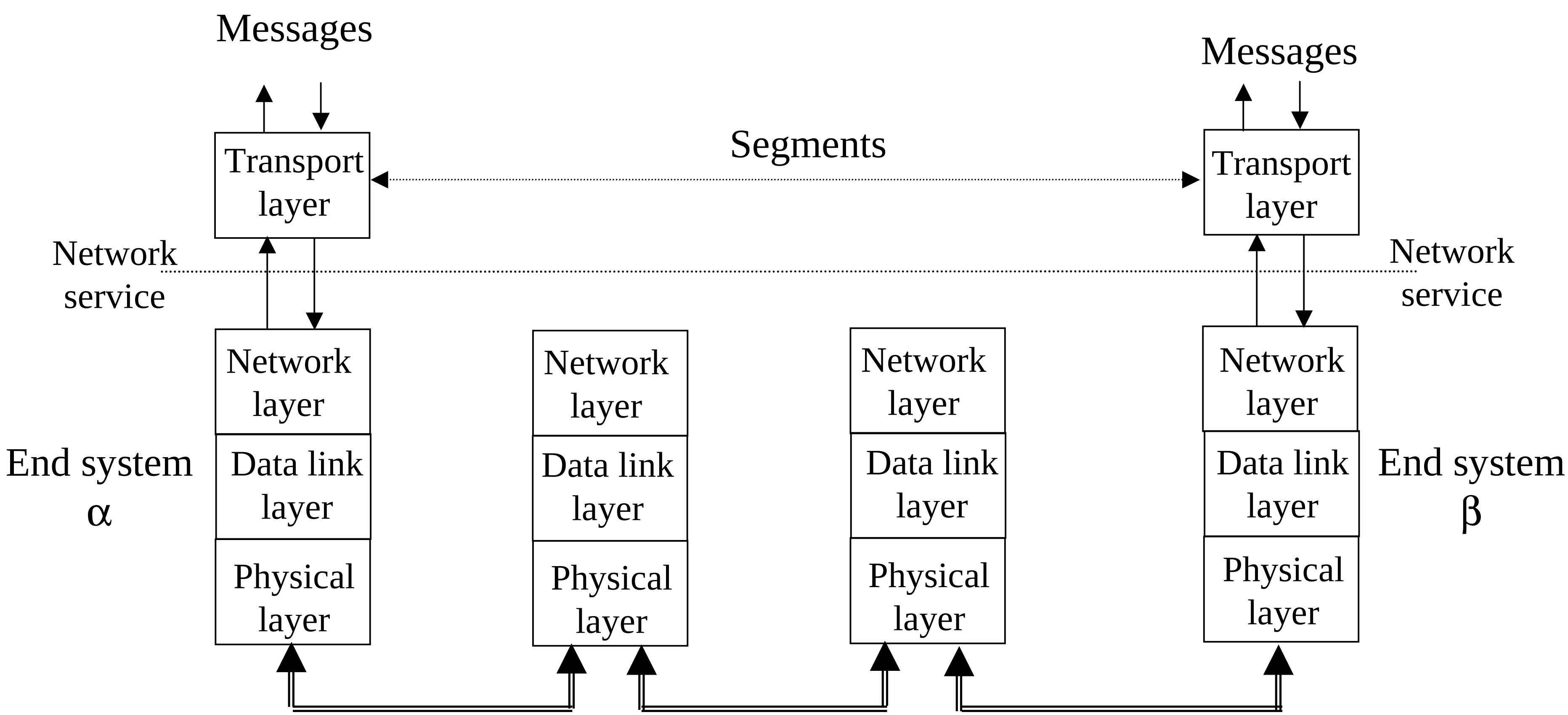
Network Layer

Routing

Network Layer

- Concerned with getting packets from source to destination
- 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).



 Leon-Garcia & Widjaja: *Communication Networks* Figure 7.2

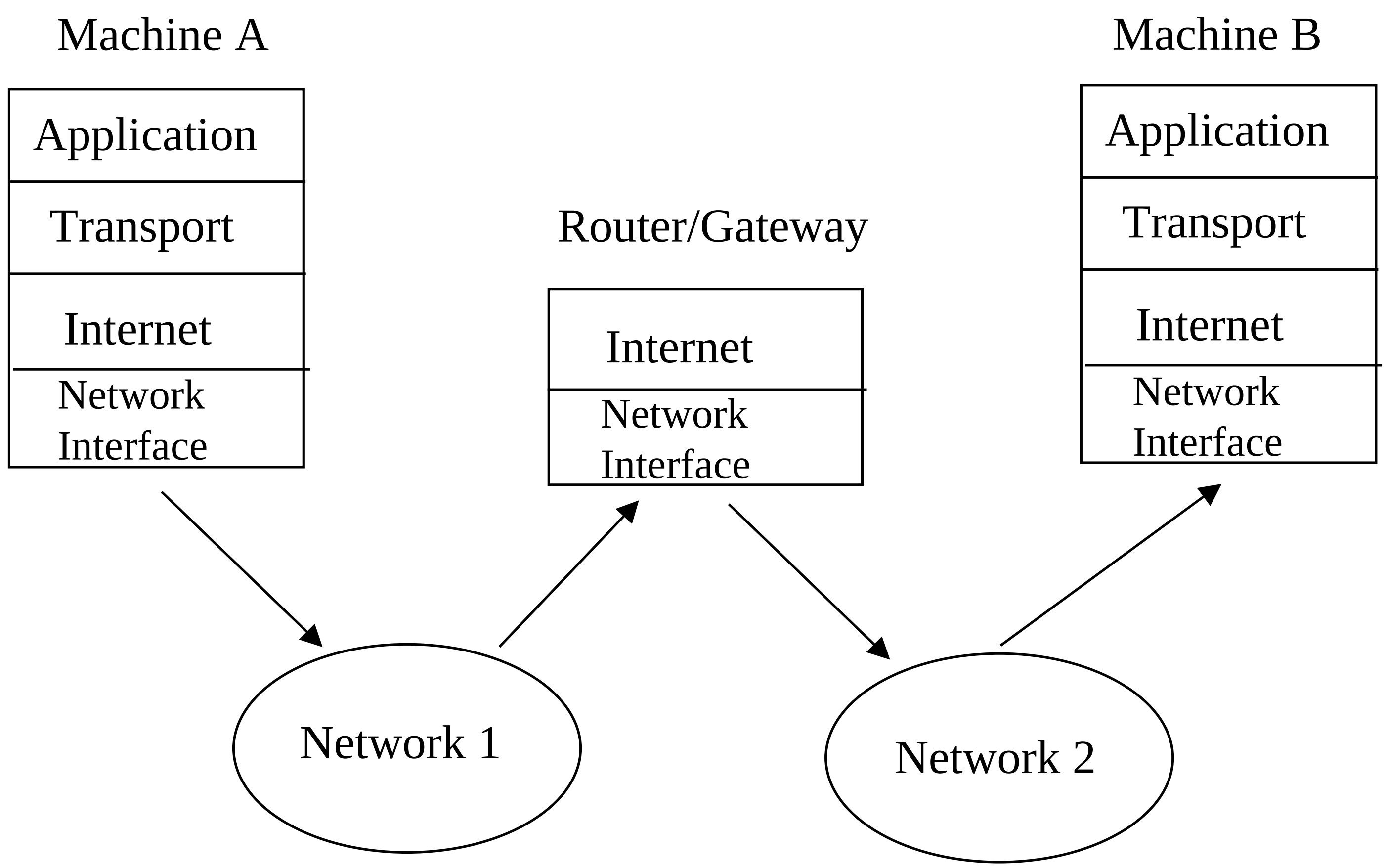
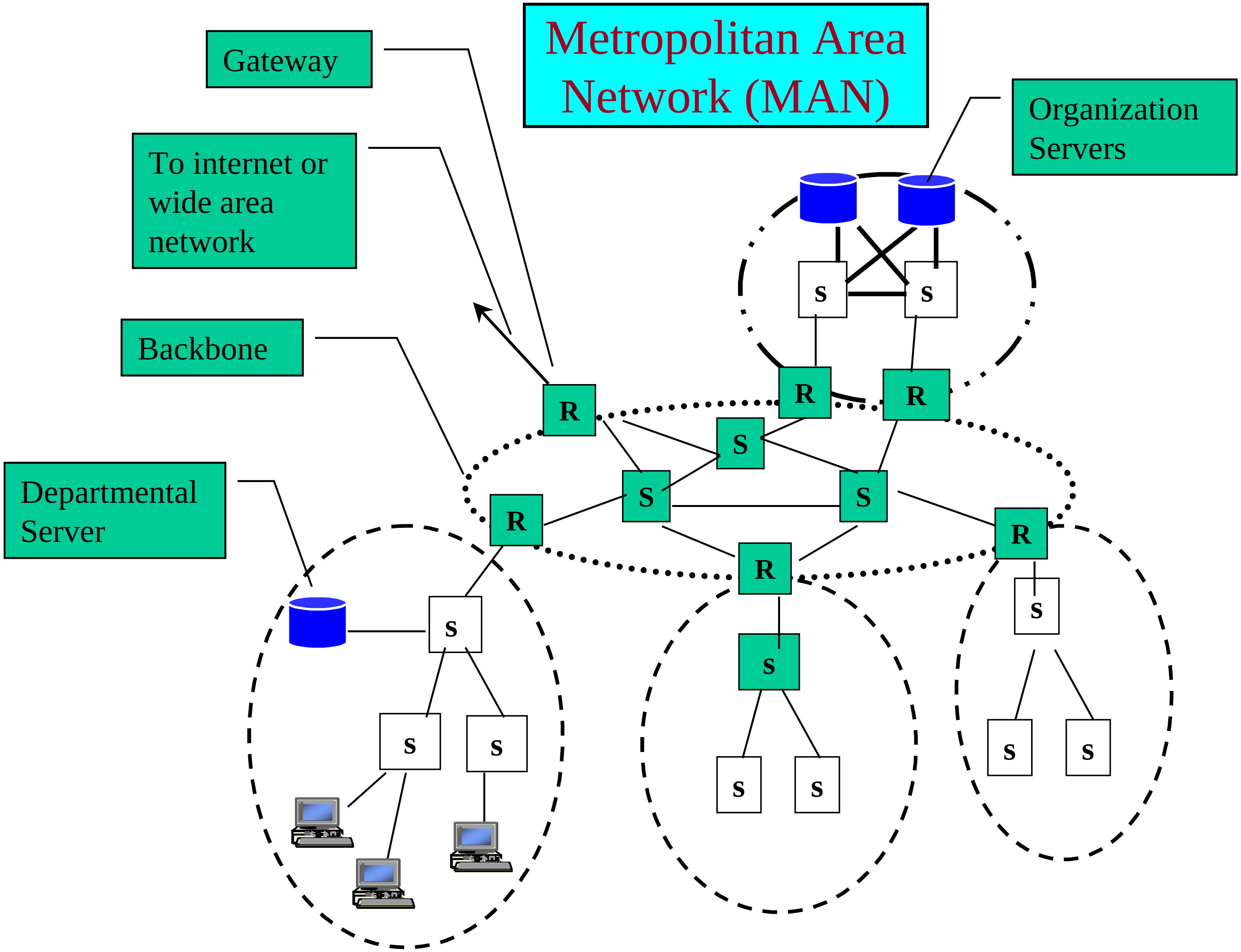

Machine A
Application
Transport
Internet
Network Interface
Router/Gateway
Internet
Network Interface
Machine B
Application
Transport
Internet
Network Interface
Network 1
Network 2

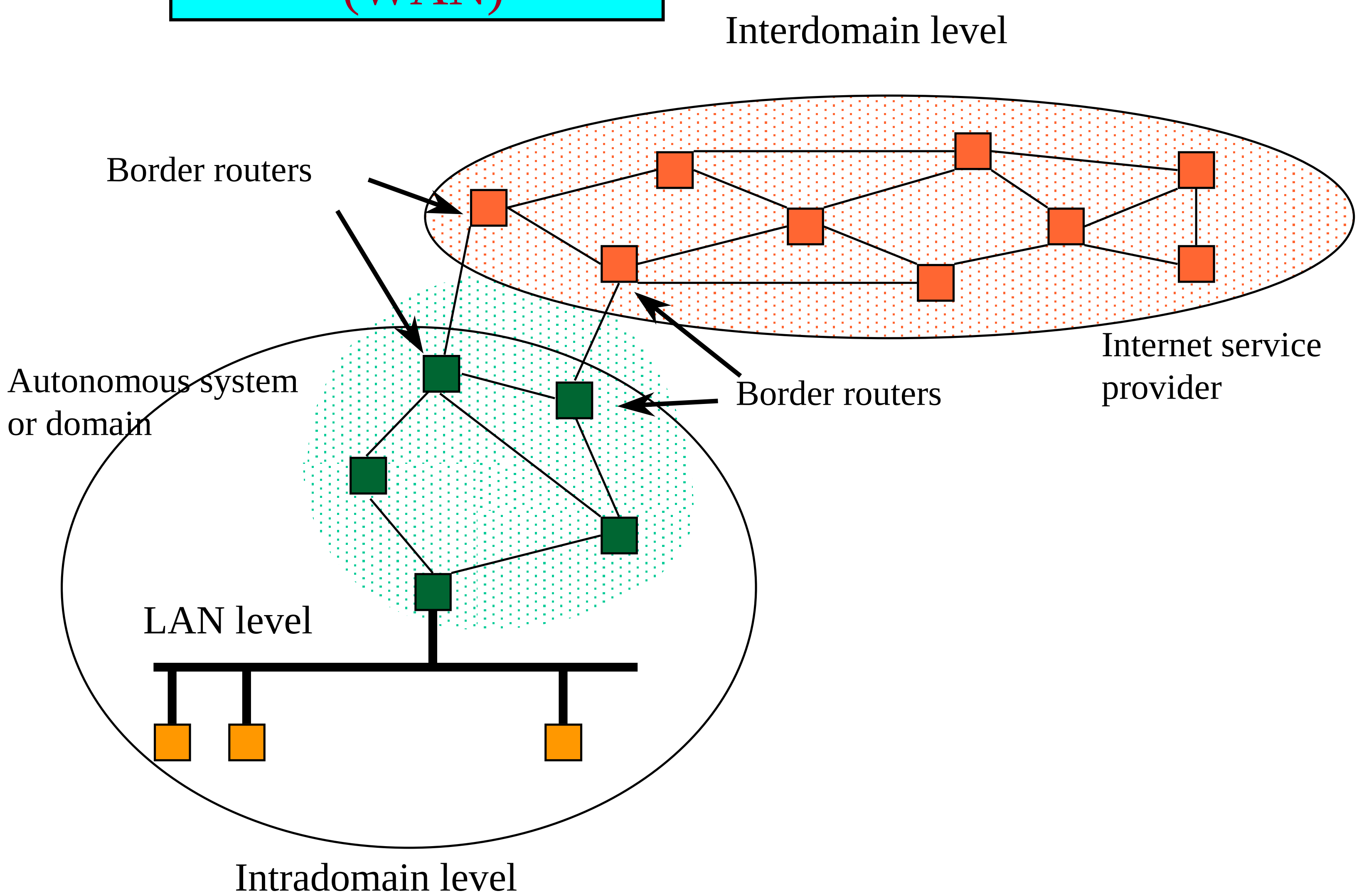
Figure 8.3



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.6

Wide Area Network (WAN)



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.7

National ISPs

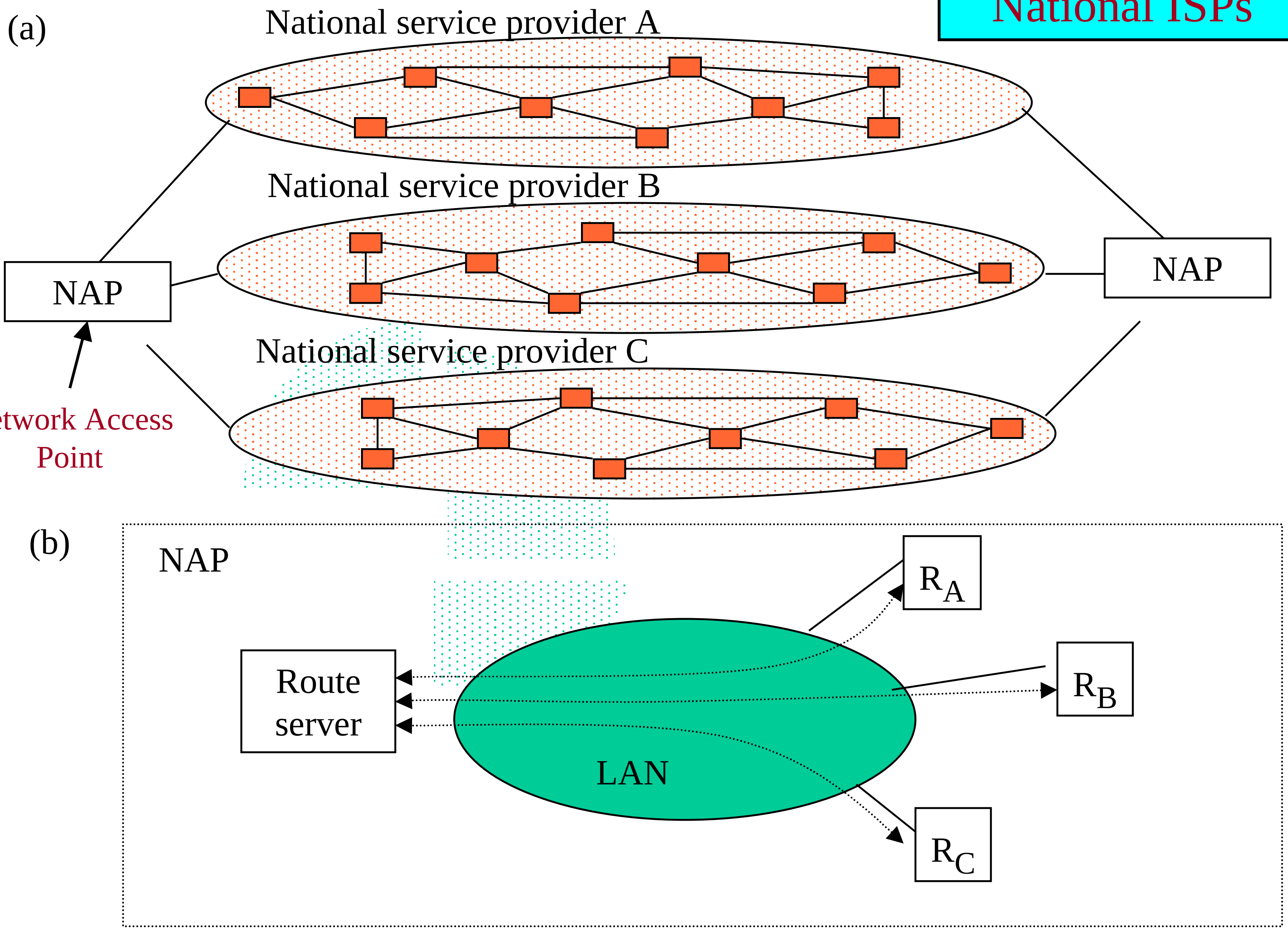
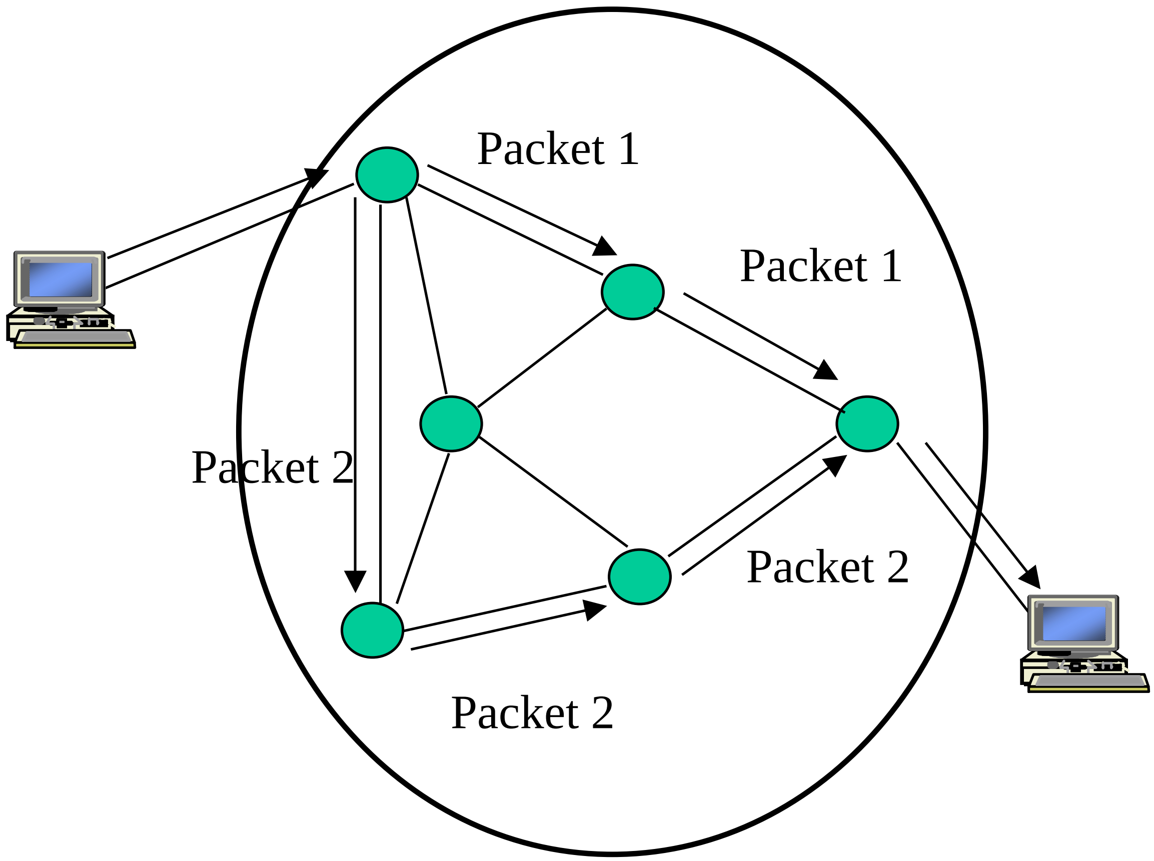


Figure 7.8

Datagram Packet Switching



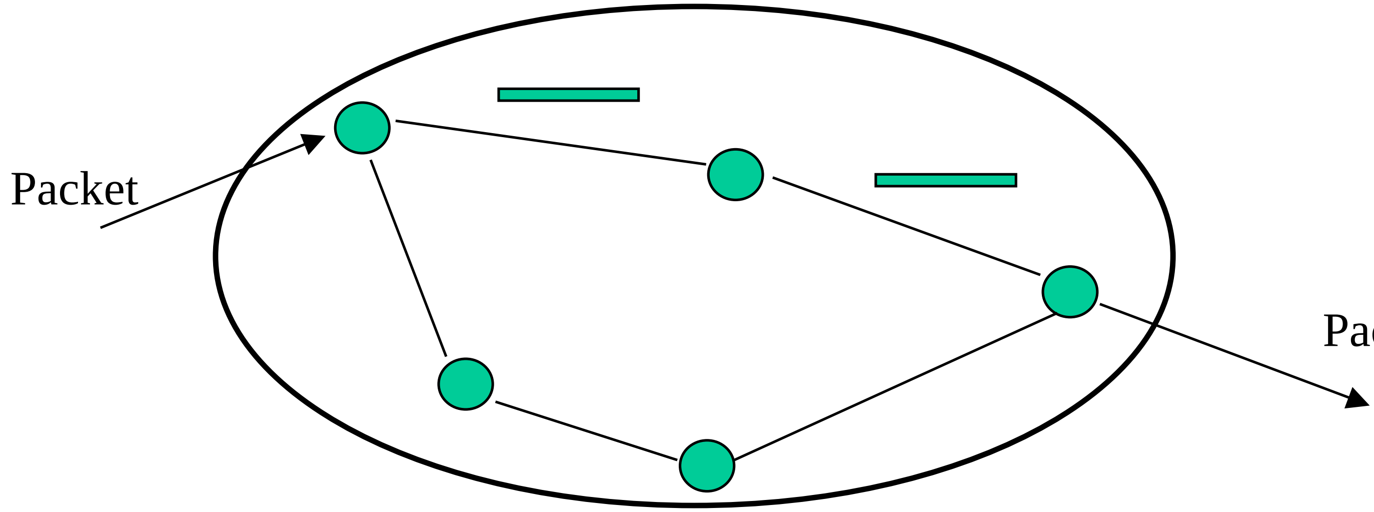
 Leon-Garcia & Widjaja: *Communication Networks* Figure 7.15

Routing Table in Datagram Network

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

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 → (row 15 | 15 | 23)

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.

Adaptive Routing – based on current measurements of traffic and/or topology.

- centralized, isolated, distributed

Non-adaptive Routing

- flooding
- static routing {shortest path}

Shortest Path Routing

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

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
    - [IGP] Intradomain routing (Interior Gateway Protocols)
      - Distance Vector routing [RIP]
      - Link State routing [OSPF,IS-IS,PNNI]
    - Interdomain routing [EGP] (Exterior Gateway Protocols) [BGP,IDRP]

Distance Vector Routing

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

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

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

Metric: *usually hops or delay*

Distance Vector

Information needed by node :

each router has an ID

associated with each link connected to a router there is a link cost (static or dynamic) *the metric issue!*

Each router starts with:

DV = 0 {distance measure to itself}

DV = infinity number {for ALL other destinations}

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.

Routing Information Protocol (RIP)

- RIP had widespread use because it was distributed with BSD Unix in *"routed", a router management daemon.*
- RFC1058 June1988.
- Sends packets every 30 seconds or faster.
- Runs over UDP.
- Metric = hop count
- BIG problem = max. hop count =16

  ➔ RIP limited to running on small networks
- Upgraded to RIPv2

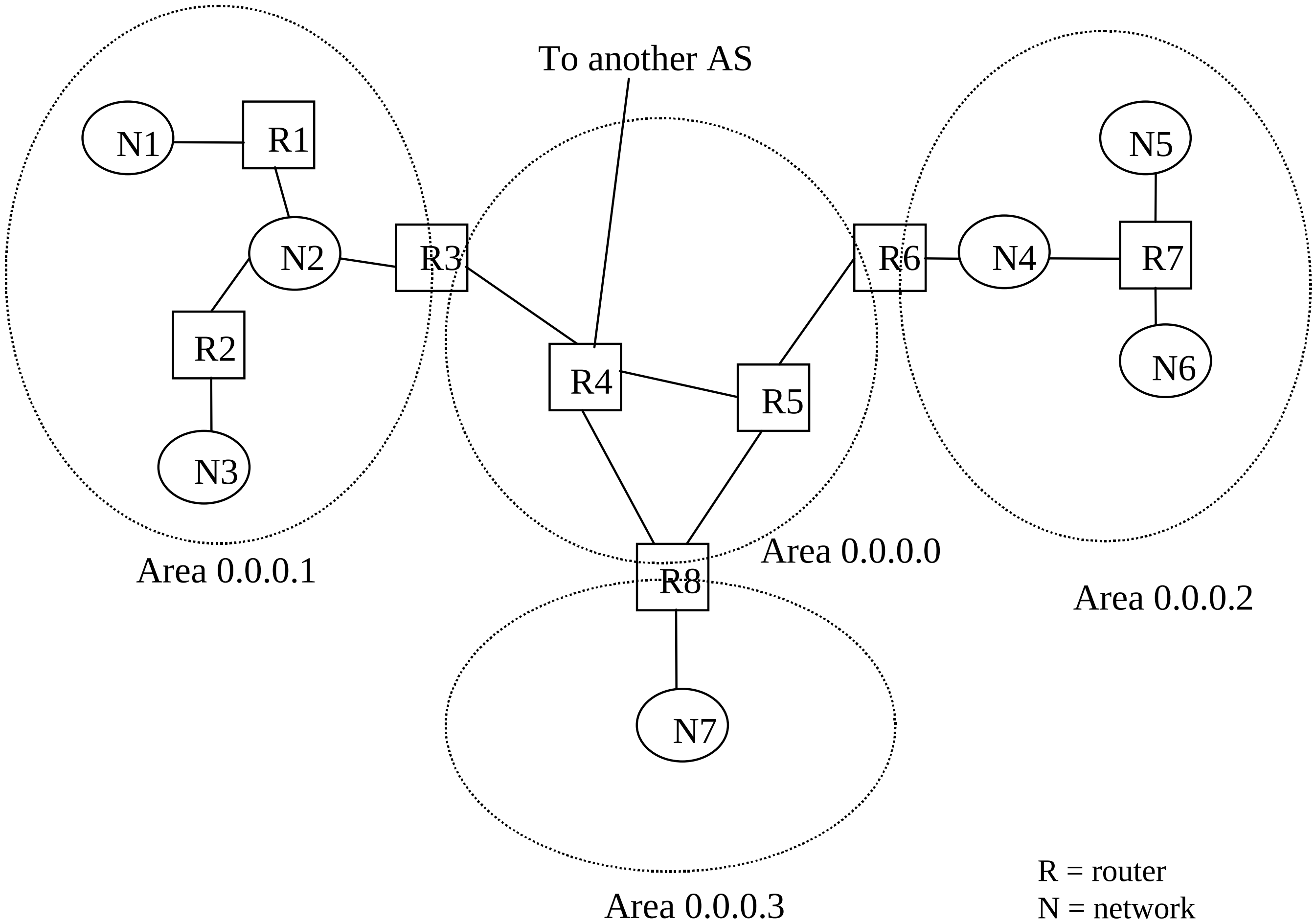
Link State Routing Algorithm

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 for 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., OSPF packet is transmitted with IP data packet header.
- 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 due to *five LSA types.*

OSPF Areas



Leon-Garcia & Widjaja: *Communication Networks*

Figure 8.33

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 Ass
- 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.