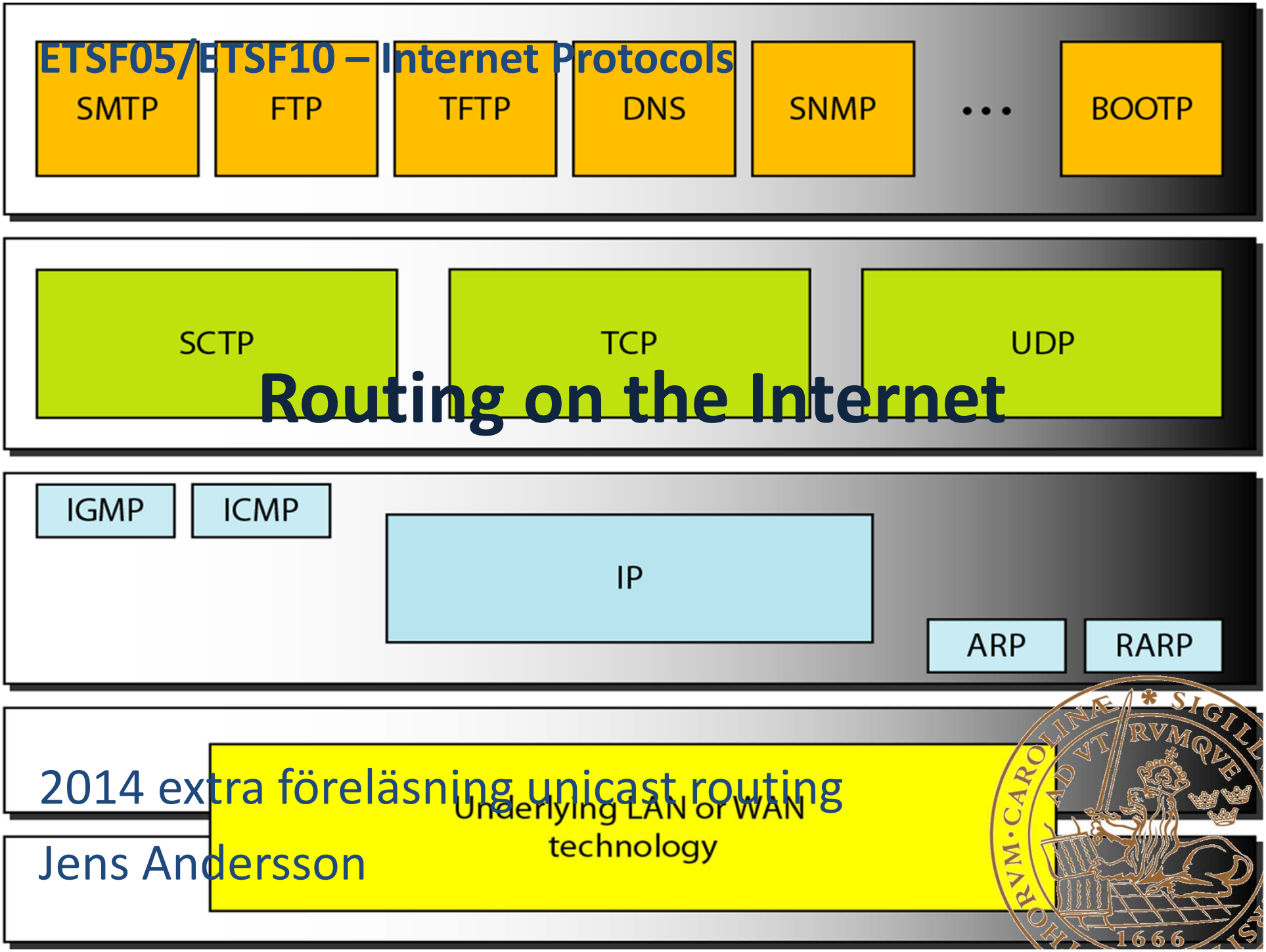
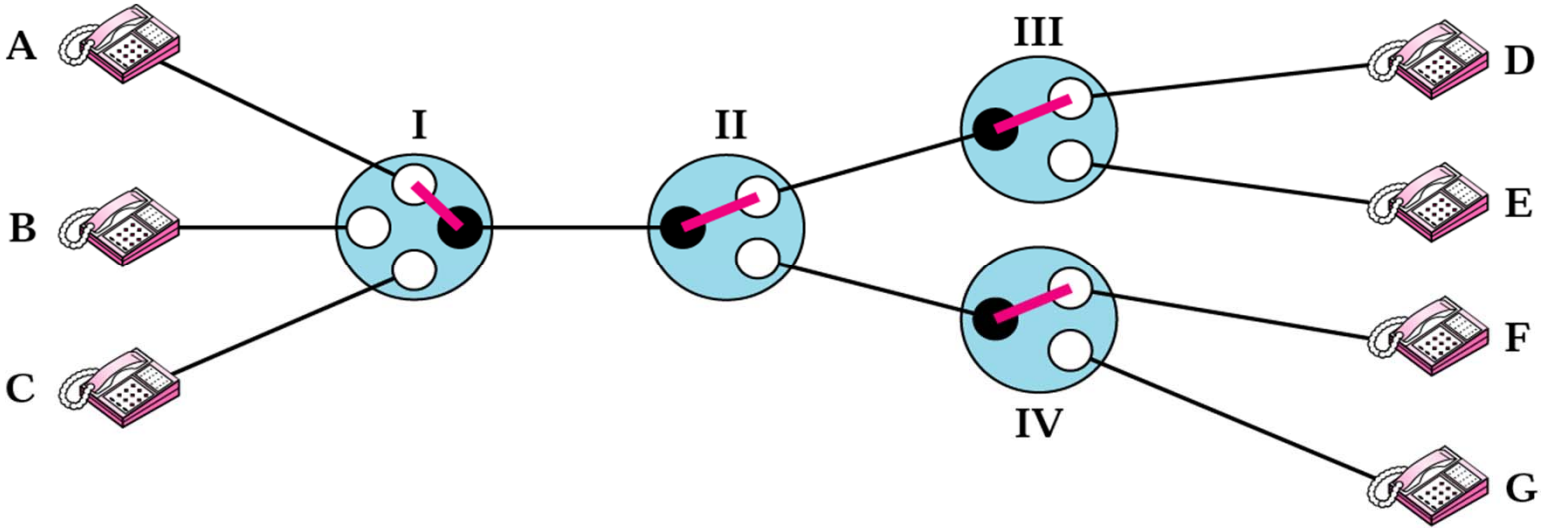


2014 extra föreläsning unicast routing

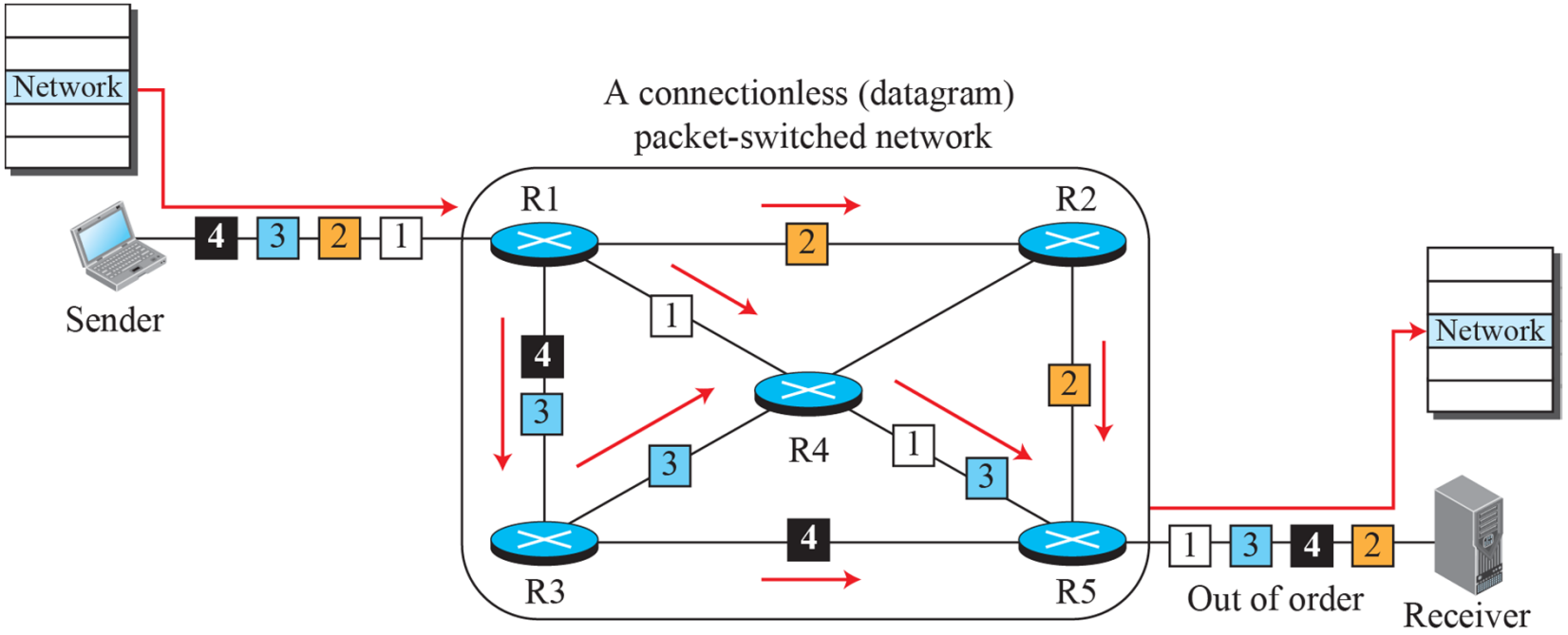
Jens Andersson

Circuit switched routing



Packet-switched Routing

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

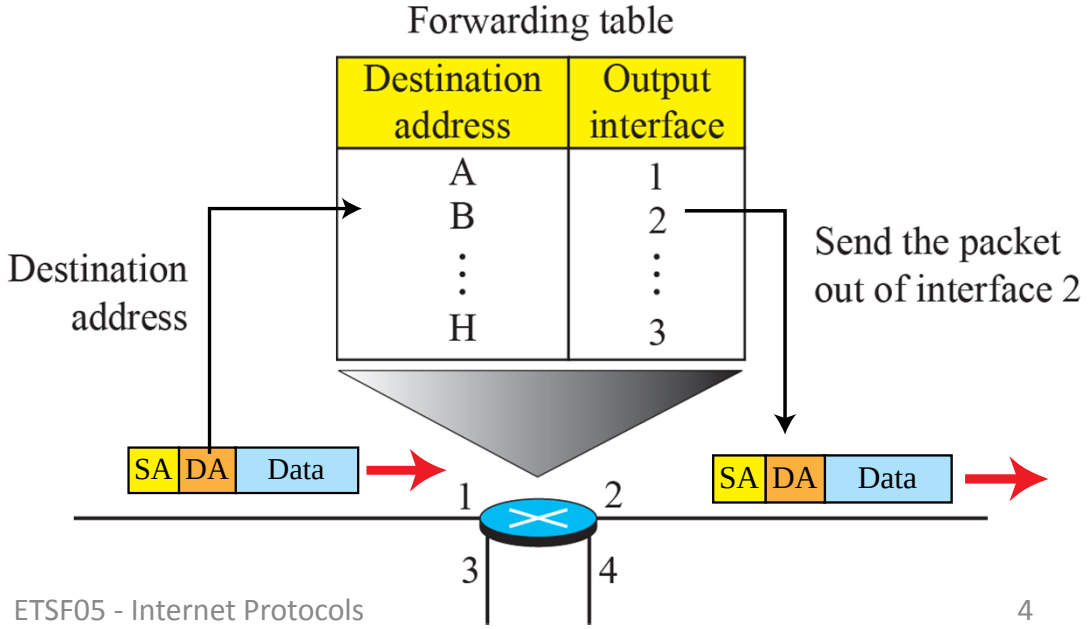


Router

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

Two functions:

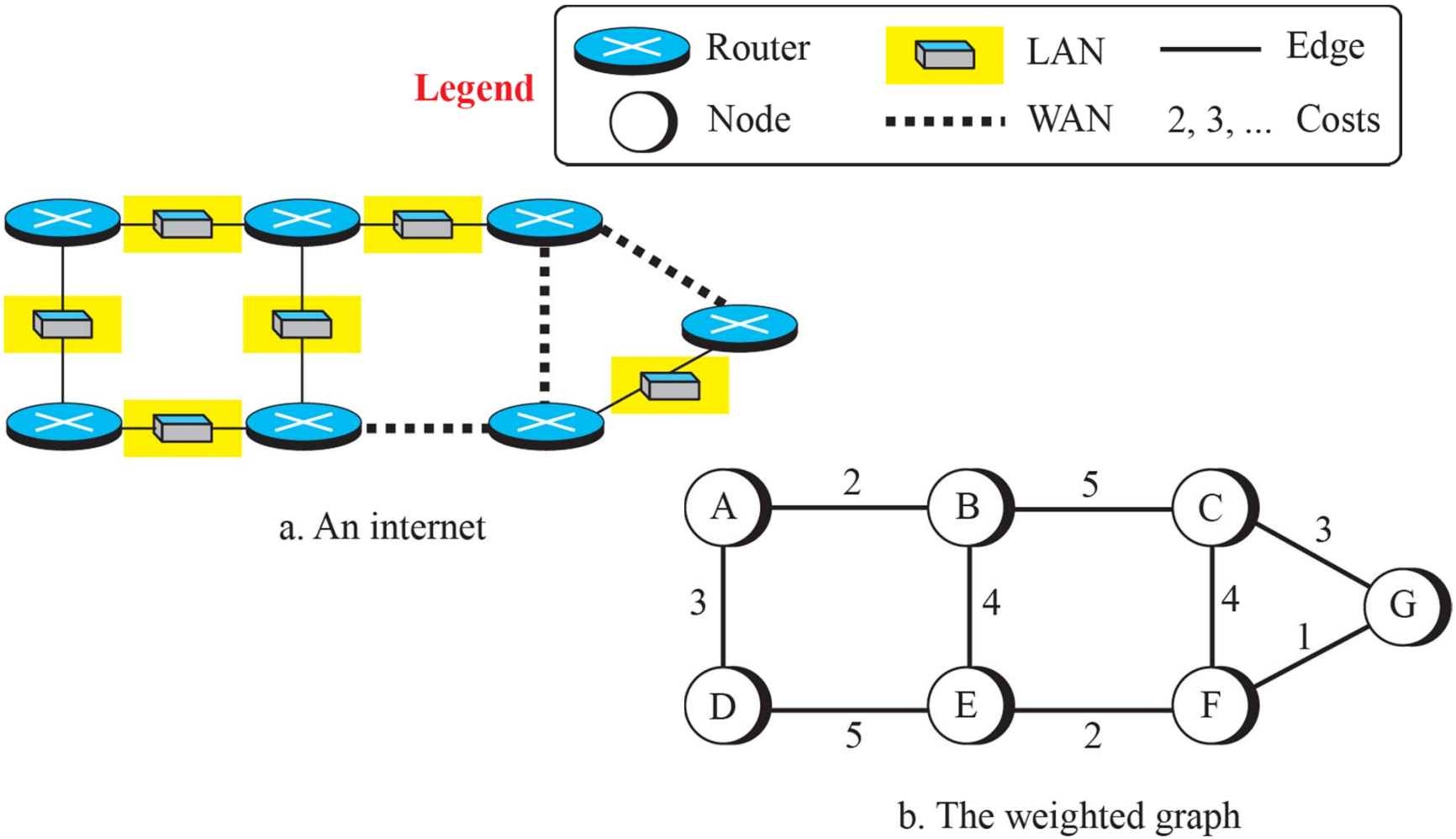
1. Routing
2. Forwarding



Common Cost Metrics

- Alternatives at the link level
  - Hop count
  - Inverse of the link bandwidth
  - Delay
  - Dynamically calculated
  - Administratively assigned
  - Combination
- Traffic monitored → metrics adjusted

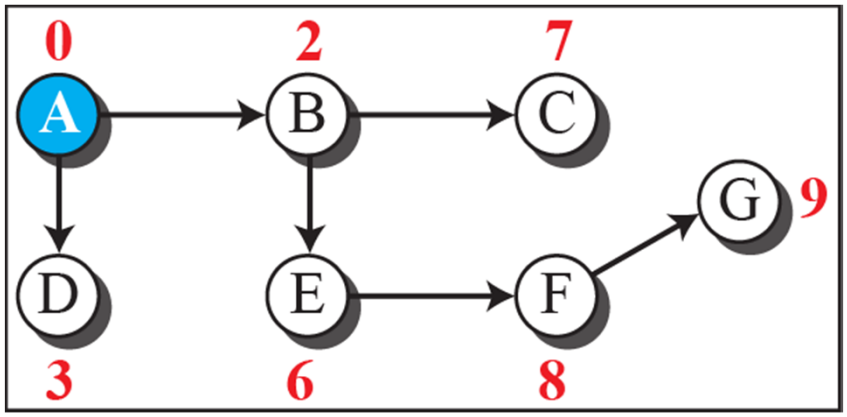
Graphical representation of a net

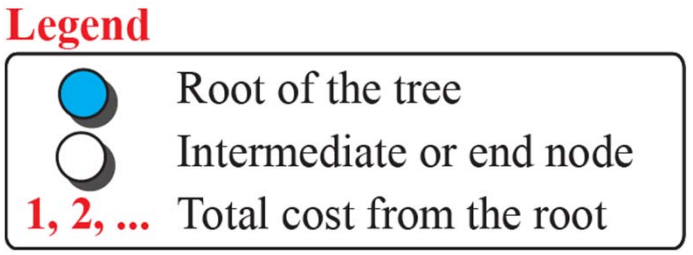


a. An internet

b. The weighted graph

What is an end node?

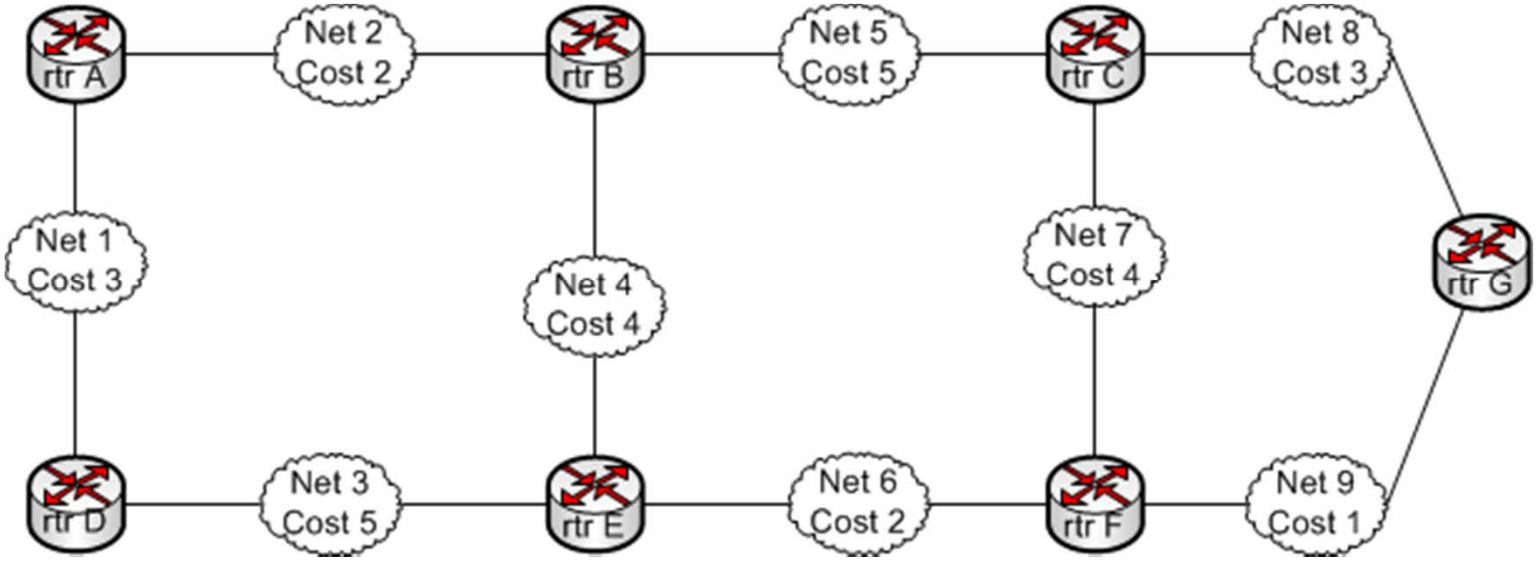




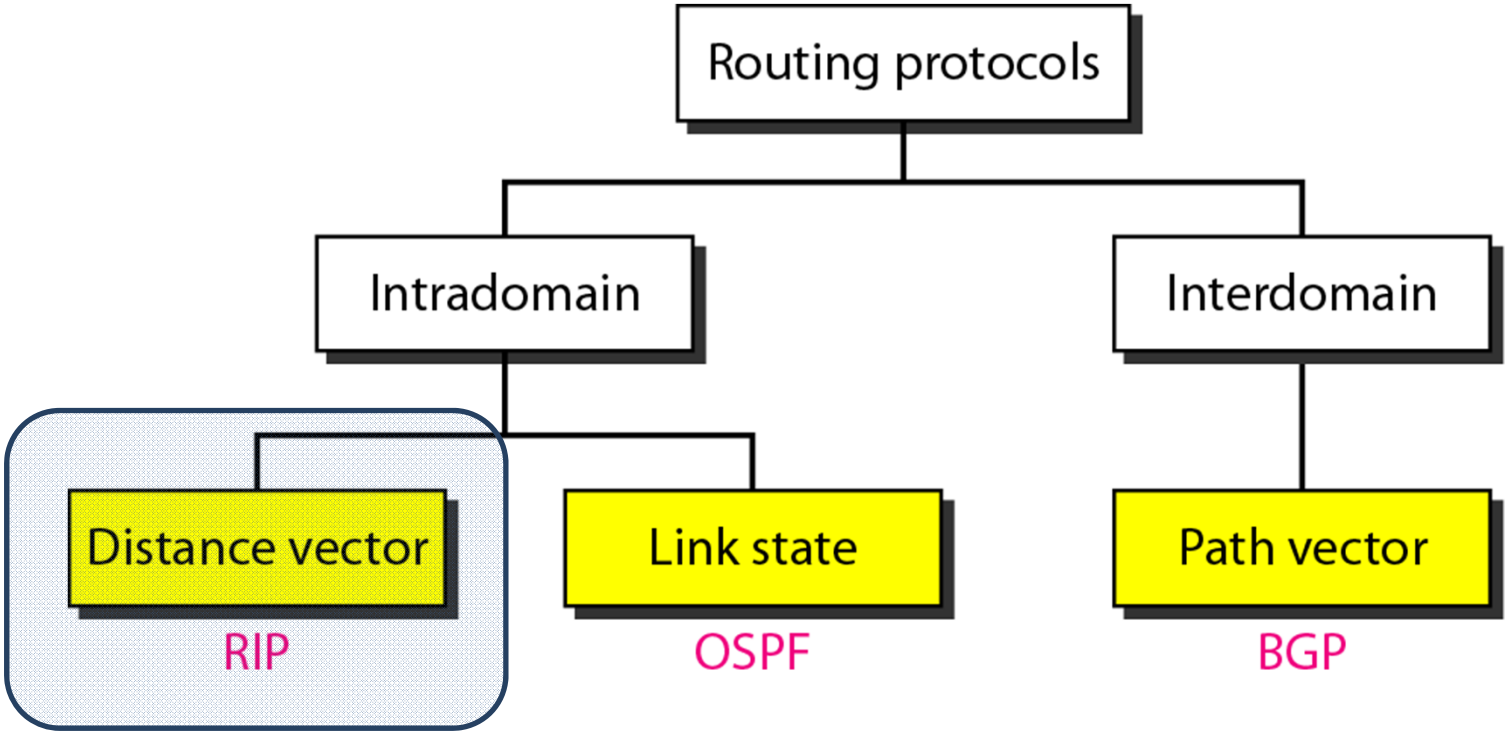
Problem: The LANs are our destinations/end nodes, not the routers

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



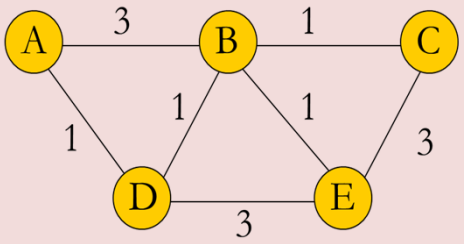
RIP (Routing Information Protocol)

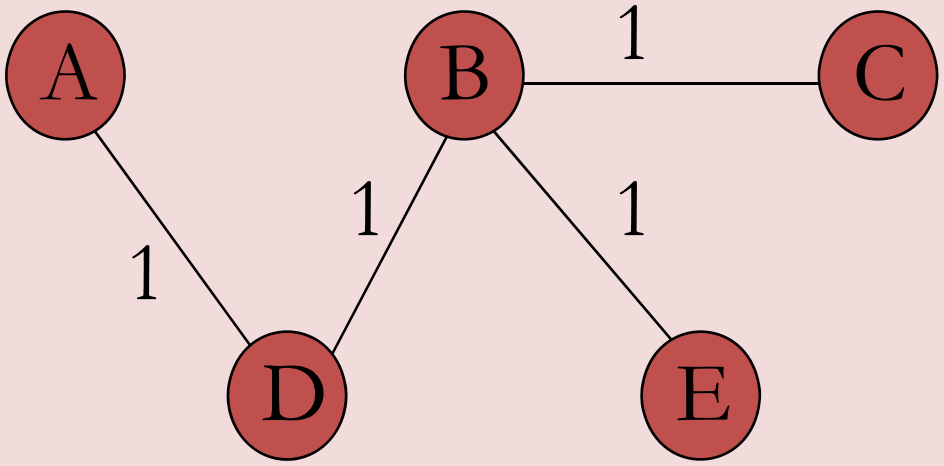
- Included in BSD-UNIX Distribution in 1982
- Distance metric:
  - **# of hops** (max 15) to destination network
- Distance vectors:
  - exchanged among neighbours every 30” via Response Message (advertisement)
- Implementation:
  - Application layer protocol, uses UDP/IP

Distance Vector Routing

- Best path info **shared** locally
  - Periodically
  - Upon any change
- Routing tables **updated** for
  - New entries
  - Cost changes
- Metric
  - Not necessarily hop count!

Nätgraf som träd

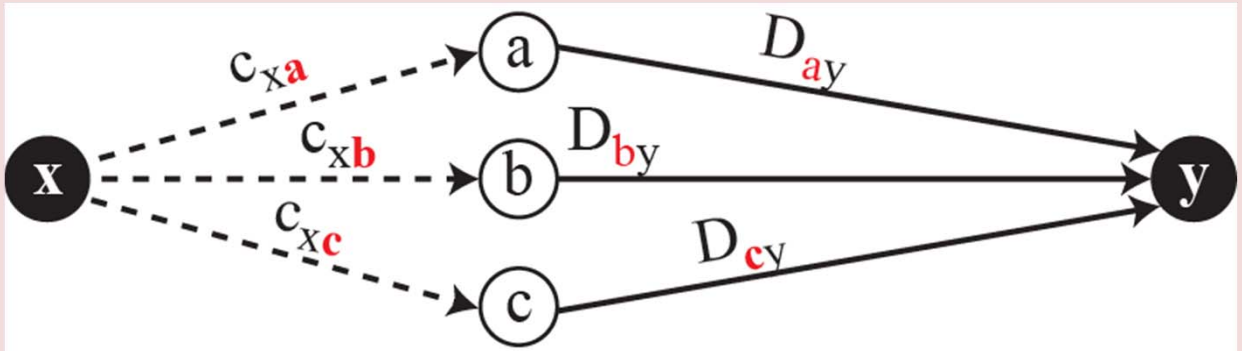




Distansvektor för A

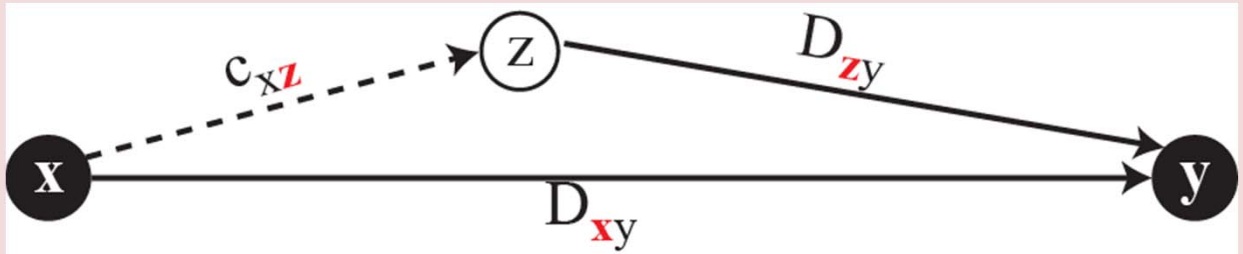
Nod	Dist
A	0
B	2
C	3
D	1
E	3

Bellman-Ford



a. General case with three intermediate nodes

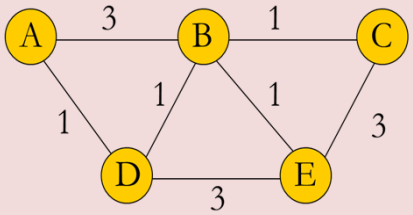
$$D_{xy} = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}) \dots\}$$



b. Updating a path with a new route

$$D_{xy} = min\{D_{xy}, (c_{xz} + D_{zy})\}$$

Uppdateringar



$$A[\,] = \min(A[\,], 3 + B[\,])$$

A, uppdaterad

Nod	Dist
A	0
B	3
C	4
D	1
E	4

A, ursprunglig

Nod	Dist
A	0
B	3
D	1

B

Nod	Dist
A	3
B	0
C	1
D	1
E	1

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

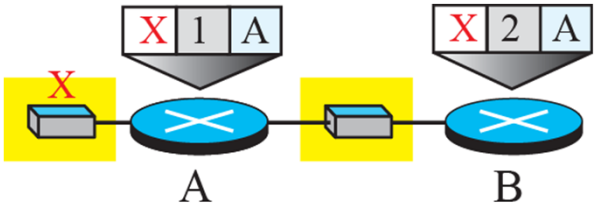
RIP Updating Algorithm (Bellman-Ford)

```
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
    }
```

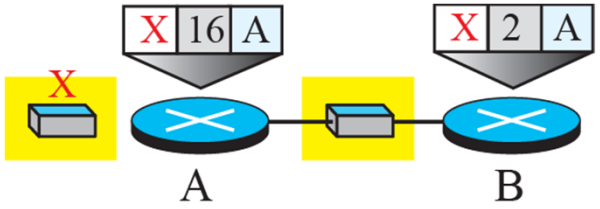
RIP Example


New R1
Des.
N. R.
Cost
N1
N2
N3
N4
N5
1
1
1
R2
R2
2
2
Old R1
Des.
N. R.
Cost
N1
N2
N3
1
1
1
R2 Seen by R1
Des.
N. R.
Cost
N3
N4
N5
R2
R2
R2
2
2
2

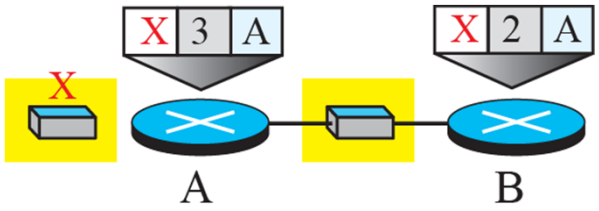
Two node instability/Count to inifinity



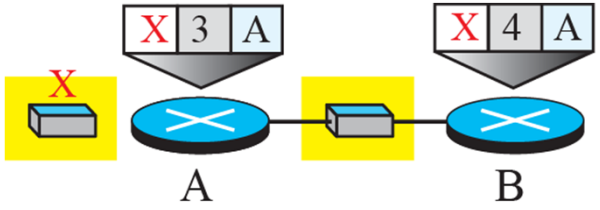
a. Before failure



b. After link failure

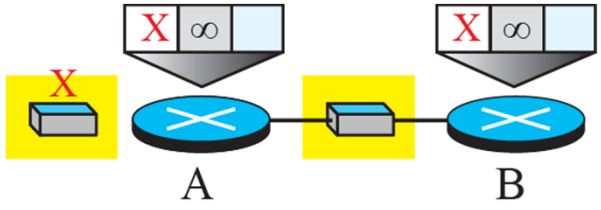


c. After A is updated by B



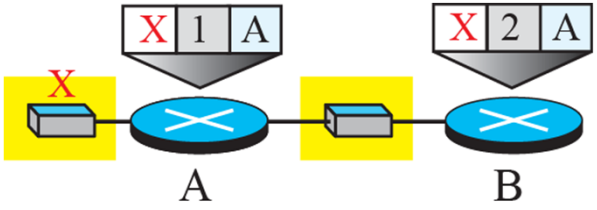
d. After B is updated by A

• • •

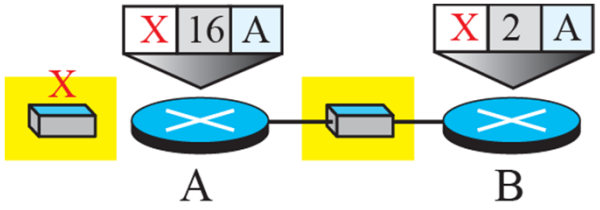


e. Finally

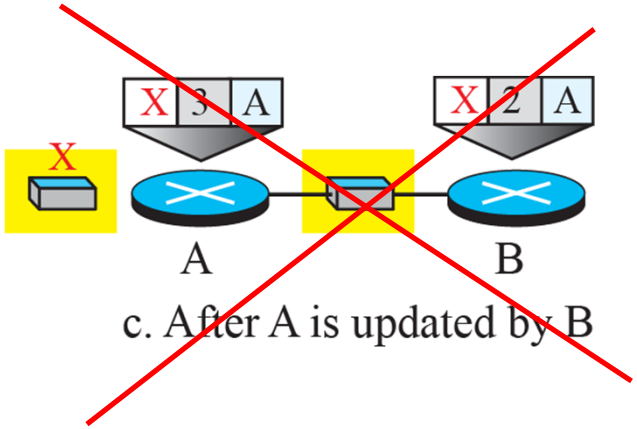
Split Horizon breaks Count to inifinity



a. Before failure



b. After link failure



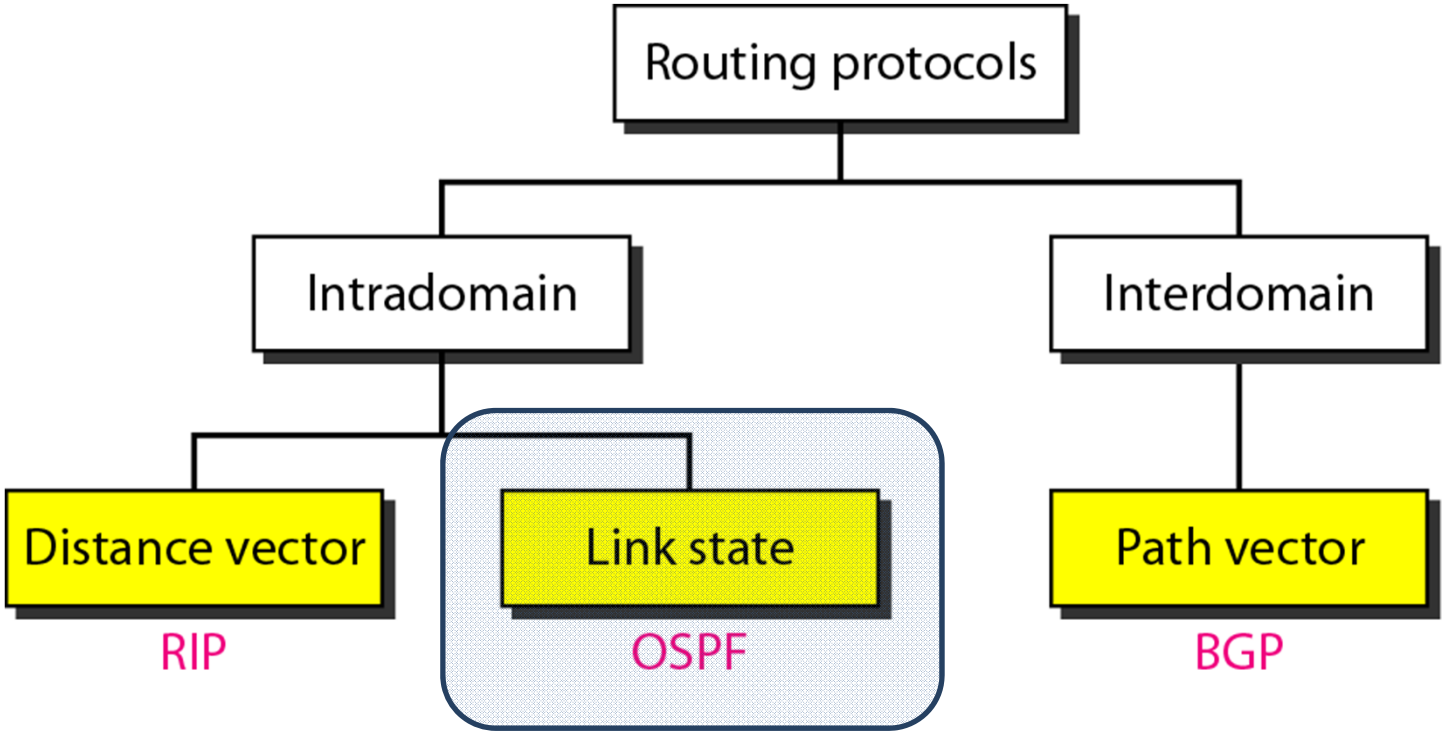
c. After A is updated by B

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



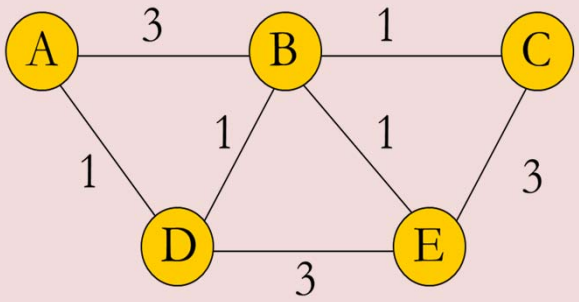
OSPF (Open Shortest Path First)

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

Link State Routing

- Local topology info **flooded globally**
  - Periodically (very seldom …)
  - Upon any change
- Routing tables **updated** for
  - Link state changes
  - Cost changes

Exempel på *link state* databas



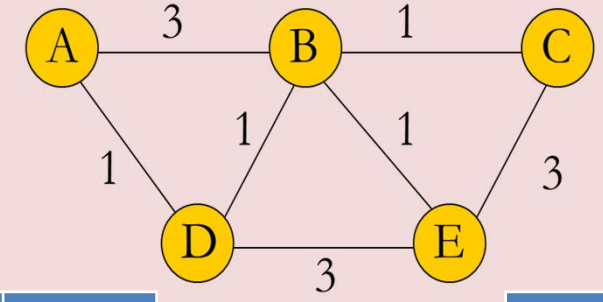
	A	B	C	D	E
A	0	3	∞	1	∞
B	3	0	1	1	1
C	∞	1	0	∞	3
D	1	1	∞	0	3
E	∞	1	3	3	0

Link State Advertisements

Nod	Kostn
B	3
D	1

Nod	Kostn
A	3
C	1
D	1
E	1

Nod	Kostn
B	1
E	3



Nod	Kostn
1	1
B	1
E	3

Nod	Kostn
B	1
C	3
D	3

Uppdateras vid förändring!

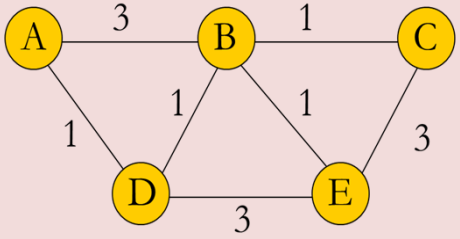
Tree Generation Algorithm (Dijkstra)

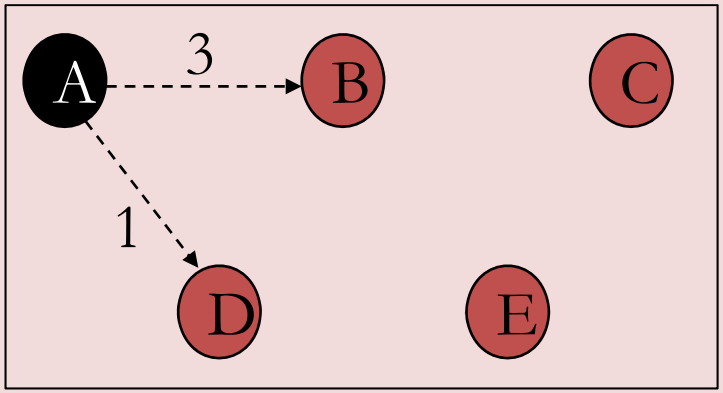
```
put yourself to tentative list
while tentative list not empty
   {
   pick node which can be reached
             with least cumulative cost
   add it to your tree*
   put its neighbours to tentative list**
             with cumulative costs to reach them
   }
```

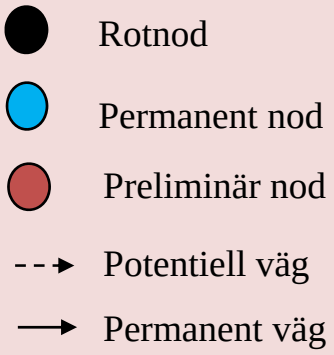
*(a.k.a. permanent list)

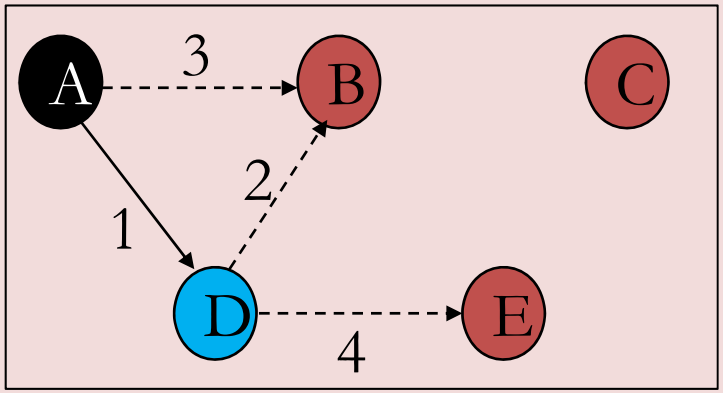
**(if not already there)

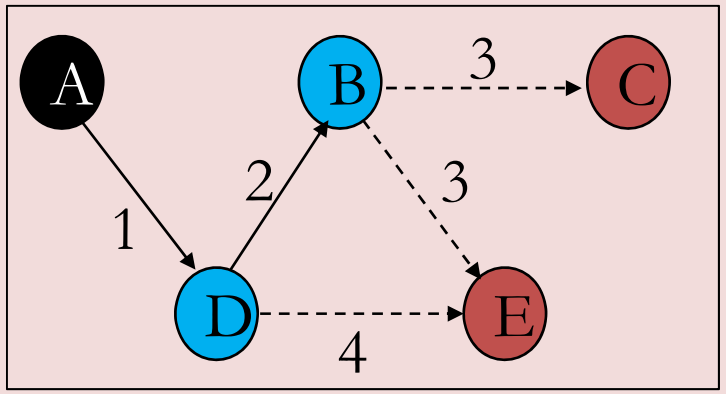
SPF exempel

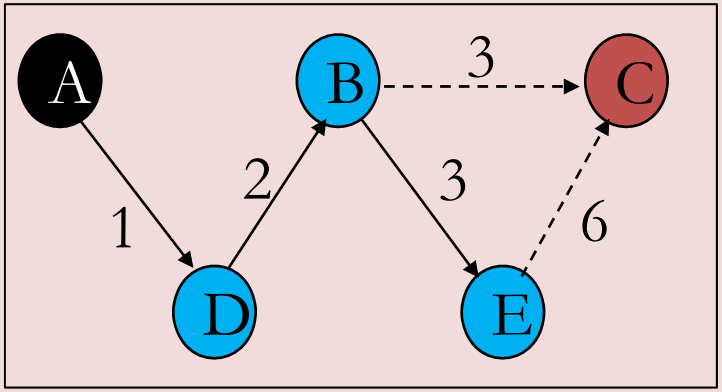

A
B
C
D
E
3
1
1
1
1
3
3

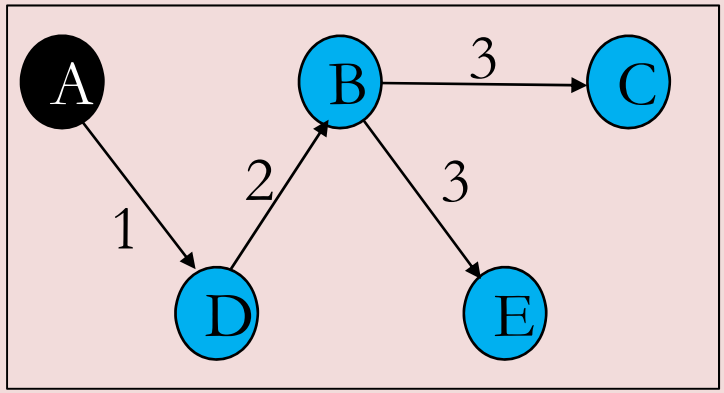

A
B
C
D
E
3
1


Rotnod
Permanent nod
Preliminär nod
Potentiell väg
Permanent väg

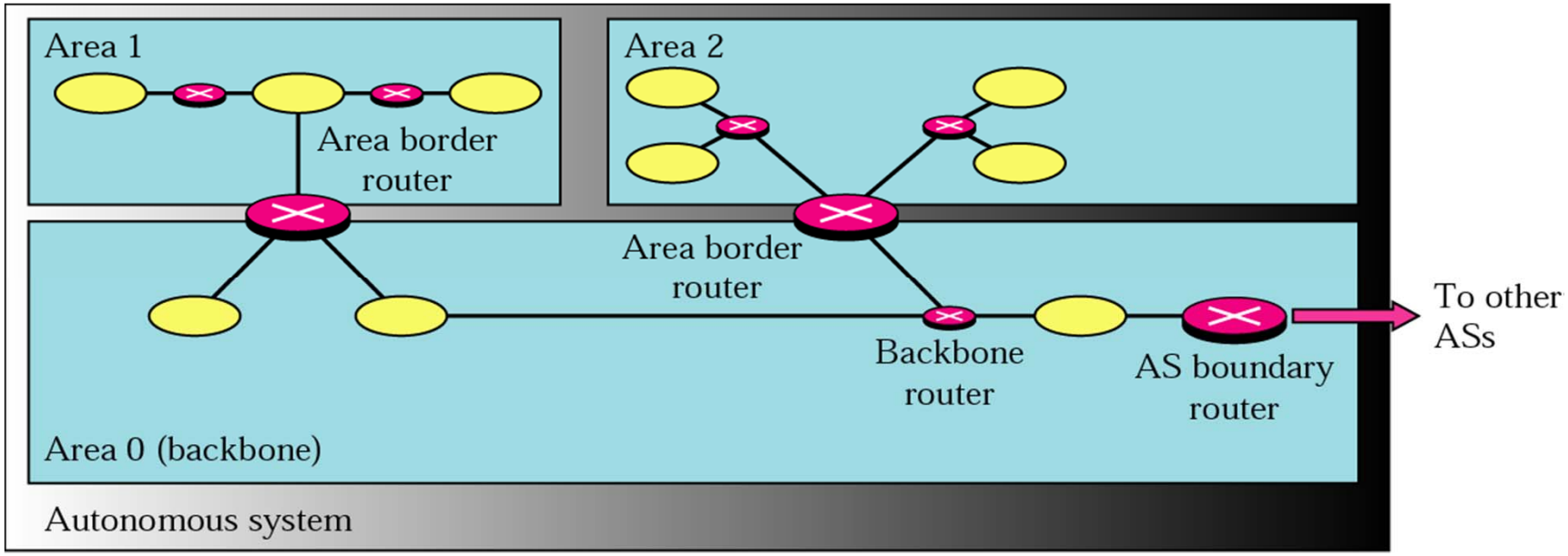

A
B
C
D
E
3
2
1
4


A
B
C
D
E
3
2
3
1
4


A
B
C
D
E
3
2
3
1
6

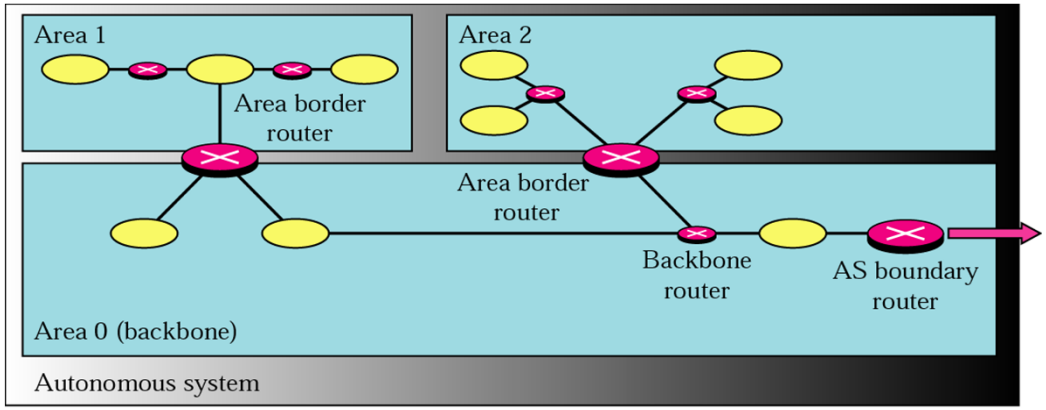

A
B
C
D
E
3
2
3
1

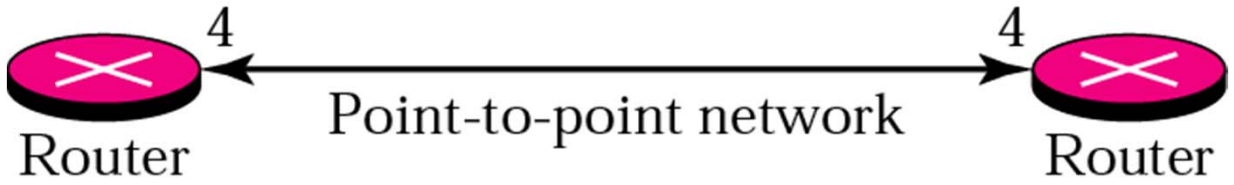
Areas, Router and Link Types



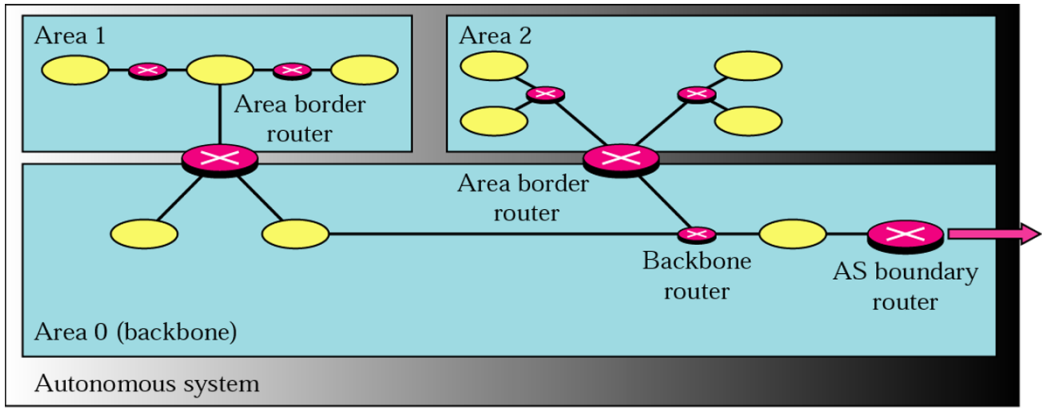
Point-to-Point Link

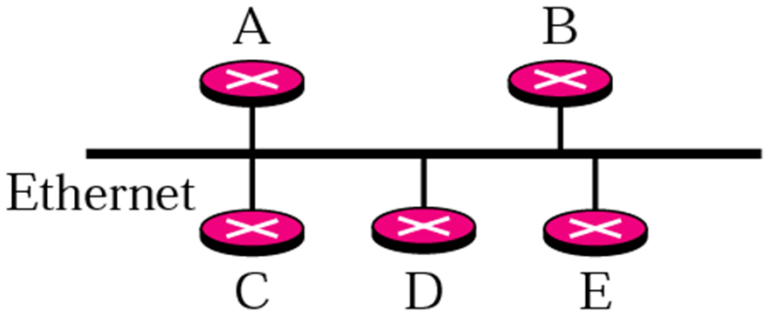
- Connects two routers
- No need for addresses



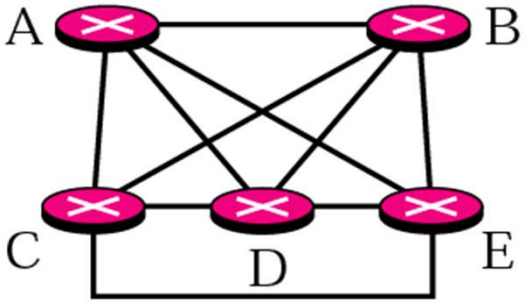


Transient Link

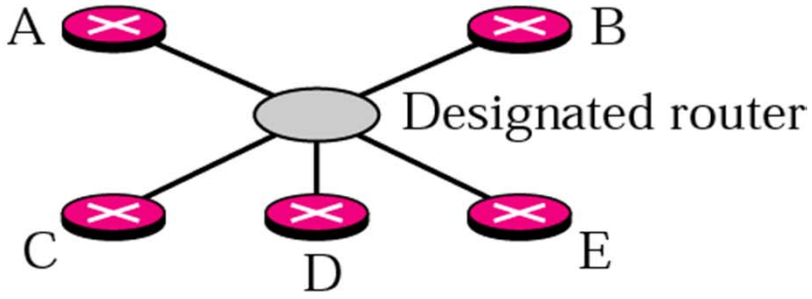




a. Transient network

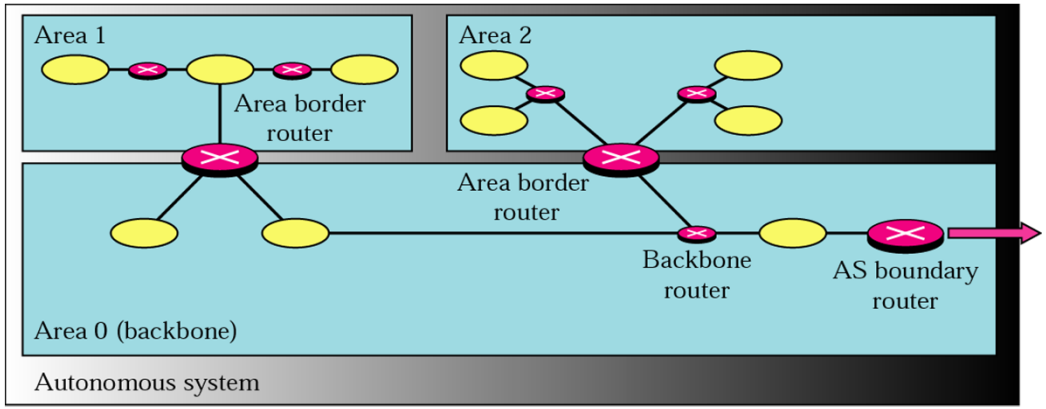


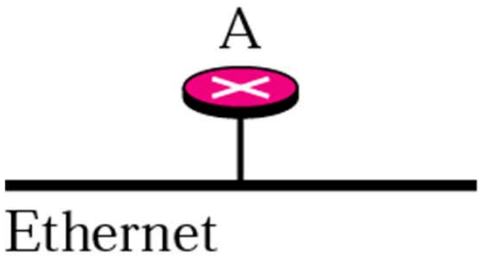
b. Unrealistic representation



c. Realistic representation

Stub Link



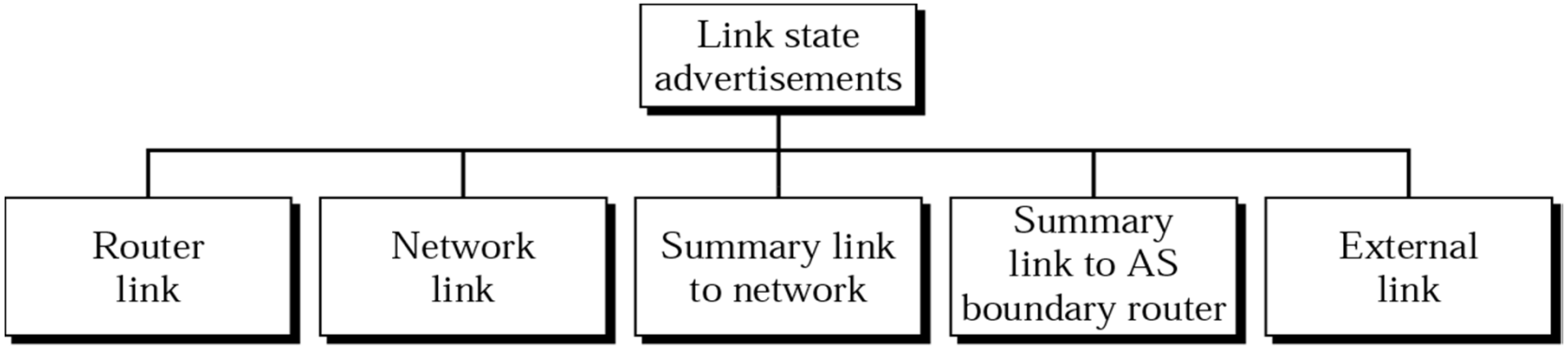


a. Stub network

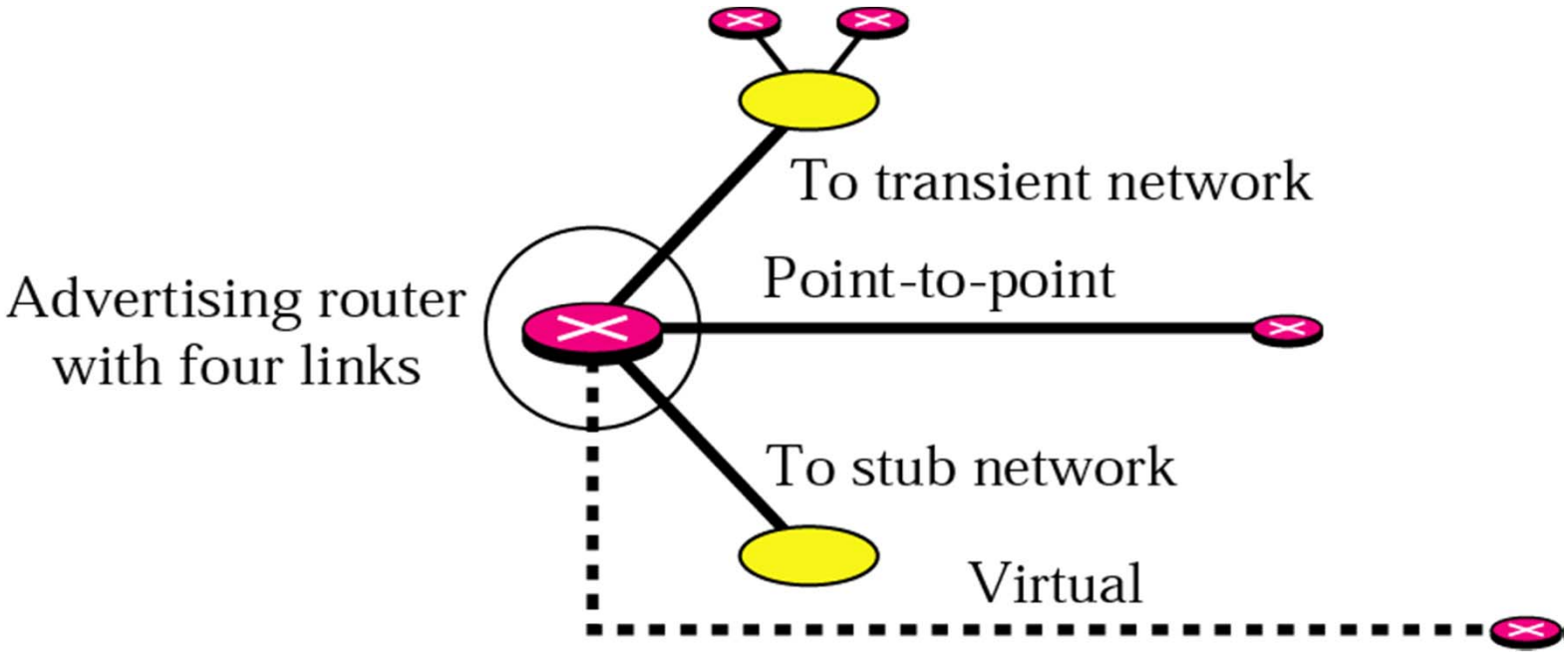
b. Representation

Link State Advertisements

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

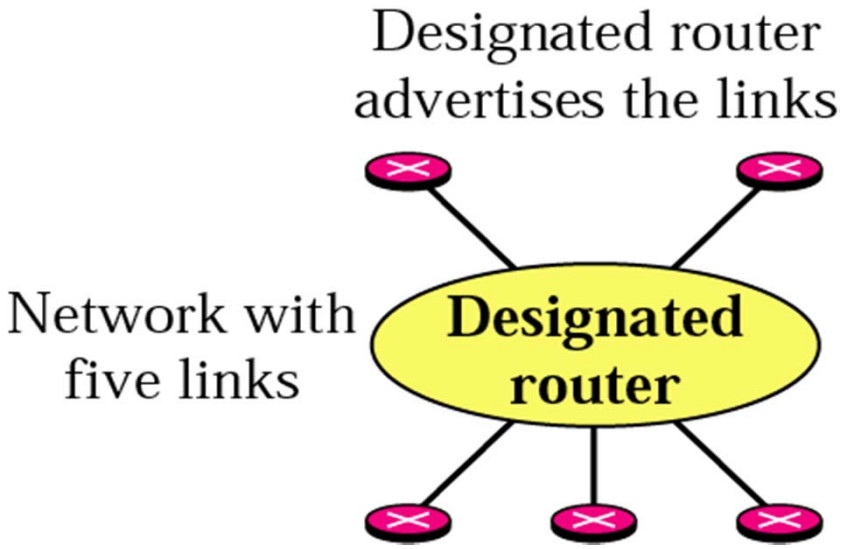


Router Link Advertisement



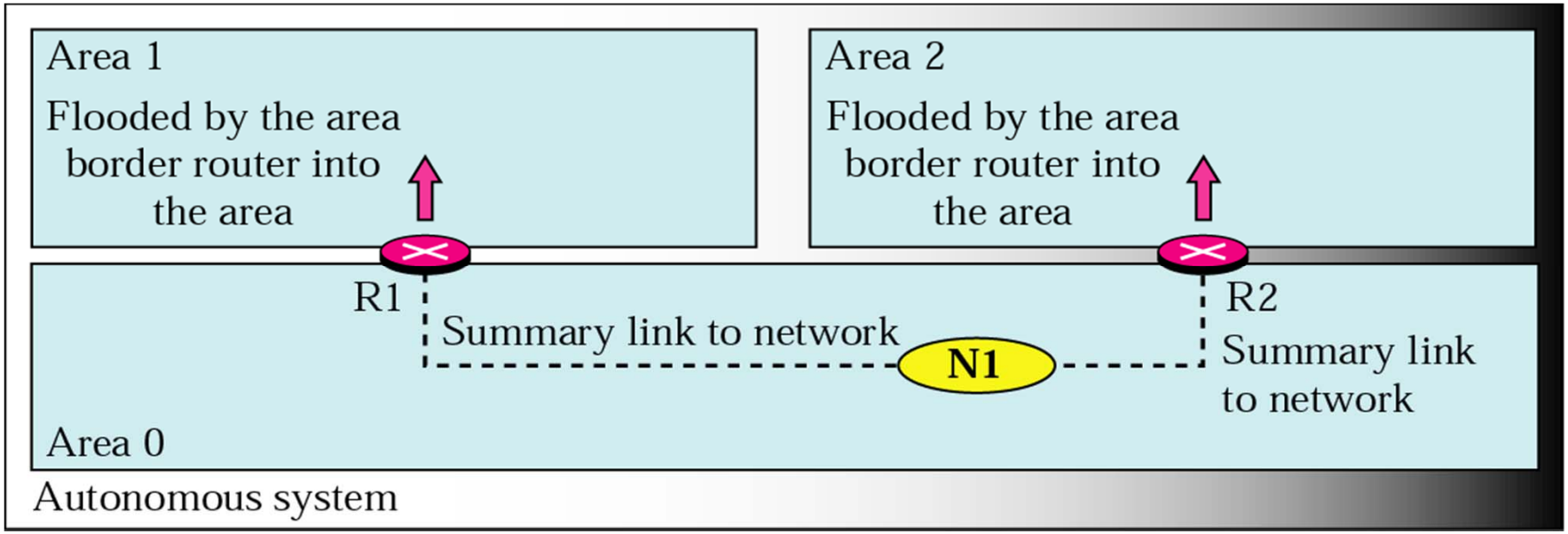
Network Link Advertisement

- Network is a passive entity
  - It cannot advertise itself



Summary Link to Network

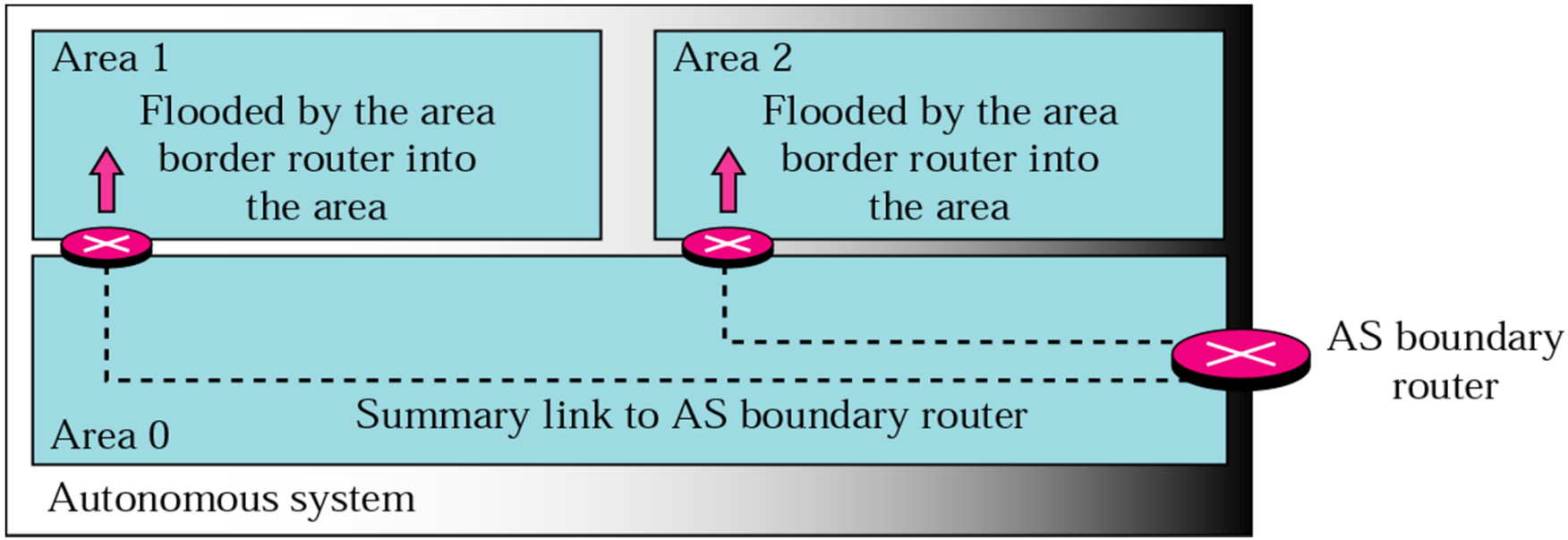
- Done by area border routers
  - Goes through the backbone



Summary Link to AS Boundary Router

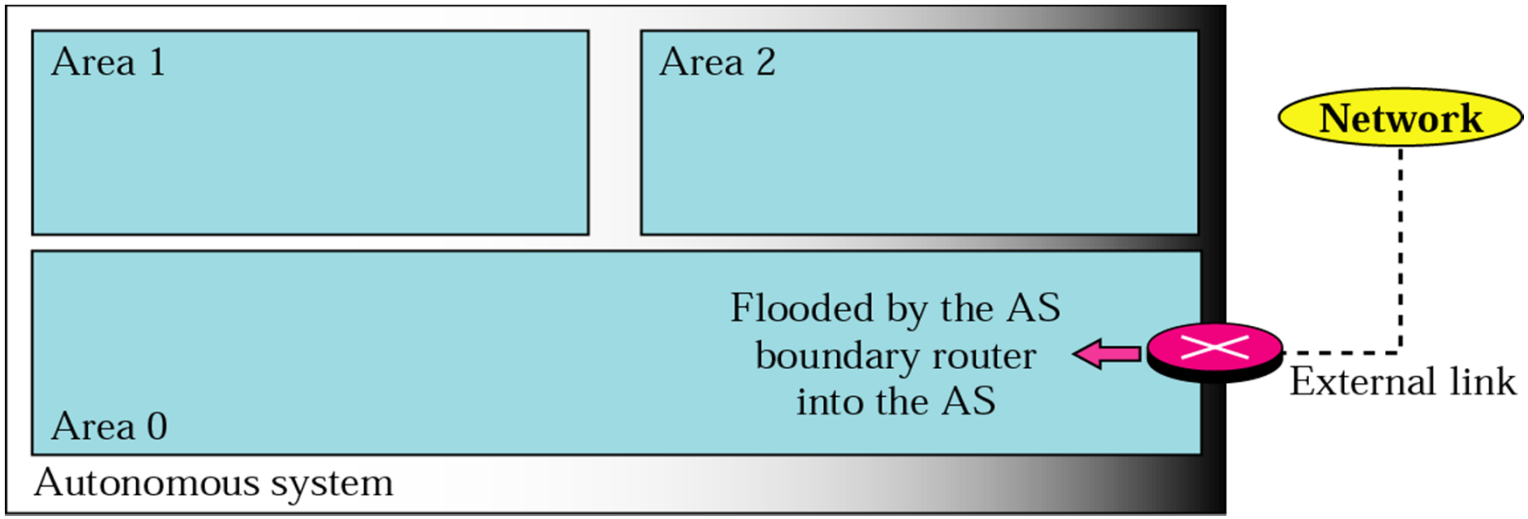
- Links to other domains

  ”autonomous systems”



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