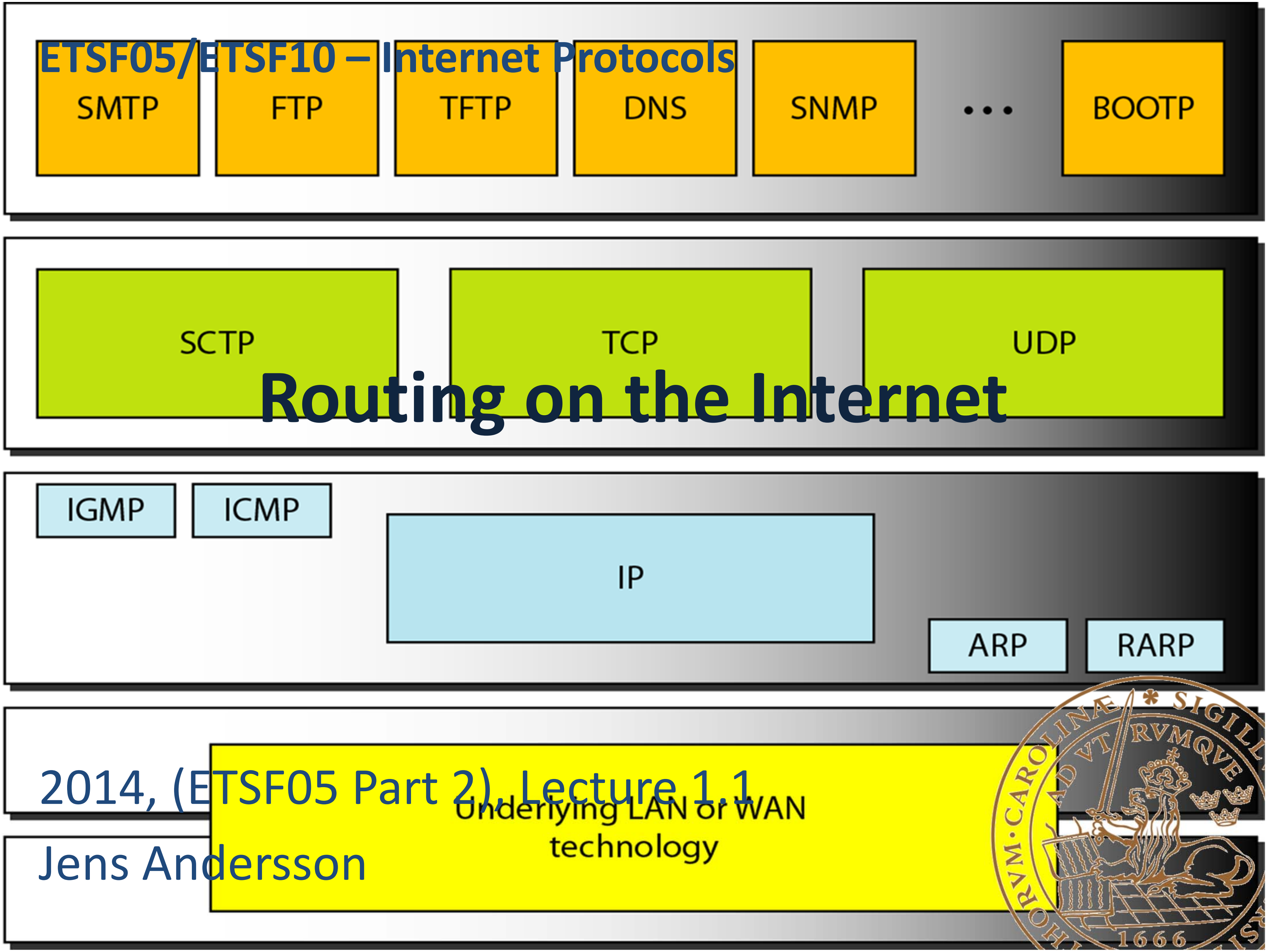
ETSF05/ETSF10 – Internet Protocols

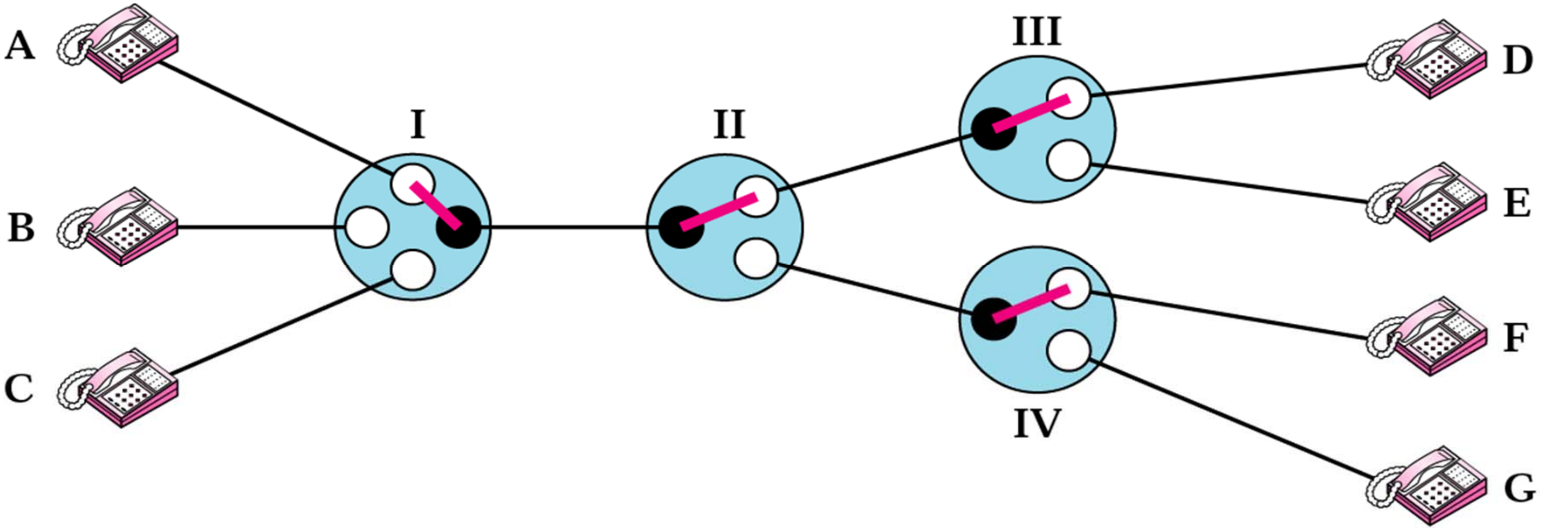
Routing on the Internet

2014, (ETSF05 Part 2), Lecture 1.1

Jens Andersson

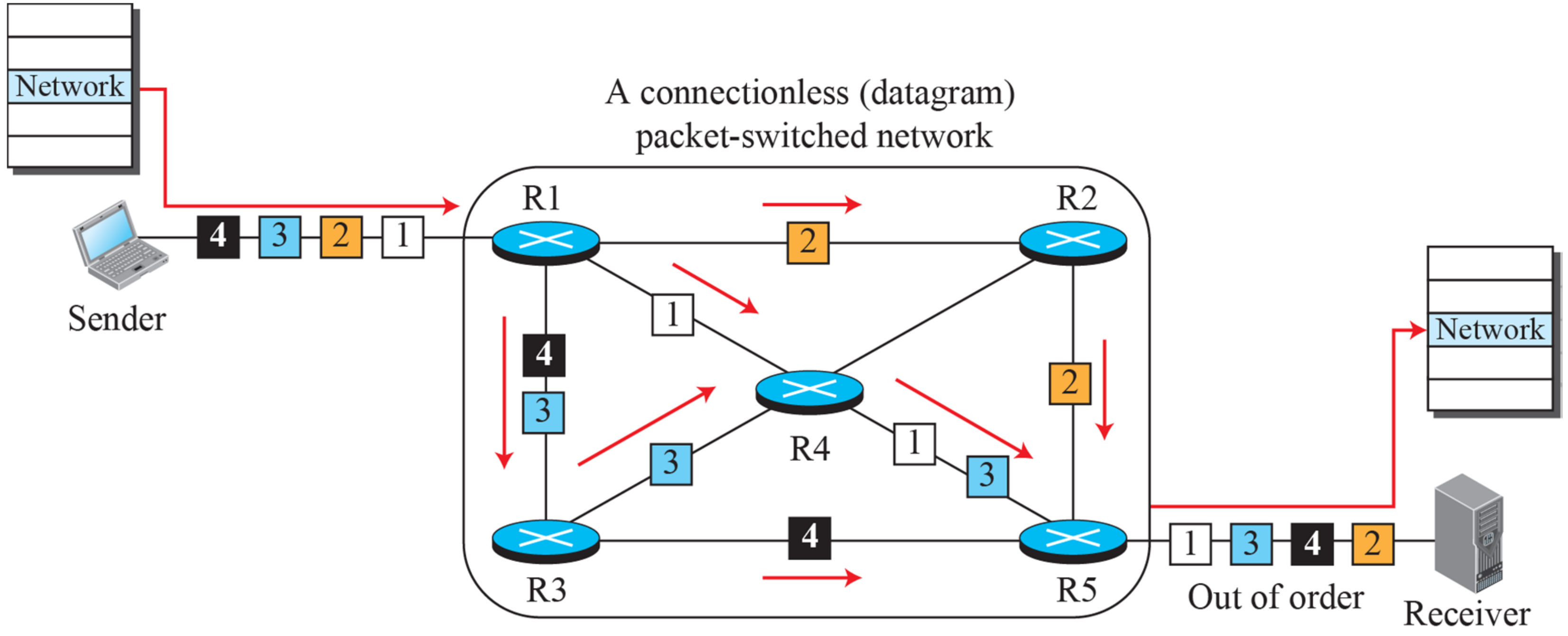


Circuit switched routing



Packet-switched Routing

- Choosing an optimal path
  - According to a cost metric
  - Decentralised: each router has full information



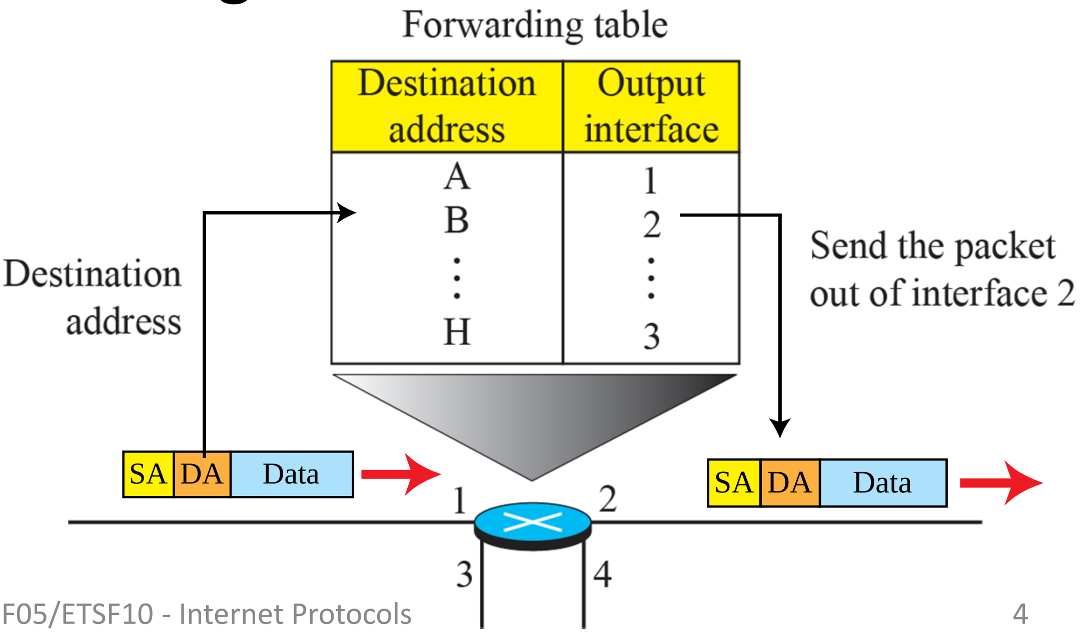
Router

- Internetworking device
  - Passes data packets between networks
  - Checks ***Network Layer*** addresses
  - Uses Routing/forwarding tables

Two functions:

❶ Routing

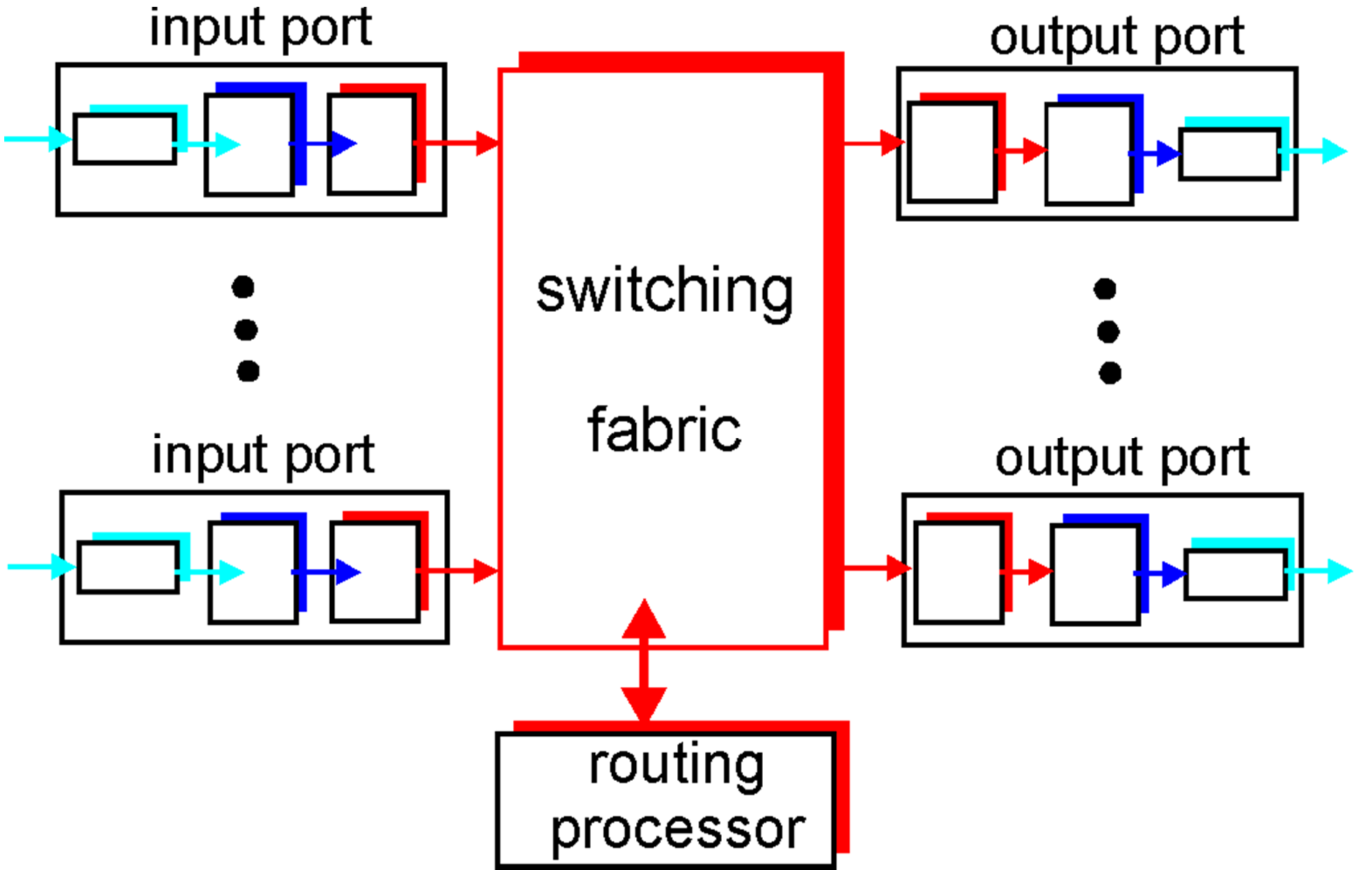
❷ Forwarding



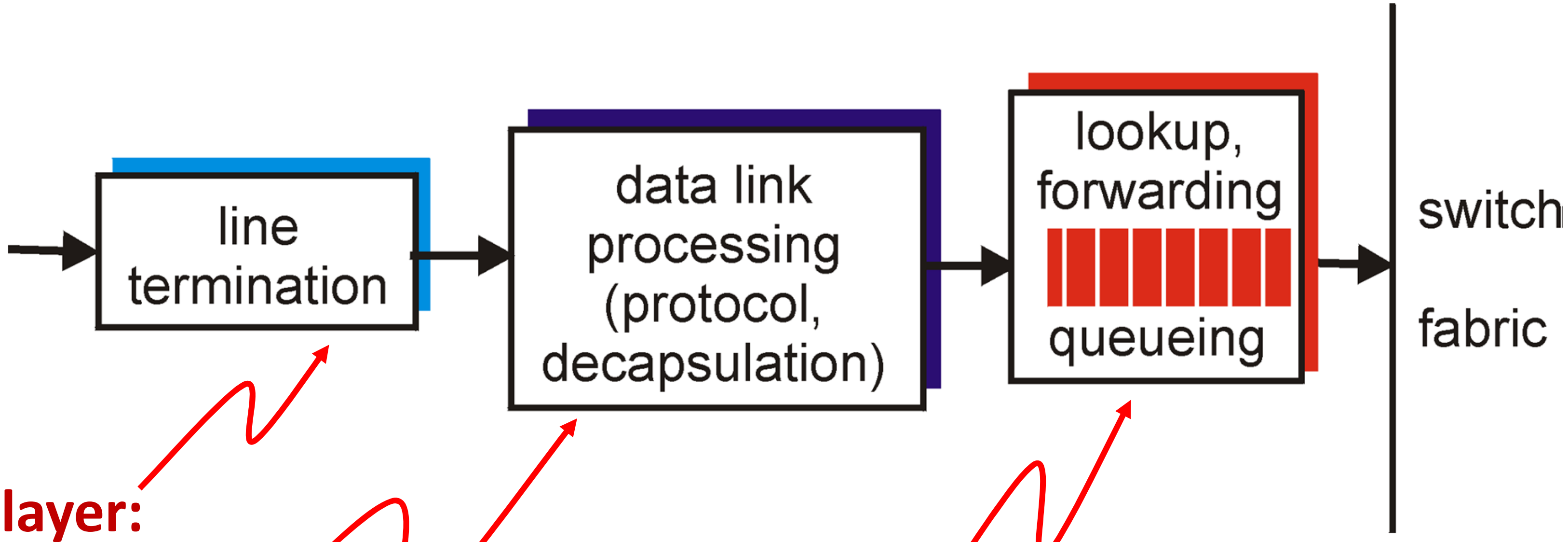
Forwarding table

Destination address	Output interface
A	1
B	2
⋮	⋮
H	3

Router Architecture Overview



Input Port



Physical layer:
bit-level reception

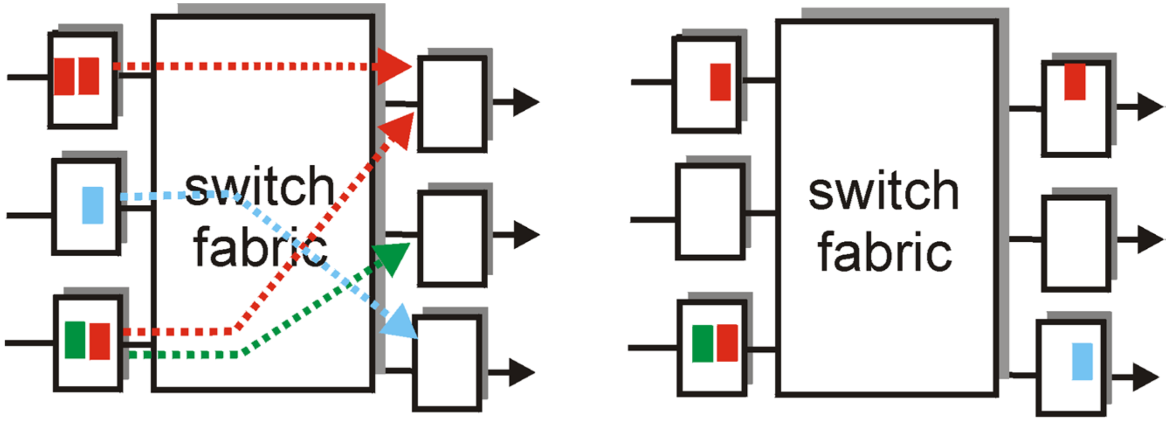
Data link layer:
e.g., Ethernet

Decentralized switching*:*

- Given destination, lookup output port using routing table in input port memory
- Goal: complete input port processing at ‘line speed’

Input Port Queuing

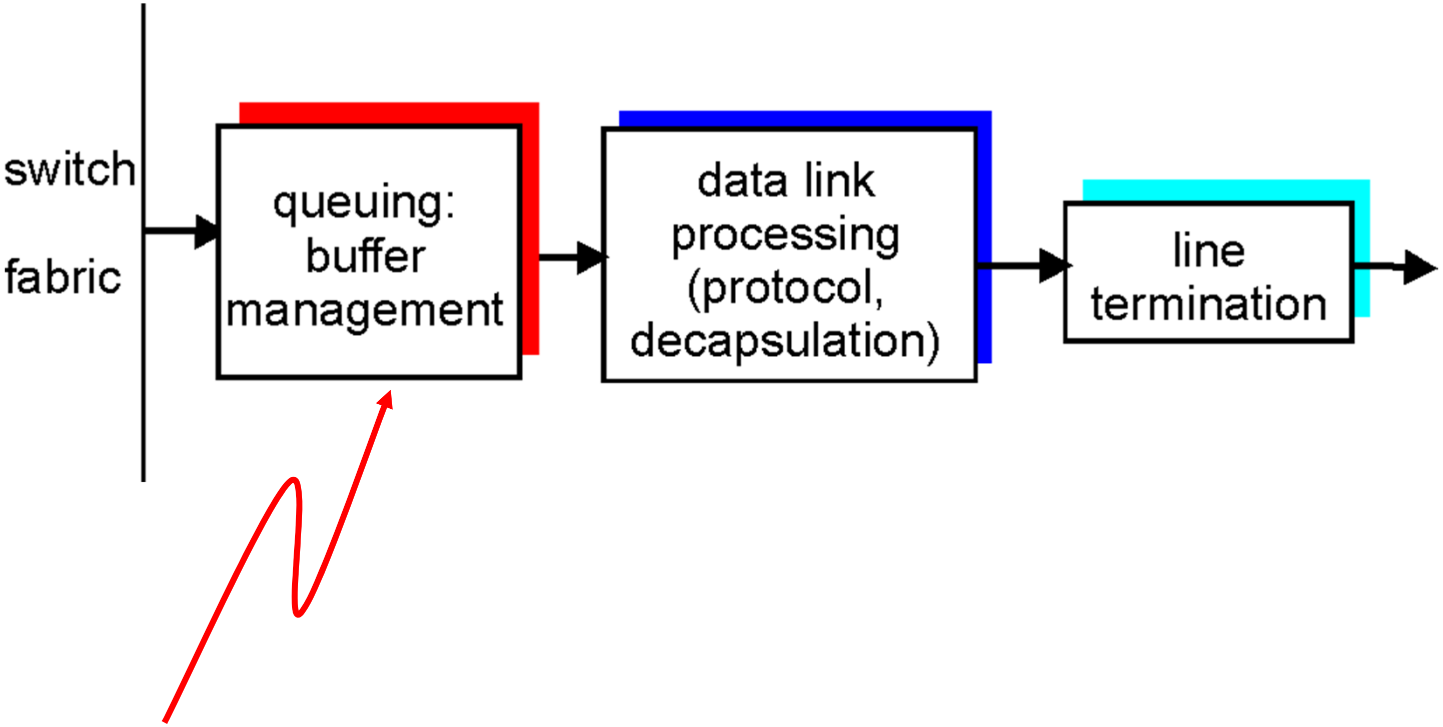
- Fabric slower that sum of input ports → **queuing**
- **Delay and loss** due to input buffer overflow
- **Head-of-the-Line (HOL) blocking**: Datagram at front of queue prevents others in queue from proceeding



output port contention at time t - only one red packet can be transferred

green packet experiences HOL blocking

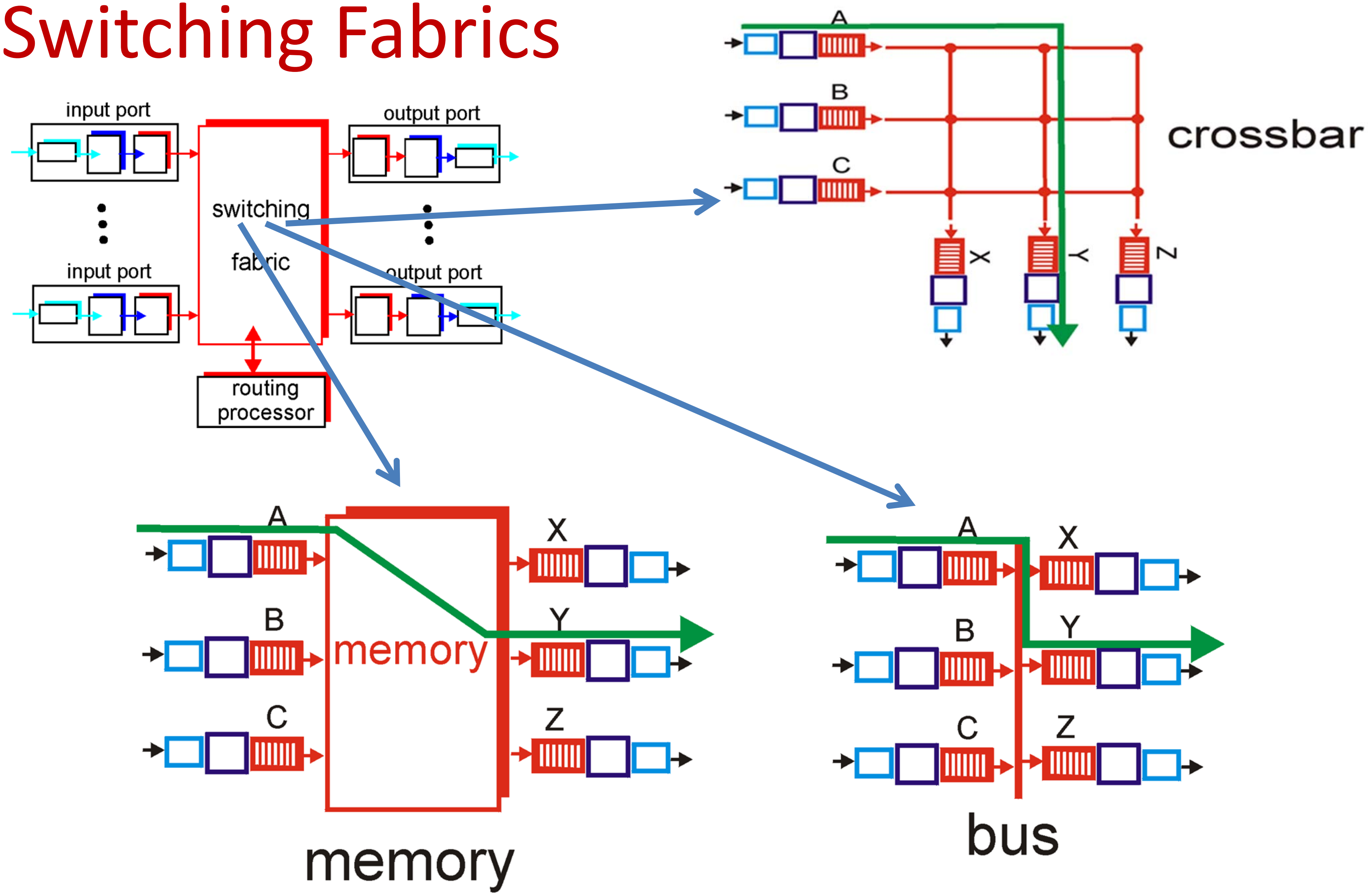
Output Port



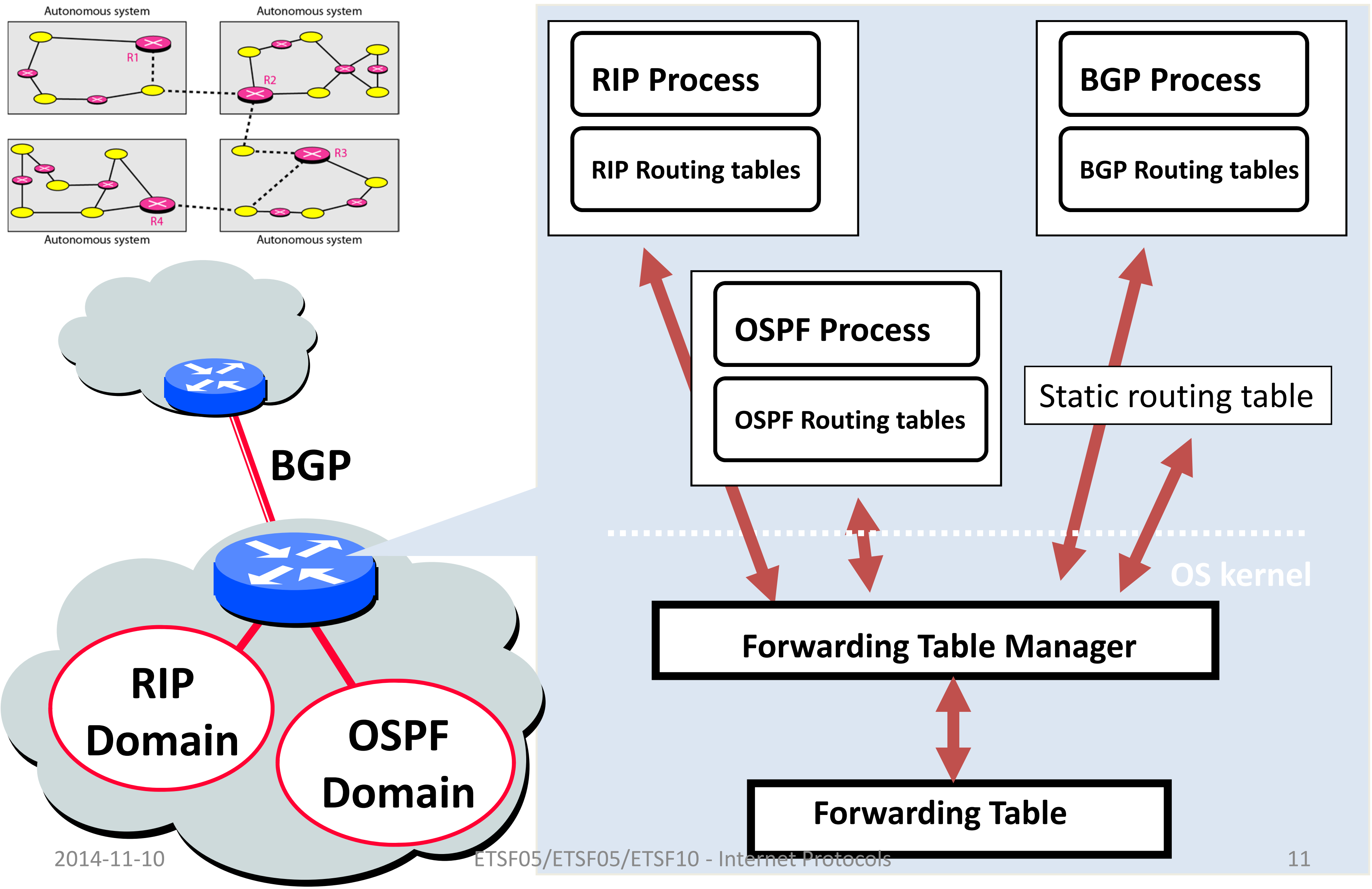
Priority Scheduling:

- Scheduling discipline may choose among queued datagrams for transmission

Switching Fabrics



Routing Tables and Forwarding Table



RIP Process

RIP Routing tables

BGP Process

BGP Routing tables

OSPF Process

OSPF Routing tables

Static routing table

OS kernel

Forwarding Table Manager

Forwarding Table

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

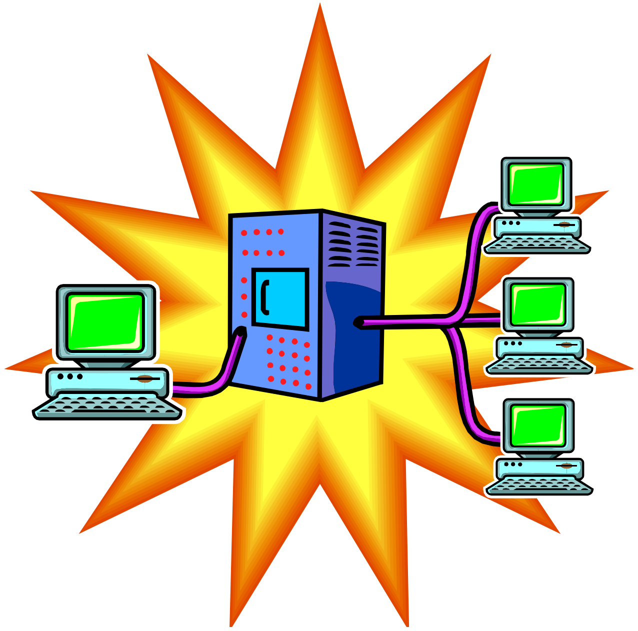
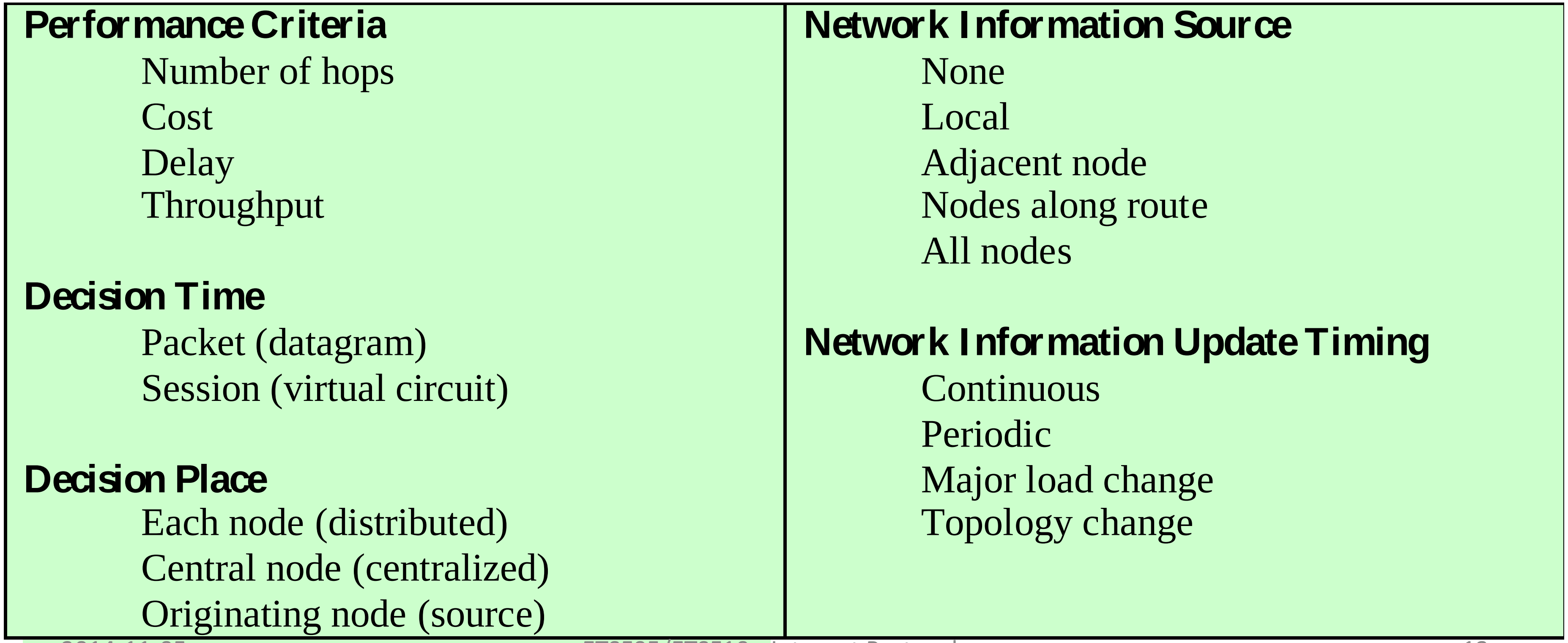


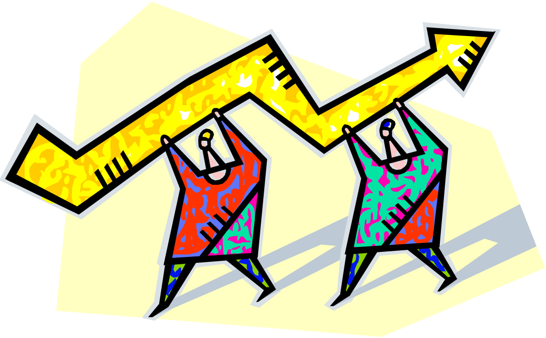
Table 19.1
Elements of Routing Techniques for Packet-Switching Networks



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

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”



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

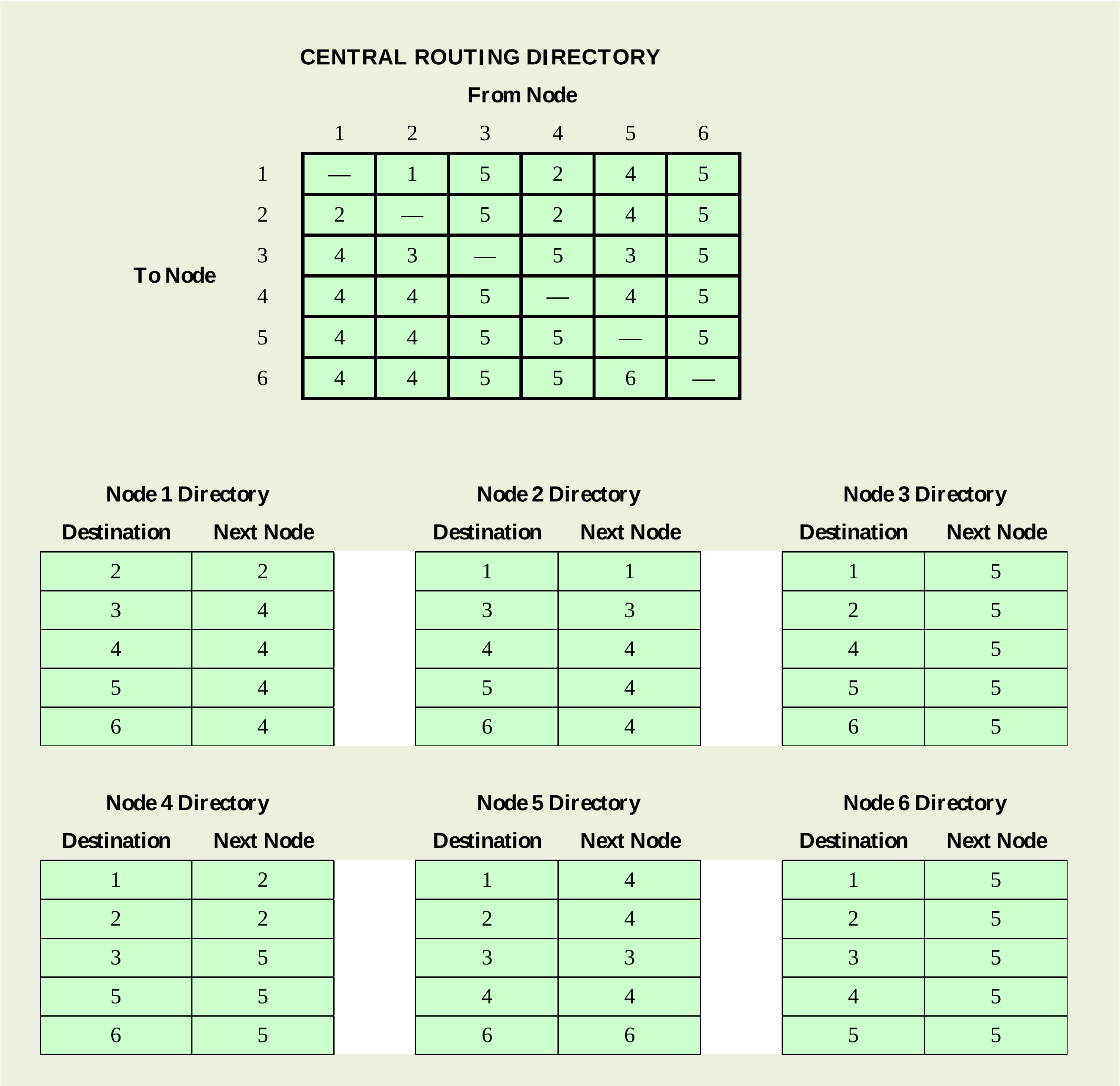
- Routing decisions usually based on knowledge of network, traffic load, and link cost
  - Distributed routing
    - Using local knowledge, information from adjacent nodes, information from all nodes on a potential route
  - Central routing
    - Collect information from all nodes

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

To Node \ From 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)

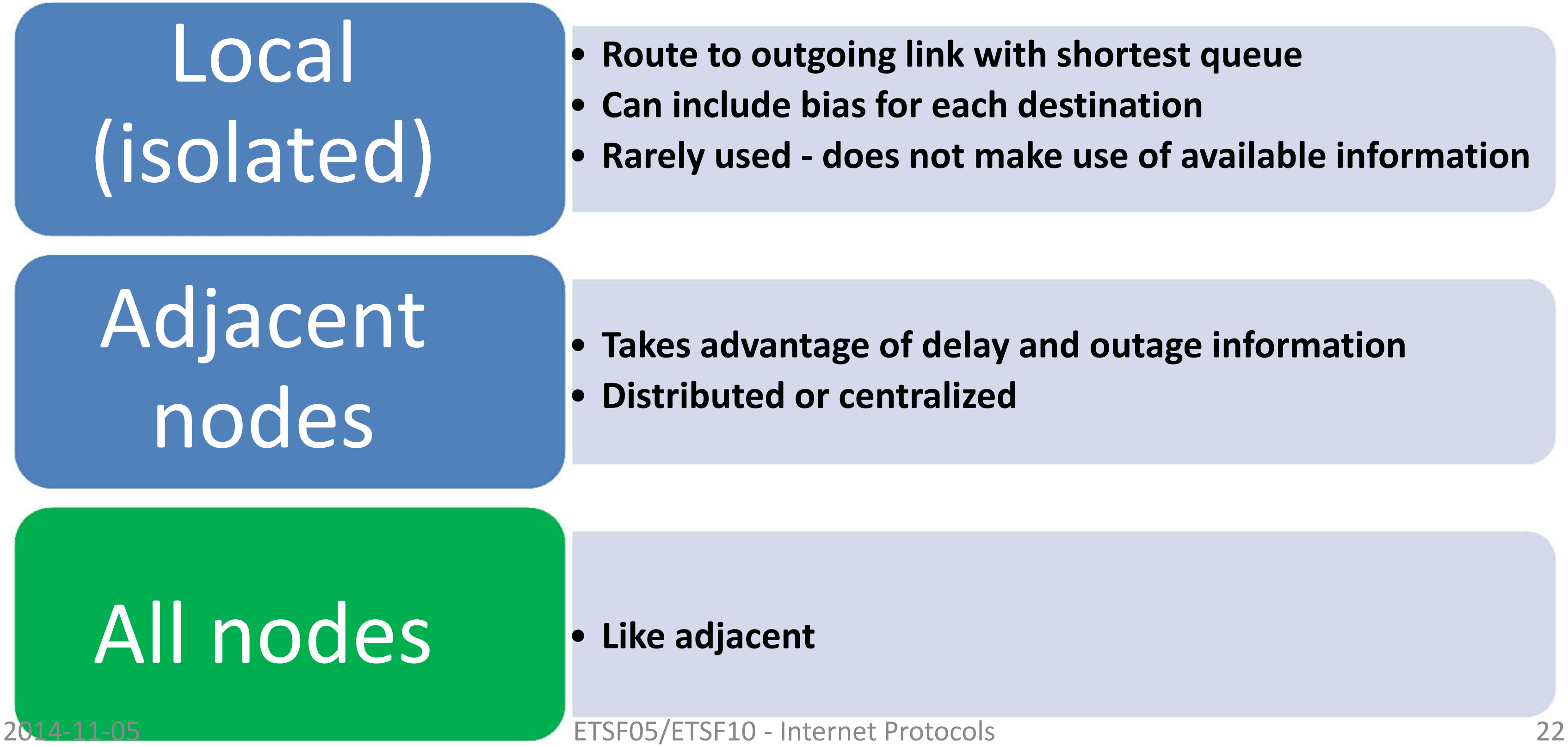
Routing Strategies - Adaptive Routing

- 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

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

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

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**

Autonomous Systems (AS)

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

Interior Router Protocol (IRP)
Interior Gateway Protocol (IGP)

- A shared routing protocol which passes routing information between routers within an AS
- Custom tailored to specific applications and requirements

Examples

- *Routing Information Protocol (RIP)*
- *Open Shortest Path First (OSPF)*

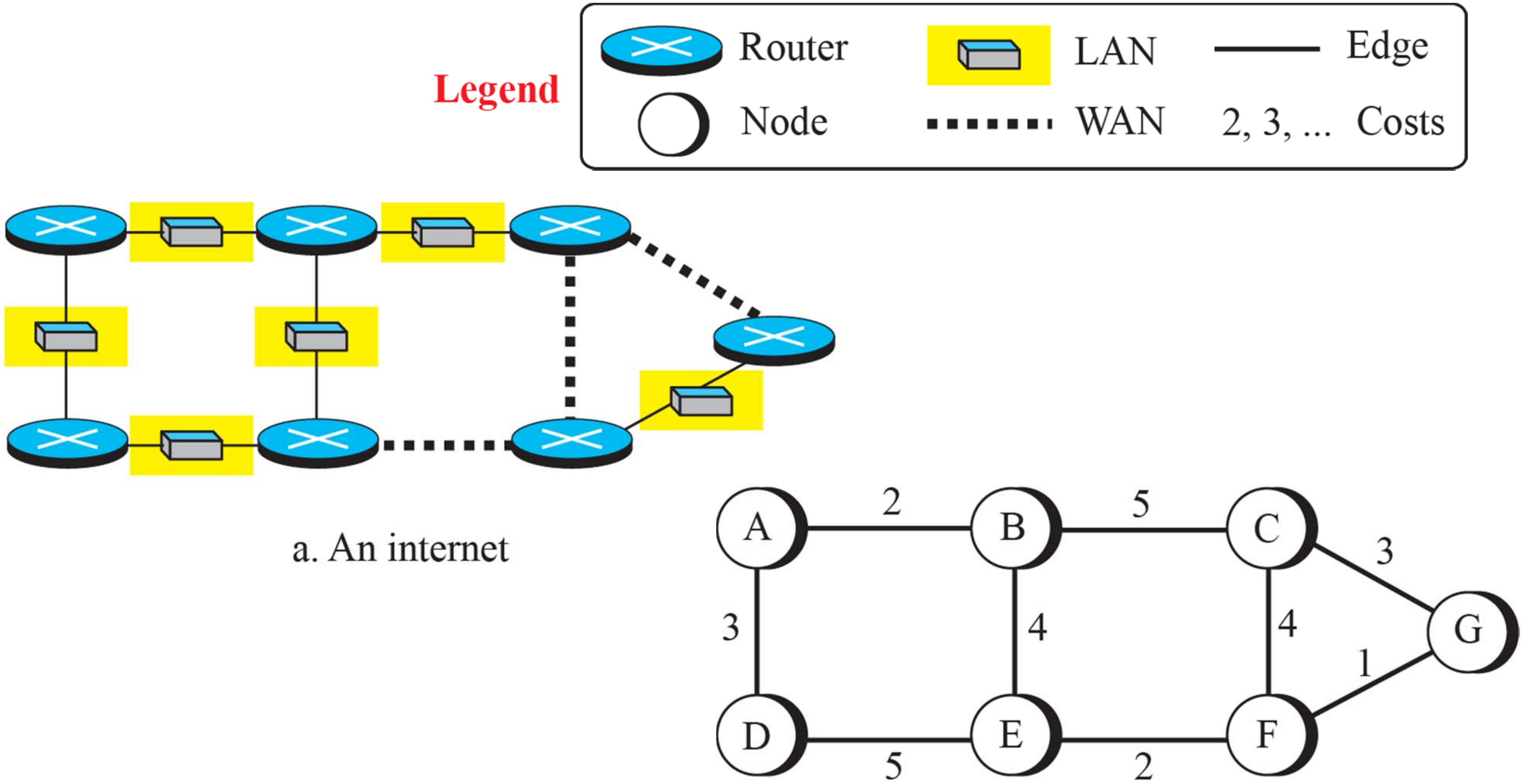
Exterior Router Protocol (ERP) Exterior Gateway Protocol (EGP)

- 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)*

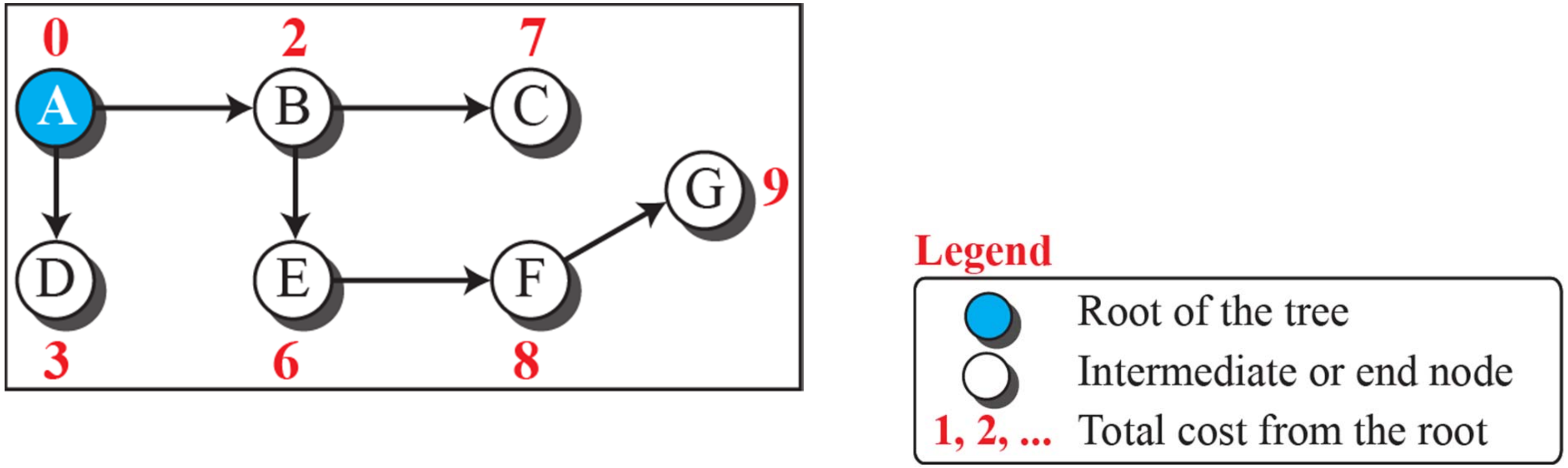
Graphical representation of a net



a. An internet

b. The weighted graph

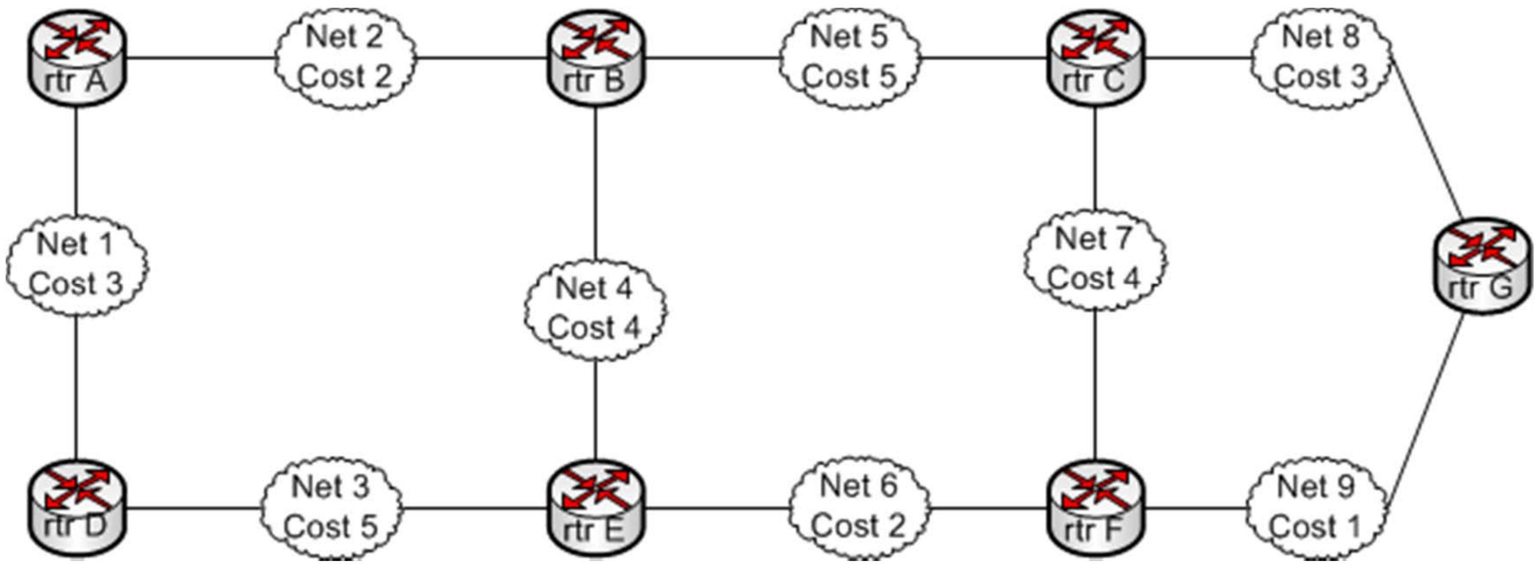
What is an end node?



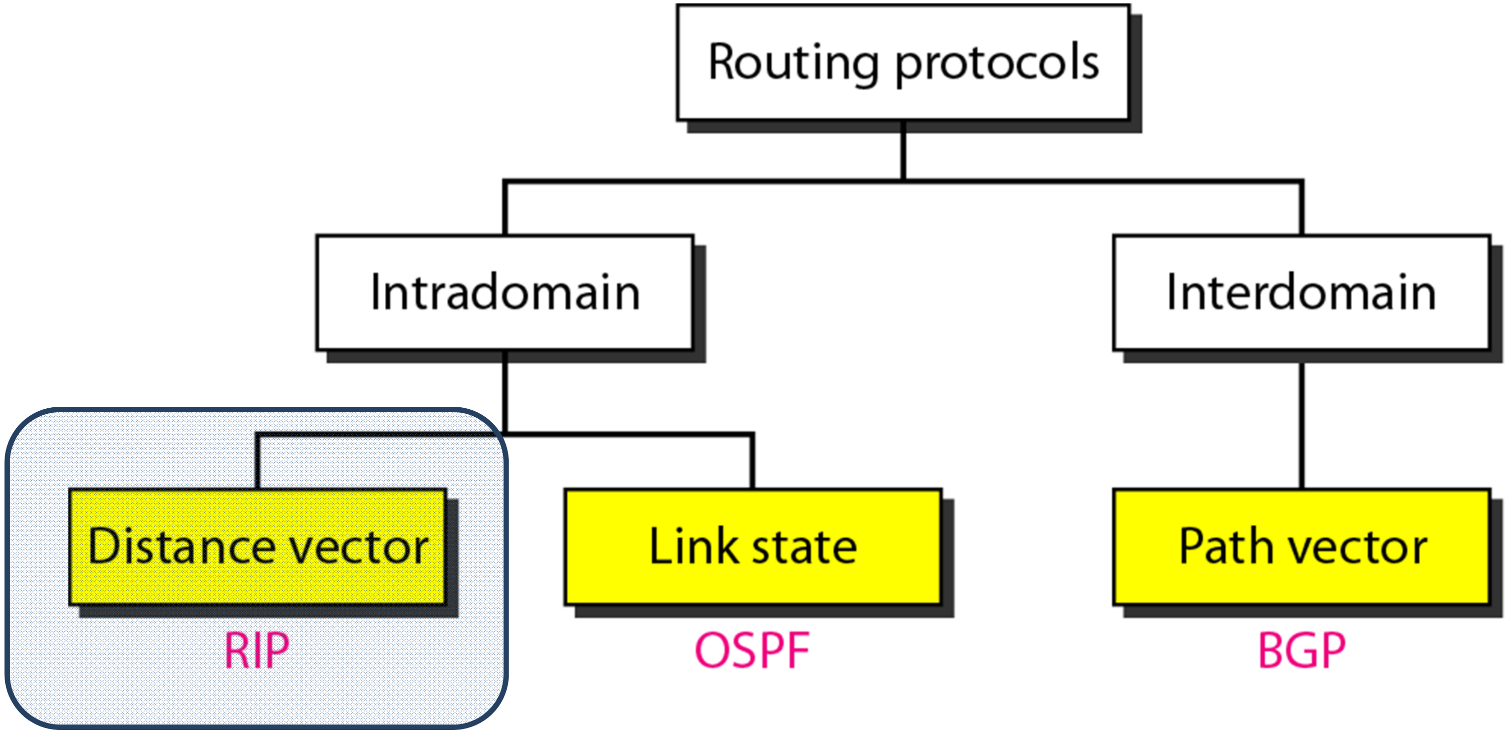
Problem: The LANs are our destinations/end nodes, not the routers

A more realistic representation

- Solution: Nets and routers are all nodes in the tree.
- Routers hold tables how to reach nets and what is the *next hop* for to get there



Routing Algorithms and Protocols



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

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

RIP (Routing Information Protocol)

- Included in BSD-UNIX Distribution in 1982
- Distance metric:
  - **# of hops** (max 15) to destination network
- Distance vectors:
  - exchanged among neighbours every 30” via Response Message (advertisement)
- Implementation:
  - Application layer protocol, uses UDP/IP

A RIP Forwarding/Routing Table

Destination=net	Cost	Next hop=router
123	3	A
32	5	D
16	3	A
7	2	-

RIP update message

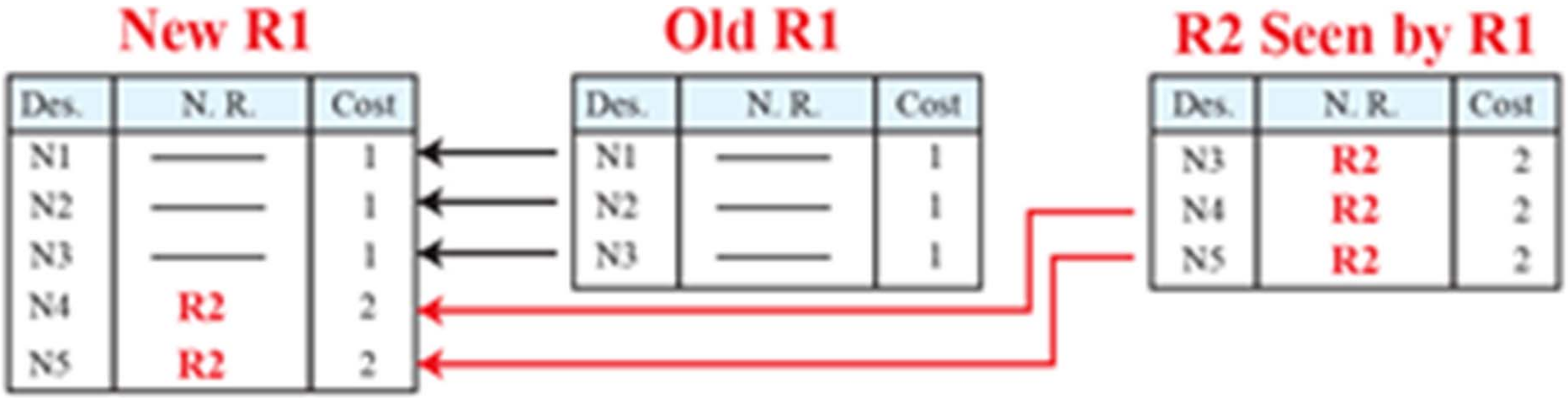
- Contains the whole forwarding table
- Add 1 to cost in received message
- Change next hop to sending router
- Apply RIP updating algorithm

- IMPORTANT! Received update msgs identify neighbours!

RIP Updating Algorithm (Bellman-Ford)

```
if (advertised destination not in table)
   {
   add new entry // rule #1
   }
else if (adv. next hop = next hop in table)
   {
   update cost // rule #2
   }
else if (adv. cost < cost in table)
   {
   replace old entry // rule #3
   }
```

RIP Example



New R1

Des.	N. R.	Cost
N1	——	1
N2	——	1
N3	——	1
N4	R2	2
N5	R2	2

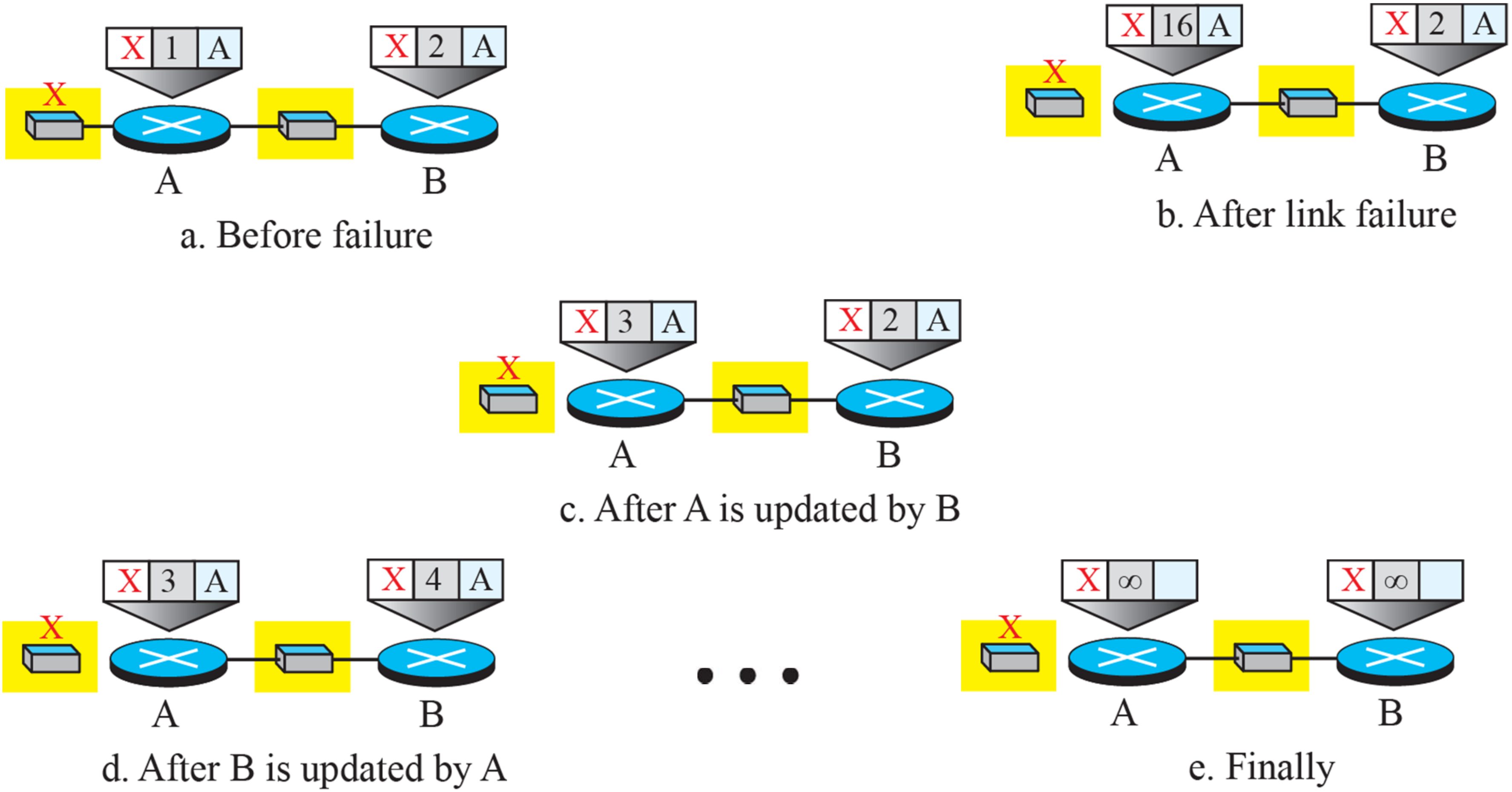
Old R1

Des.	N. R.	Cost
N1	——	1
N2	——	1
N3	——	1

R2 Seen by R1

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

Two node instability/Count to inifinity



a. Before failure

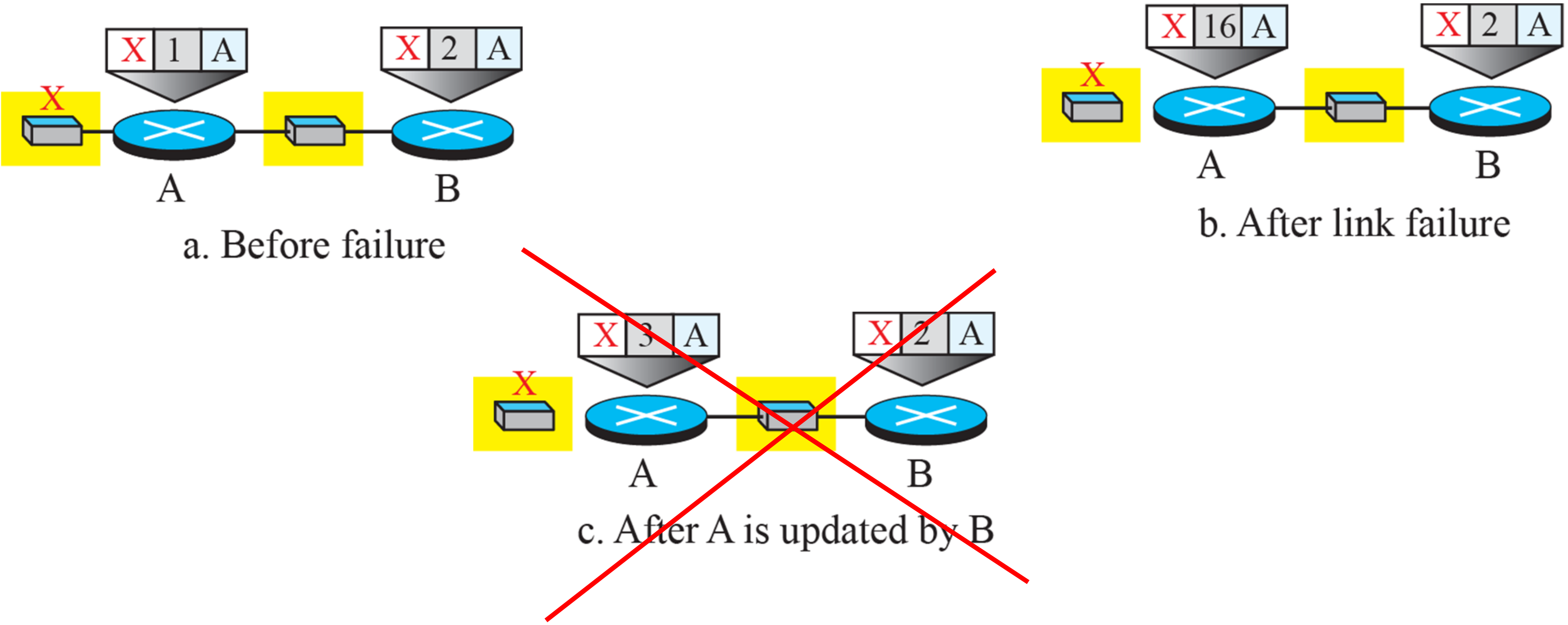
b. After link failure

c. After A is updated by B

d. After B is updated by A

e. Finally

Split Horizon breaks Count to inifinity



a. Before failure

b. After link failure

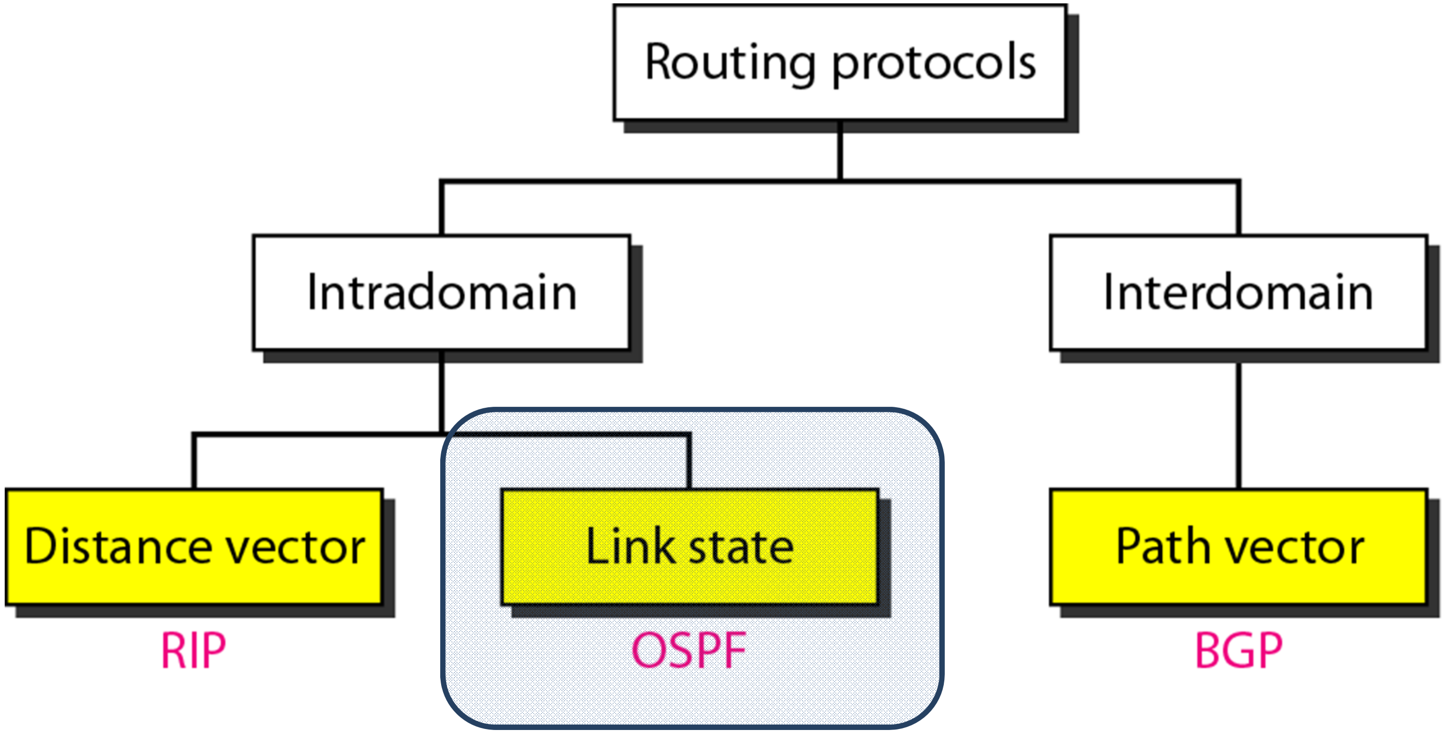
c. After A is updated by B

I have a route to X, but I got it from A so I won’t tell A about it!

RIP: Link Failure and Recovery

- If no advertisement heard after 180”
  - Neighbour/link declared dead
  - Routes via neighbour invalidated (infinite distance = 16 hops)
  - New advertisements sent to neighbours (triggering a chain reaction if tables changed)
  - “Poison reverse” used to prevent count to infinity loops
  - “Good news travel fast, bad news travel slow”

Routing Algorithms and Protocols



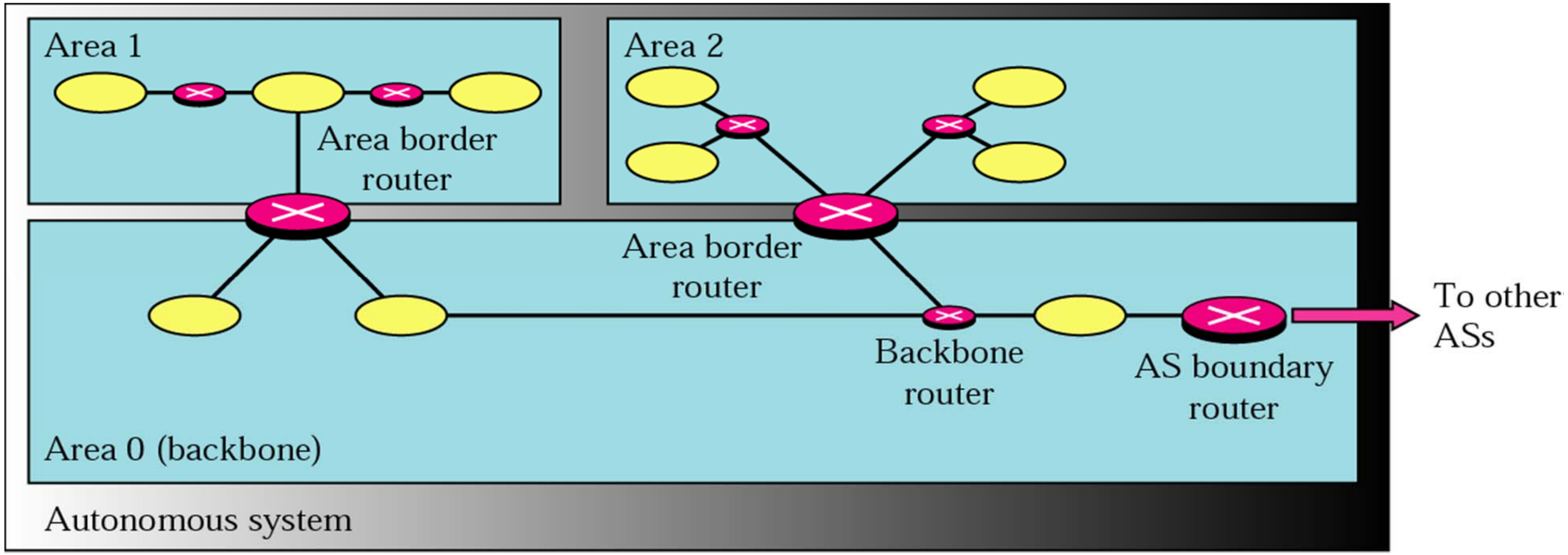
Open Shortest Path First (OSPF) Protocol

- 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

OSPF (Open Shortest Path First)

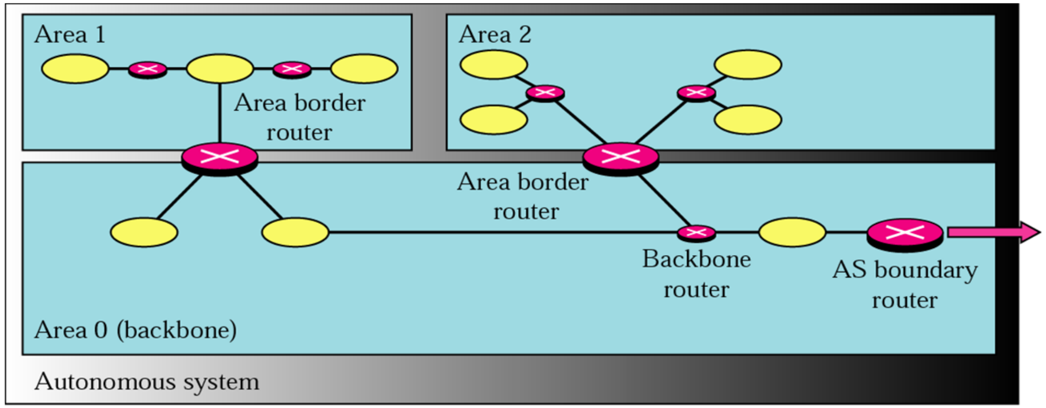
- Divides domain into areas
  - Limits flooding for efficiency
  - One ”backbone” area connects all
- Distance metric:
  - Cost to destination network

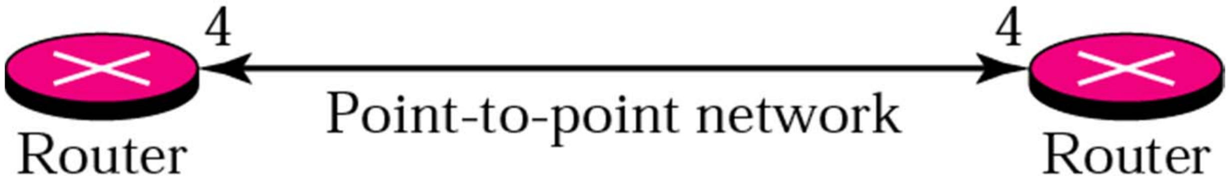
Areas, Router and Link Types



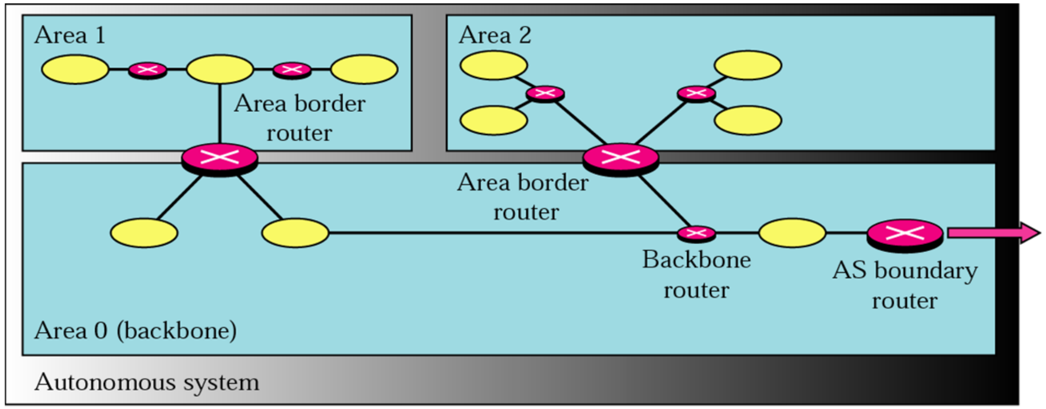
Point-to-Point Link

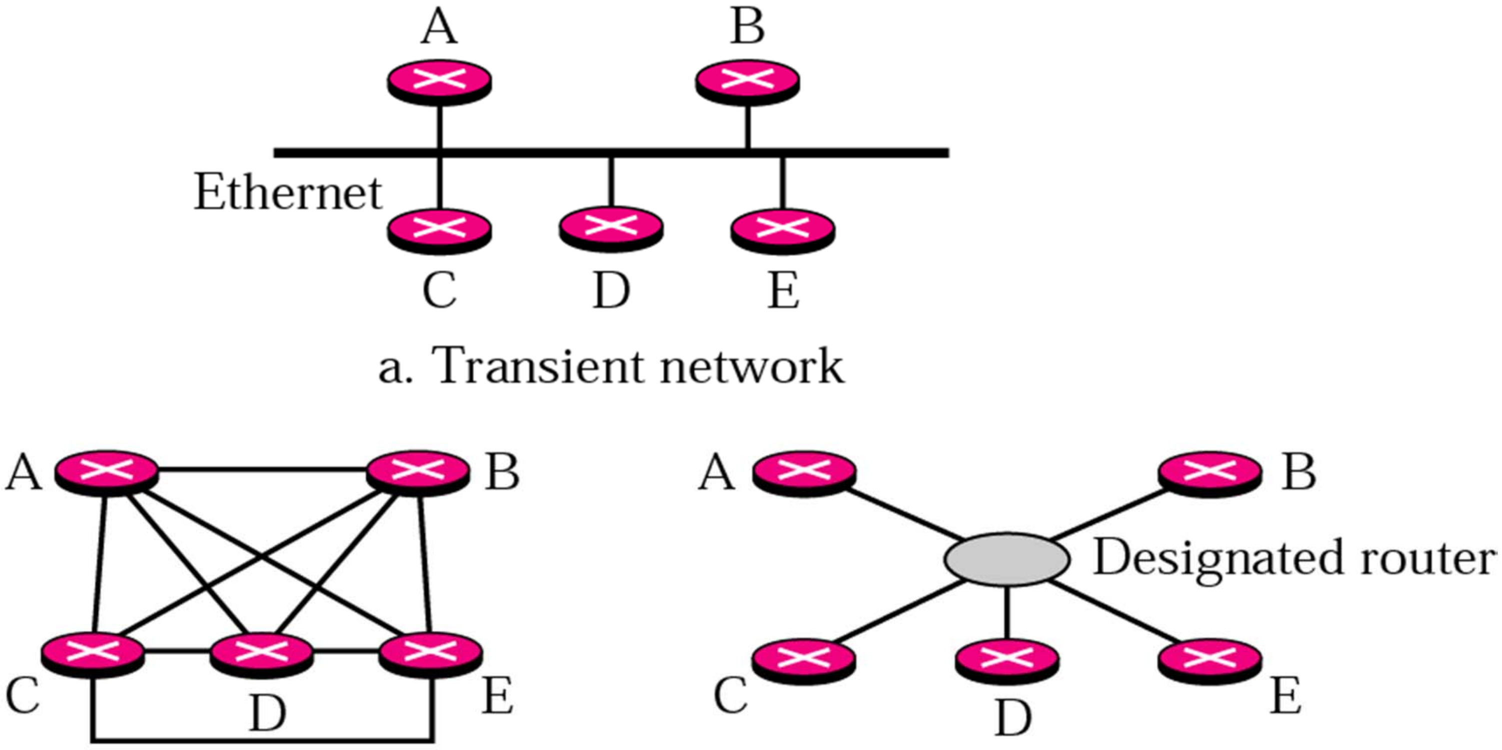
- Connects two routers
- No need for addresses





Transient Link



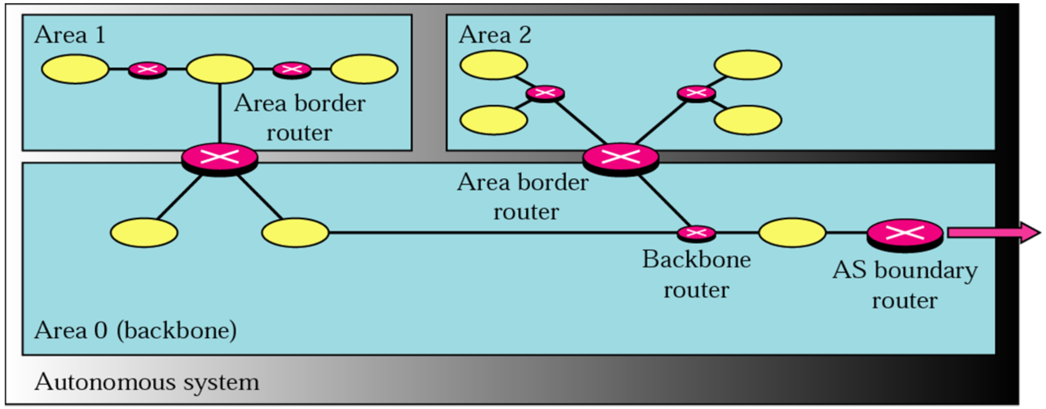


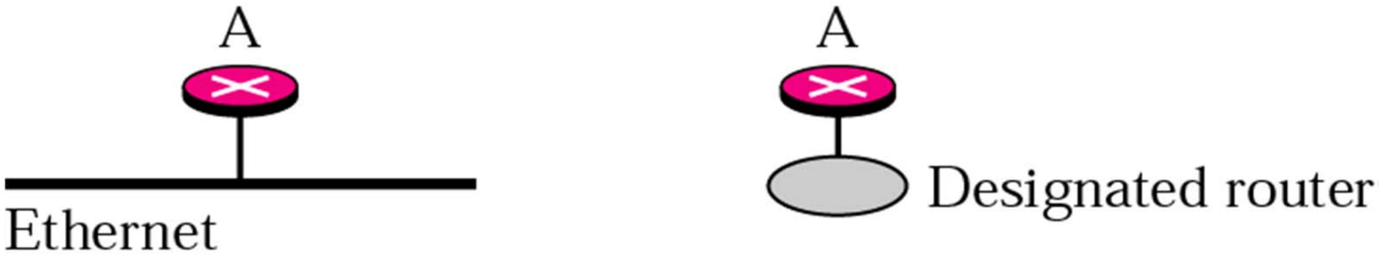
a. Transient network

b. Unrealistic representation

c. Realistic representation

Stub Link

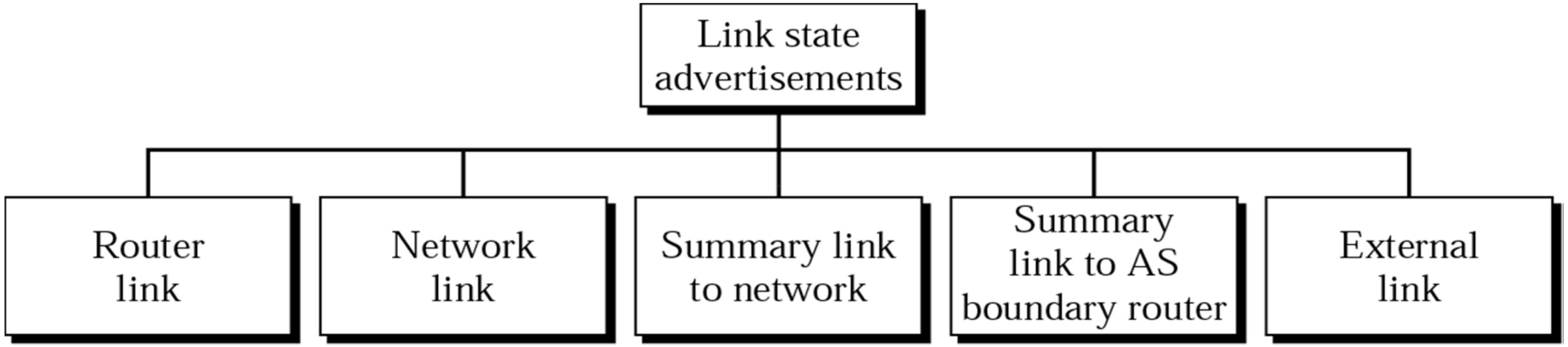




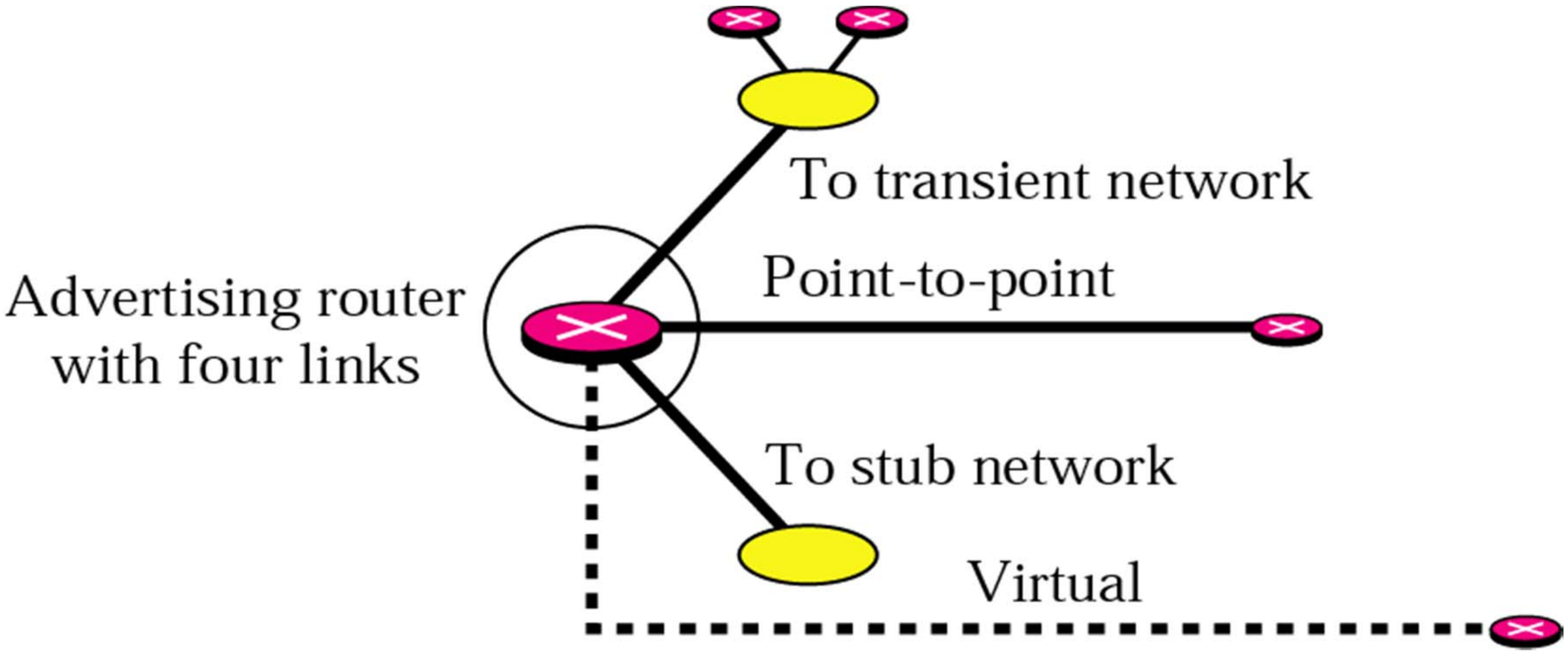
a. Stub network b. Representation

Link State Advertisements

- What to advertise?
  - Different entities as nodes
  - Different link types as connections
  - Different types of cost

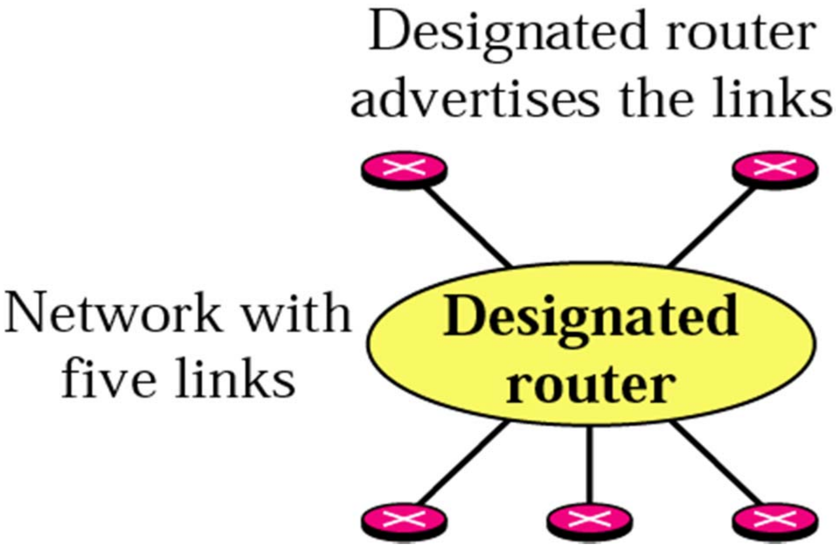


Router Link Advertisement



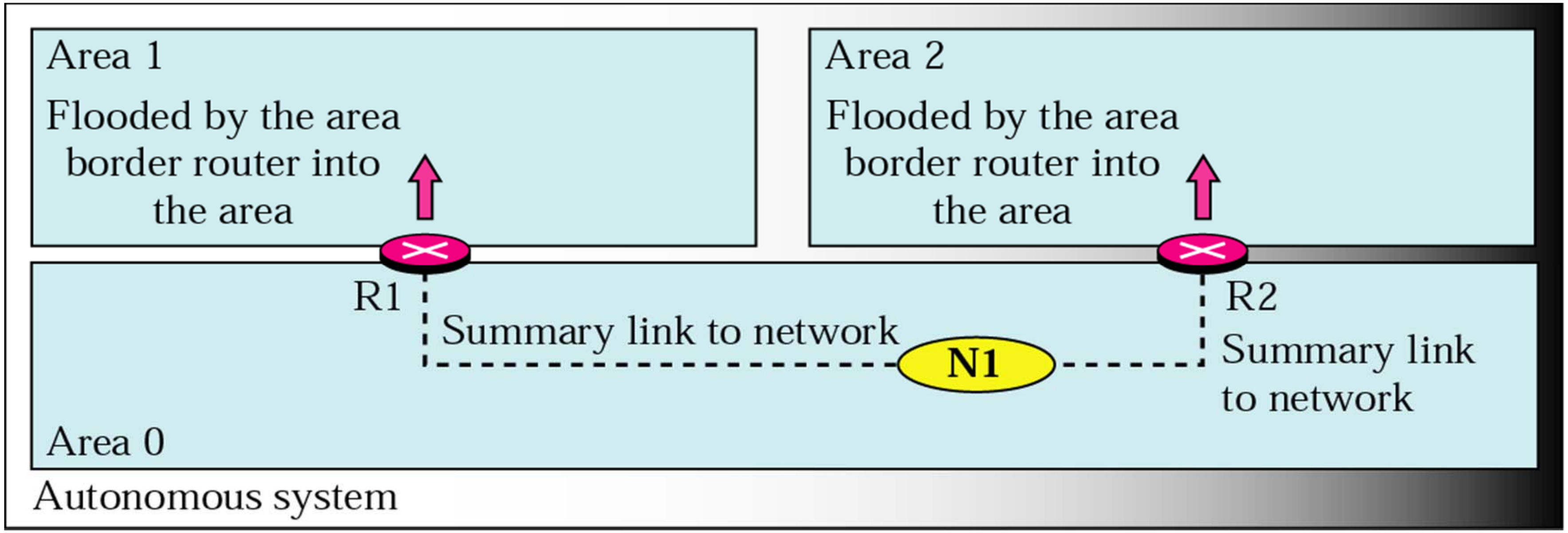
Network Link Advertisement

- Network is a passive entity
  - It cannot advertise itself



Summary Link to Network

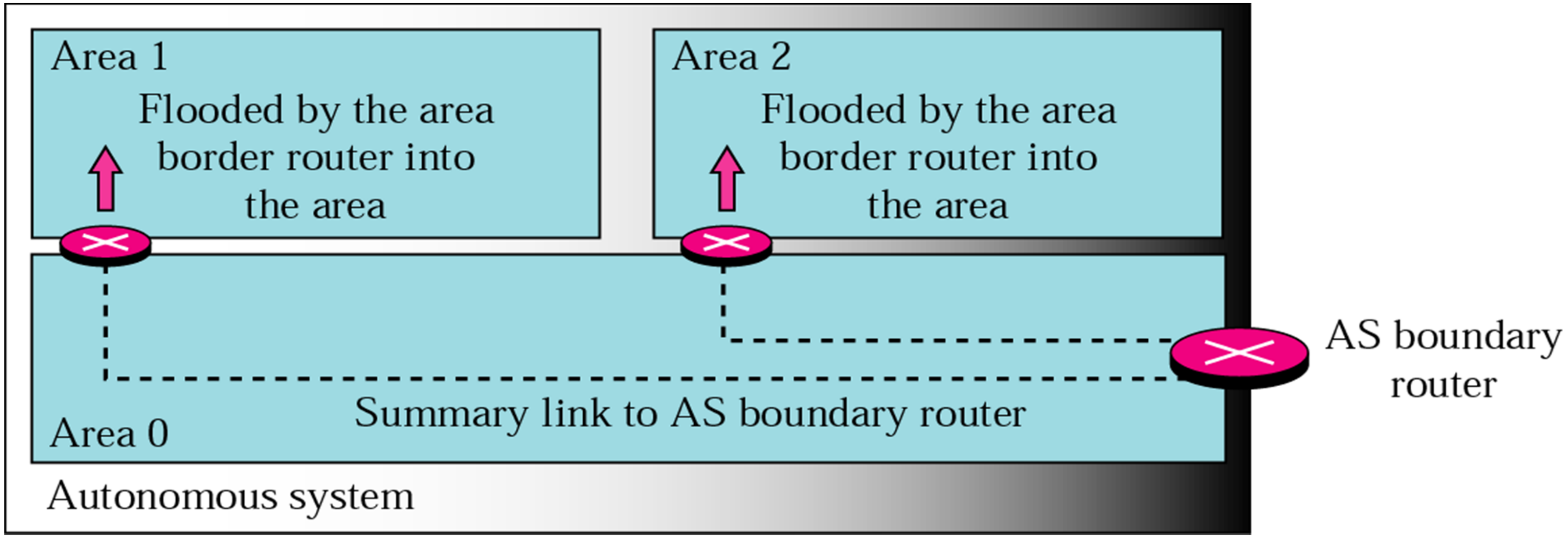
- Done by area border routers
  - Goes through the backbone


Area 1
Flooded by the area border router into the area
Area 2
Flooded by the area border router into the area
R1
R2
Summary link to network
N1
Summary link to network
Area 0
Autonomous system

Summary Link to AS Boundary Router

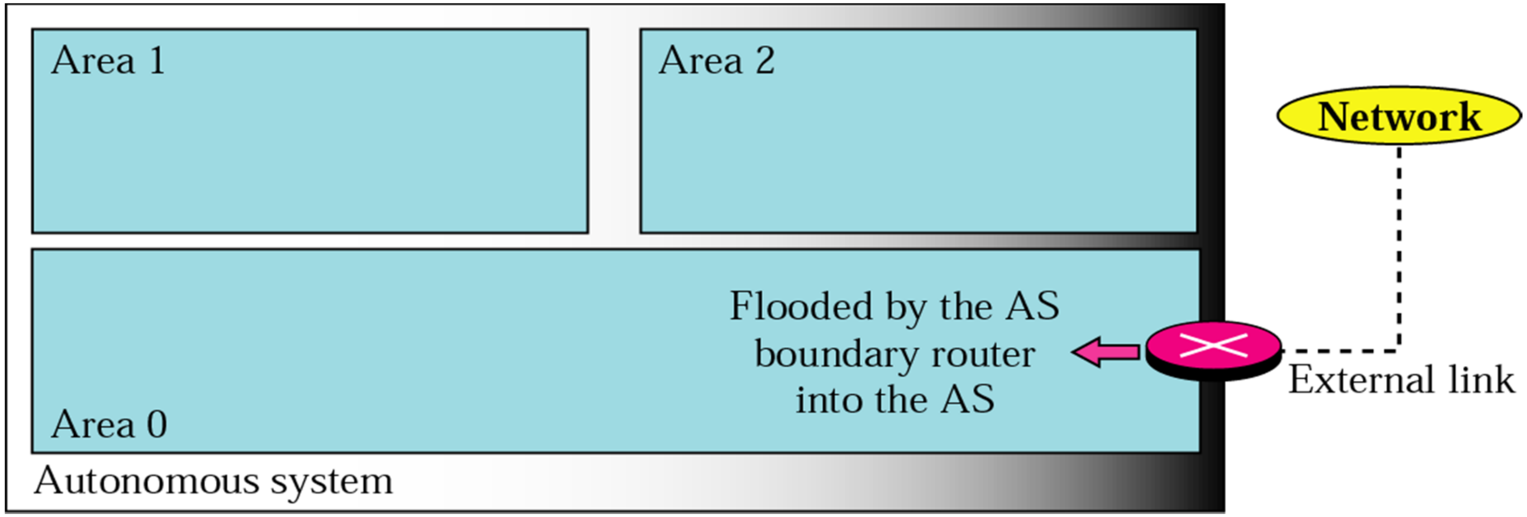
- Links to other domains

  ”autonomous systems”


Area 1
Flooded by the area border router into the area
Area 2
Flooded by the area border router into the area
AS boundary router
Summary link to AS boundary router
Area 0
Autonomous system

External Link Advertisement

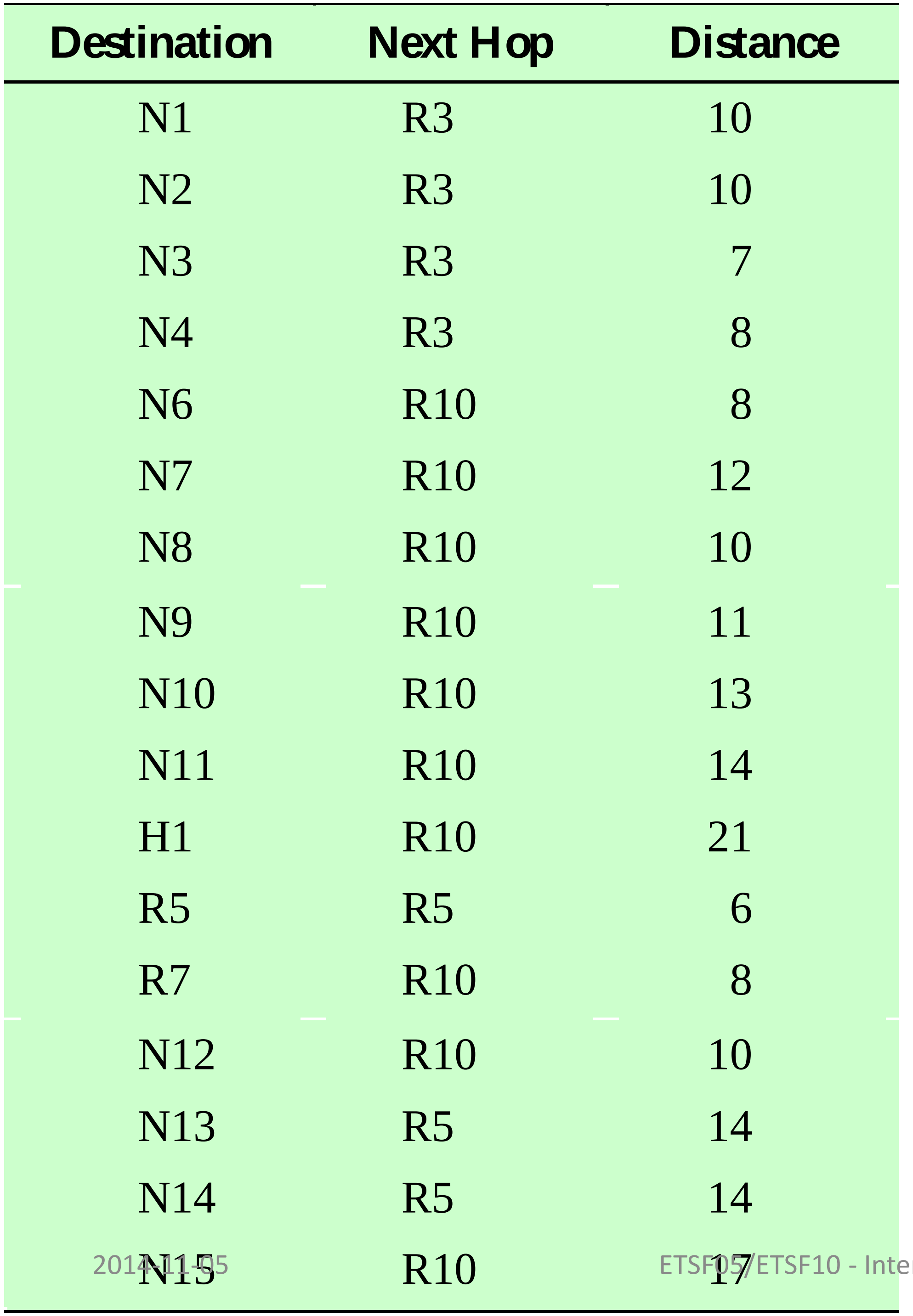
- Link to a single network outside the domain



Hello message

- Find neighbours
- Keep contact with neighbours: I am still alive!
- Sent out periodically (typically every 10th second)
- If no hellos received during holdtime (typically 30 seconds), neighbour declared dead.

- Compare RIP update messages



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

Table 19.3

Routing Table for R6

Dijkstra’s Algorithm

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