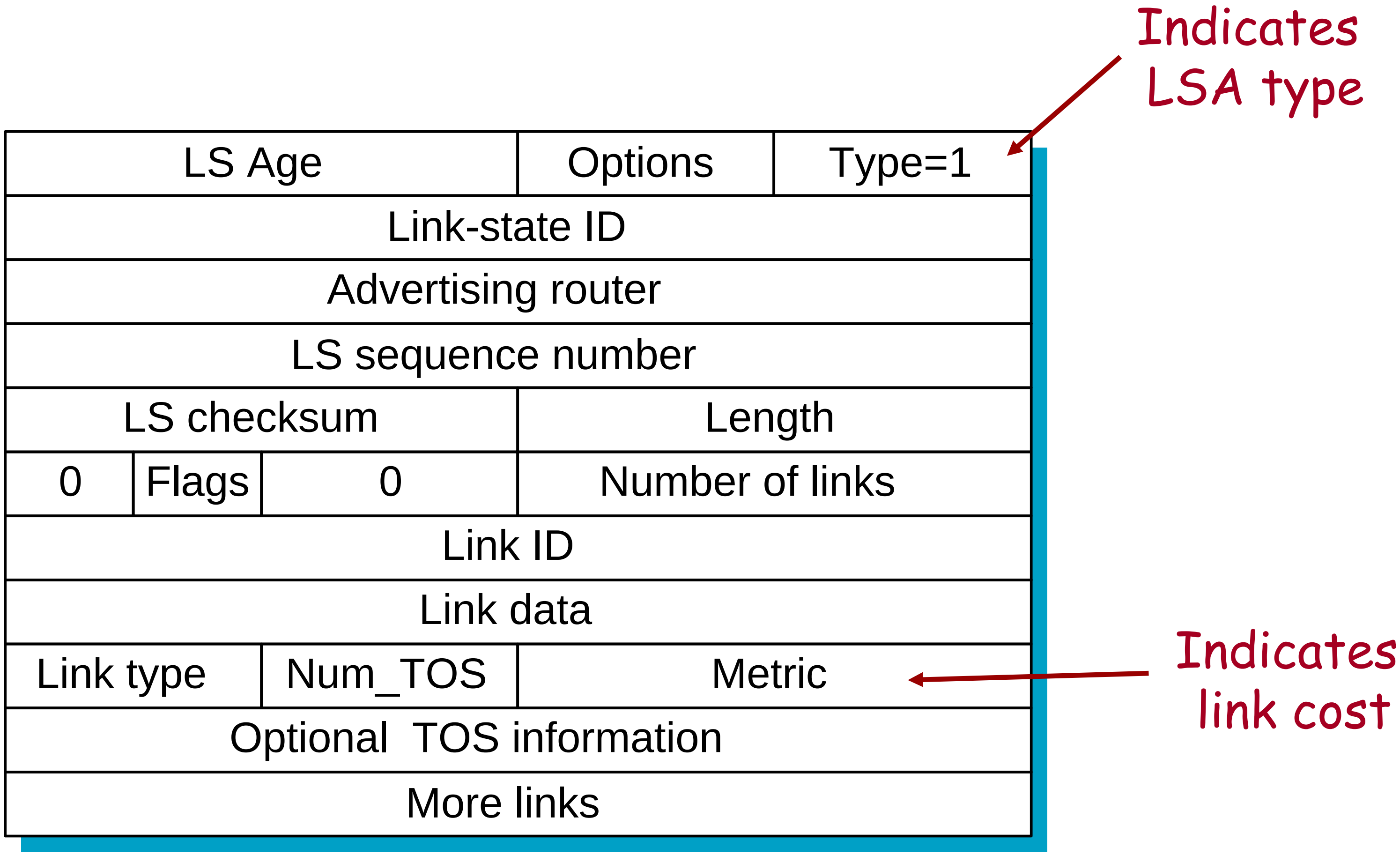
Backbone router :: a level 2 router.

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

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

OSPF LSA Types

1. Router link advertisement [Hello message]
2. Network link advertisement
3. Network summary link advertisement
4. AS border router’s summary link advertisement
5. AS external link advertisement



<table>
<tr><td colspan="3">LS Age</td><td>Options</td><td>Type=1</td></tr>
<tr><td colspan="5">Link-state ID</td></tr>
<tr><td colspan="5">Advertising router</td></tr>
<tr><td colspan="5">LS sequence number</td></tr>
<tr><td colspan="3">LS checksum</td><td colspan="2">Length</td></tr>
<tr><td>0</td><td>Flags</td><td>0</td><td colspan="2">Number of links</td></tr>
<tr><td colspan="5">Link ID</td></tr>
<tr><td colspan="5">Link data</td></tr>
<tr><td colspan="2">Link type</td><td>Num_TOS</td><td colspan="2">Metric</td></tr>
<tr><td colspan="5">Optional TOS information</td></tr>
<tr><td colspan="5">More links</td></tr>
</table>

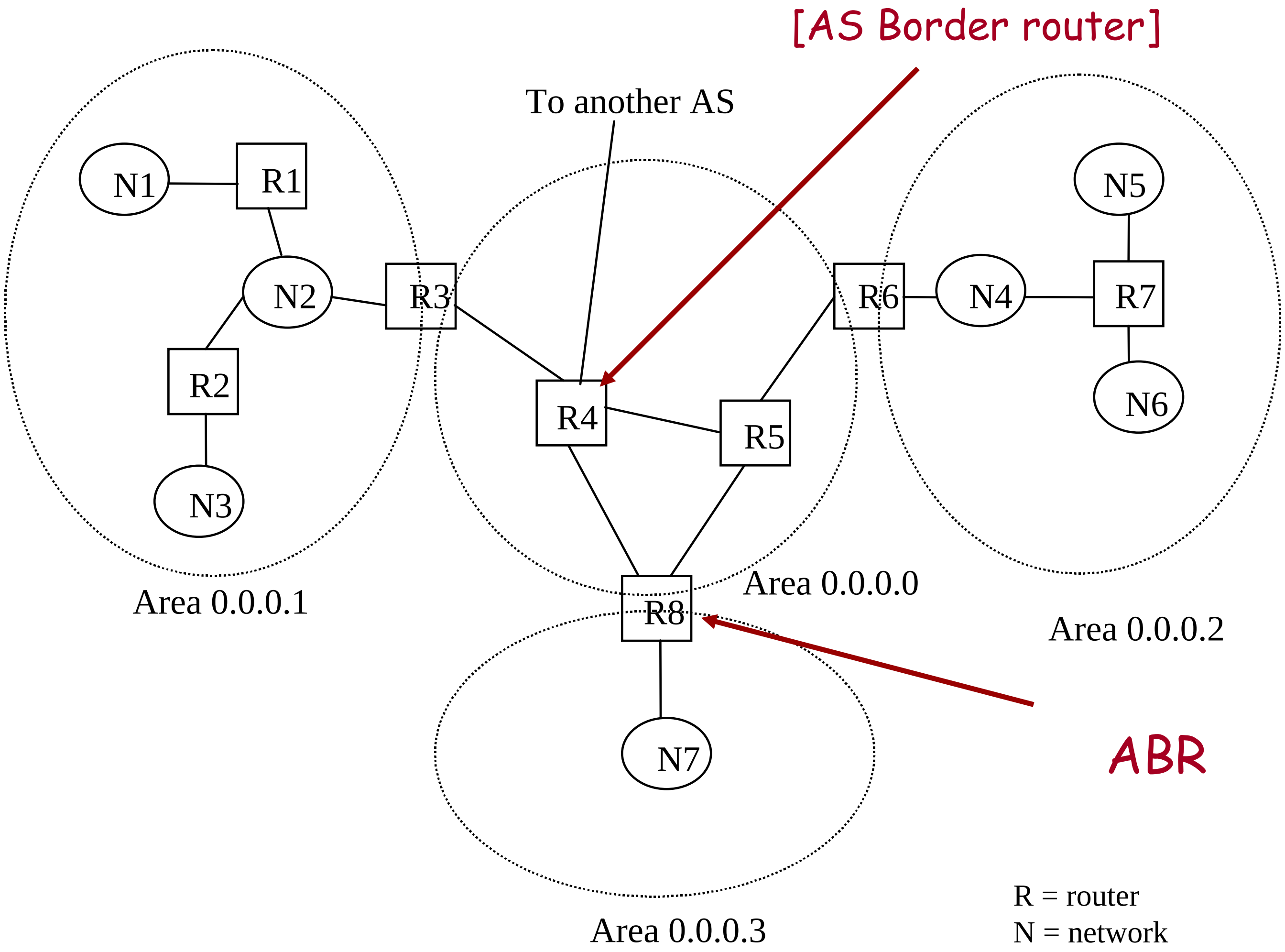
Indicates LSA type

Indicates link cost

Figure 4.21 OSF Type 1 Link-State Advertisement

P&D slide

OSPF Areas



Leon-Garcia & Widjaja: *Communication Networks*

Figure 8.33

OSPF

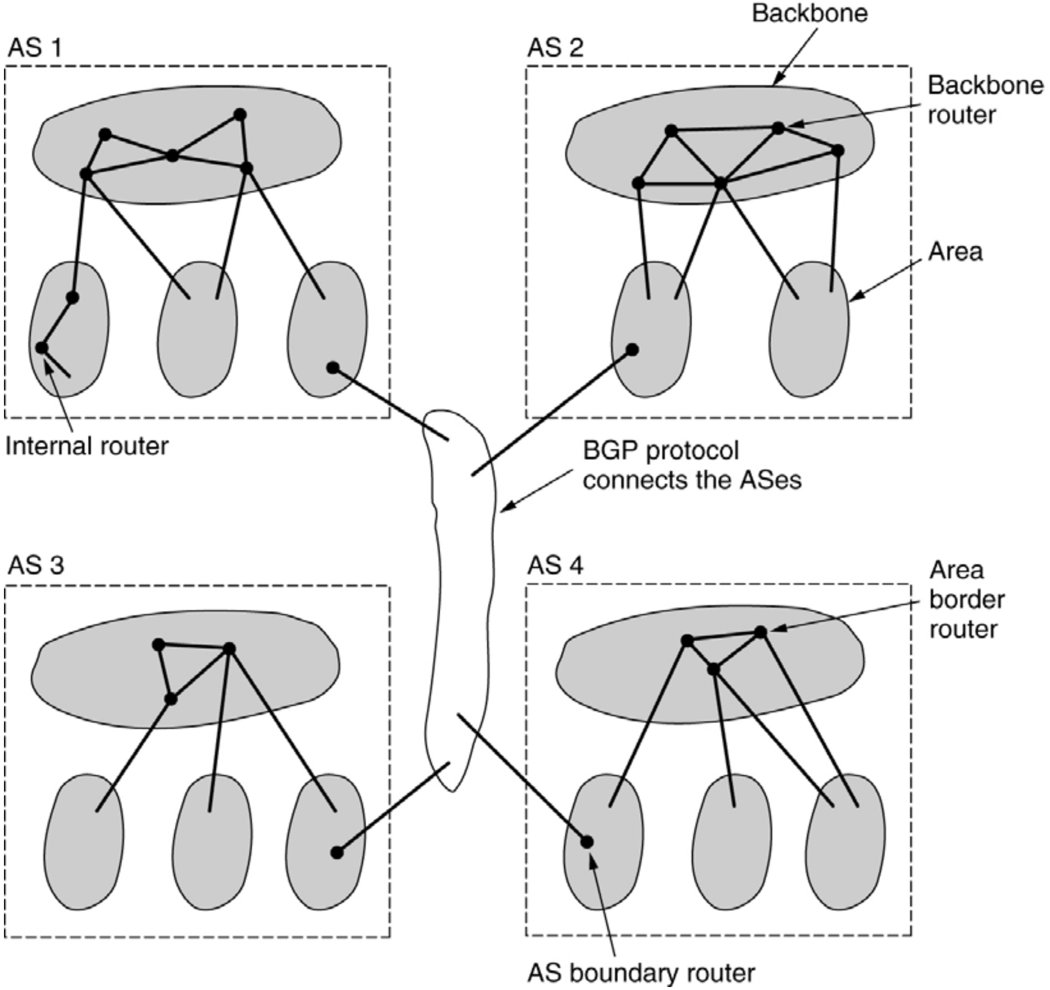


Figure 5-65.The relation between ASes, backbones, and areas in OSPF.

Tanenbaum slide

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.