Routing
(Part 1)

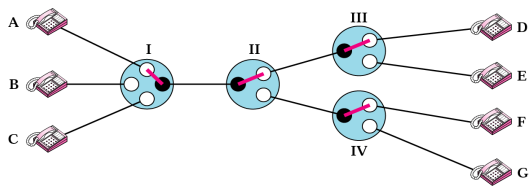
ETSF10 – Internet Protocols – 2011

Kaan Bür & Jens Andersson
Department of Electrical and Information Technology

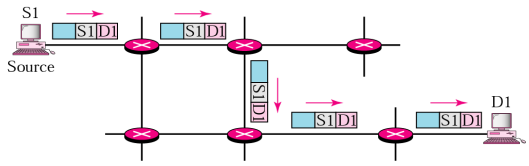
Routing

- **Introduction**
- Inside the Router §8.4
- Unicast Routing §22.3
  - Intradomain Routing
  - Interdomain Routing
- Multicast Routing §22.4

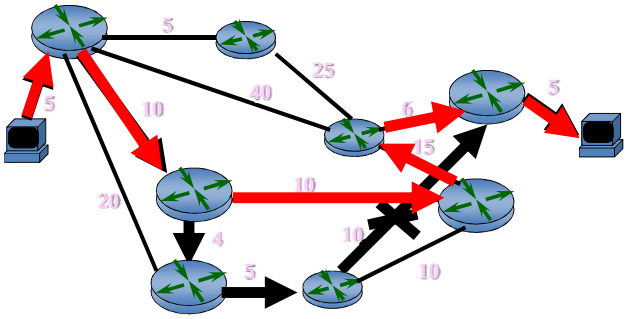
Circuit-switched Routing



Packet-switched Routing



Choosing an Optimal Path



Router

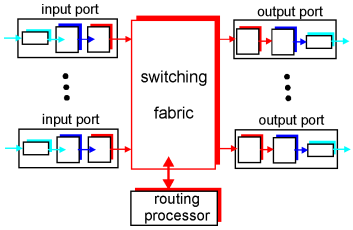
- A router is a type of internetworking device that *passes data packets* between networks, based on Layer 3 or Network Layer addresses.
- A router has the ability to *make intelligent decisions regarding the best path* for delivery of data on the network.

Routing

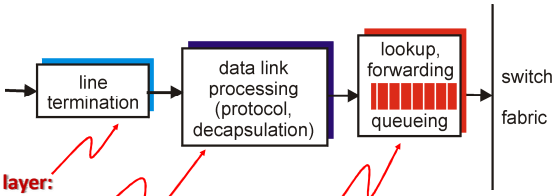
- Introduction
- **Inside the Router** §8.4
- Unicast Routing §22.3
  - Intradomain Routing
  - Interdomain Routing
- Multicast Routing §22.4

Router Architecture Overview

- Two key functions:
  - run routing algorithms/protocols (RIP, OSPF, BGP)
  - switch datagrams from incoming to outgoing link



Input Port



Physical layer:
bit-level reception

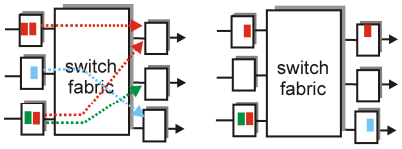
Data link layer:
e.g., Ethernet

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 → queueing
- Head-of-the-Line (HOL) blocking: Datagram at front of queue prevents others in queue from proceeding
- Delay and loss due to input buffer overflow

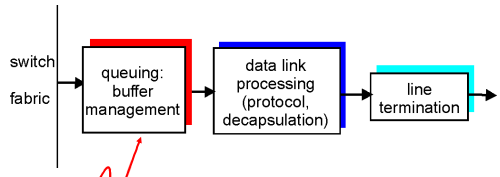


output port contention at time t - only one red packet can be transferred

green packet experiences HOL blocking

2011-10-24 10

Output Port



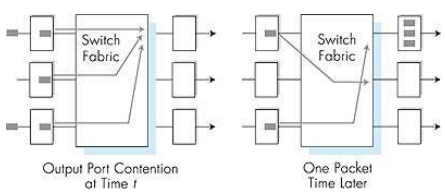
Priority Scheduling:

- Scheduling discipline may choose among queued datagrams for transmission

2011-10-24 11

Output Port Queueing

- Datagrams' arrival rate through the switch exceeds the transmission rate of the output line → buffering
- Delay and loss due to output port buffer overflow

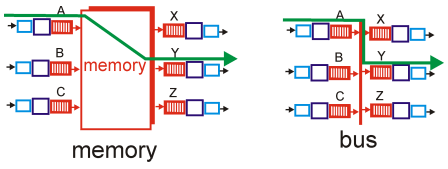


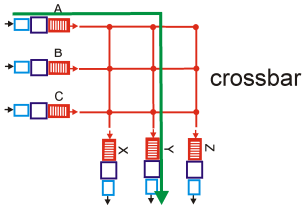
Output Port Contention at Time *t*

One Packet Time Later

2011-10-24 12

Switching Fabrics

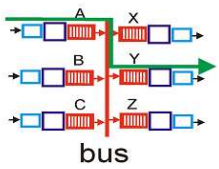




2011-10-24 13

Switching via Bus

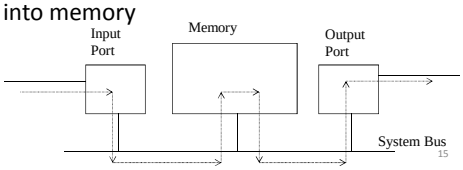
- Datagram from input port buffer to output port buffer via shared bus
- Bus contention: Switching speed limited by bus bandwidth



2011-10-24 14

Switching via Memory

- First generation routers:
  - Packet copied by system's (single) CPU
  - Speed limited by memory bandwidth
- Modern routers:
  - Input port processor performs lookup
  - A copy into memory



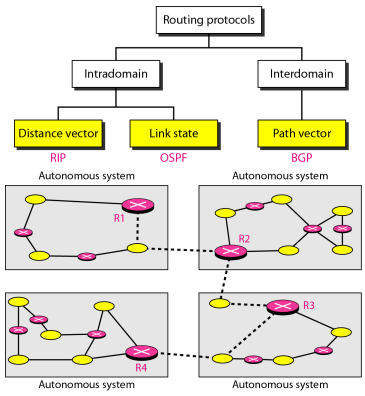
2011-10-24 15

Routing

- Introduction
- Inside the Router §8.4
- **Unicast Routing §22.3**
  - **Intradomain Routing**
  - Interdomain Routing
- Multicast Routing §22.4

2011-10-24 16

Unicast Routing



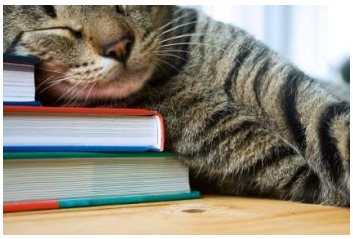
2011-10-24

Link (Cost) Metrics

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

2011-10-24 18

See you in 15' :)



- After the break
  - Distance vector routing
  - Link state routing

Intradomain Routing

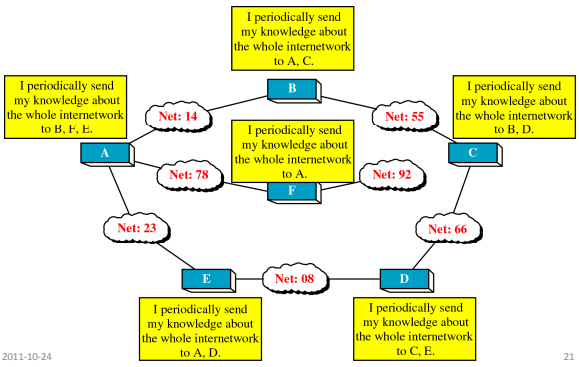
Link State

- Topology information is flooded within the routing domain
- Best end-to-end paths are computed locally at each router
- Best end-to-end paths determine next-hops
- Works only if policy is shared and uniform
- Examples: OSPF, IS-IS

Distance Vector

- Each router knows a little about network topology
- Only best next-hops are chosen by each router for each destination network
- Best end-to-end paths result from composition of all best next-hop choices
- Does not require uniform policies at all routers
- Examples: RIP, BGP

Distance Vector Routing Concept



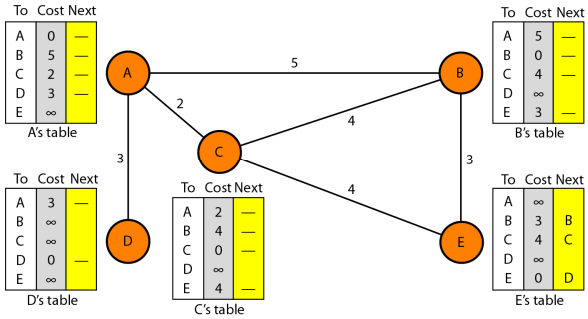
Distance Vector Routing Table

Destination	Hop Count	Next Router	Other information
163.5.0.0	7	172.6.23.4	
197.5.13.0	5	176.3.6.17	
189.45.0.0	4	200.5.1.6	
115.0.0.0	6	131.4.7.19	

Network id Metric Host id of interface

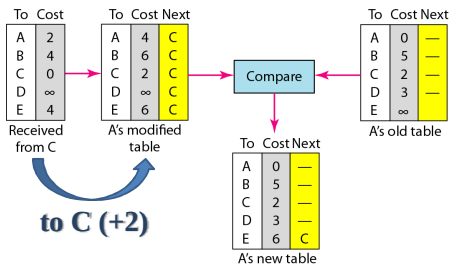
2011-10-24

Initialisation of routing tables



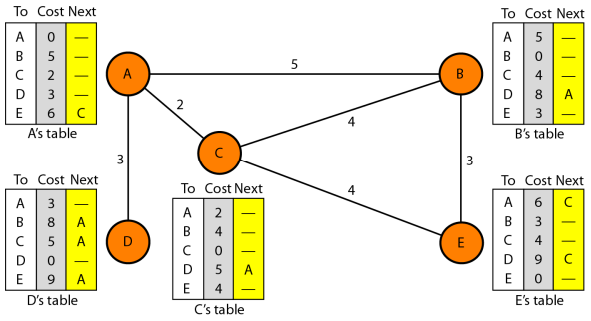
2011-10-24 23

Updating a routing table



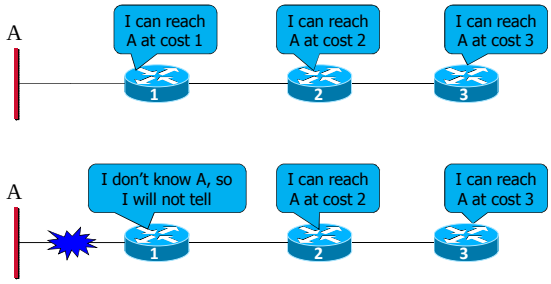
2011-10-24 24

Distance vector concept



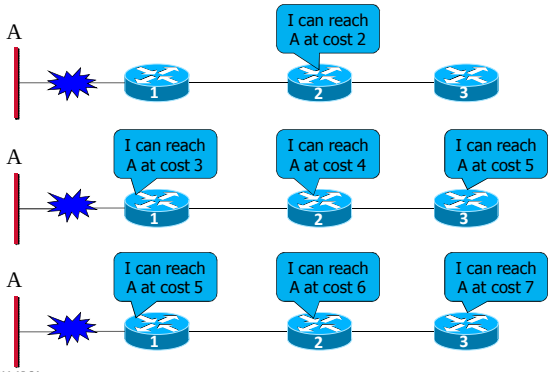
2011-10-24 25

Problem: Count to Infinity (1)



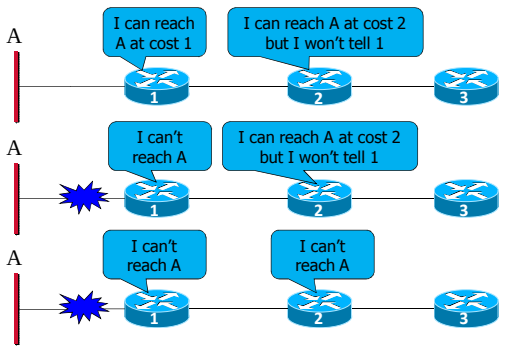
2011-10-24 26

Problem: Count to Infinity (2)



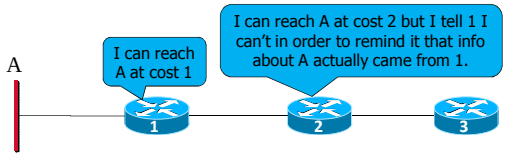
2011-10-24 27

Solution: Split Horizon



2011-10-24 28

Split Horizon with Poison Reverse



2011-10-24 29

RIP (Routing Information Protocol)

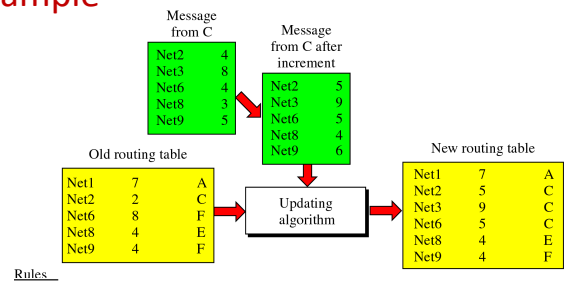
- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max 15)
- Distance vectors: exchanged among neighbours every 30 sec via Response Message (also called **advertisement**)
- Each advertisement: list of up to 25 destination nets within AS

2011-10-24 30

RIP Updating Algorithm

```
(1)      if (advertised destination not in table)
            update table
(2)      else
(2.a)    if (advertised next hop = next hop in table)
            replace entry
(2.b)    else
(2.b.i)  if (advertised hop count < hop count in table)
            replace entry
(2.b.ii) else
            do nothing
```


Example



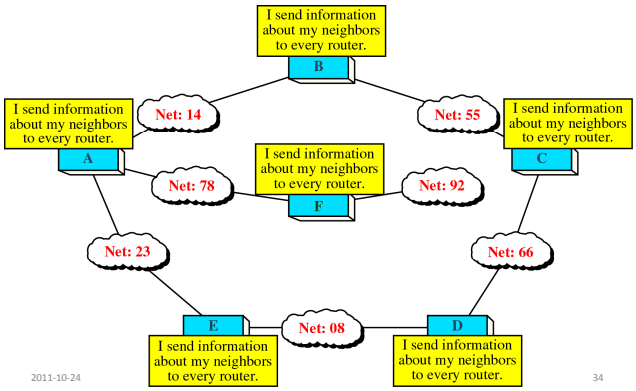
Rules
Net2: Replace **(Rule 2.a)**
Net3: Add **(Rule 1)**
Net6: Replace **(Rule 2.b.i)**
Net8: No change **(Rule 2.b.ii)**
Net9: No change **(Rule 2.b.ii)**

Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

RIP: Link Failure and Recovery

- If no advertisement heard after 180”
  - Neighbor/link declared dead
  - Routes via neighbour invalidated
  - New advertisements sent to neighbours
  - Neighbours in turn send out new advertisements (if tables changed)
  - Link failure info quickly propagates to entire net
  - Poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

Link State Routing Concept



2011-10-24 34

Link State Database

Advertiser	Network	Cost	Neighbor
A A A	14 78 23	1 3 2	B F E
B **B**	**14** **55**	**4** **2**	**A** **C**
C C	55 66	5 2	B D
D **D**	**66** **08**	**5** **3**	**C** **E**
E E	23 08	3 2	A D
F **F**	**78** **92**	**2** **3**	**A** **—**

2011-10-24 35

Link State Routing Table

Network	Cost	Next Router	Other Information
N1	5	C	
N2	7	D	
N3	10	B	
N4	11	D	
N5	15	C	

Network id Metric Host id of interface

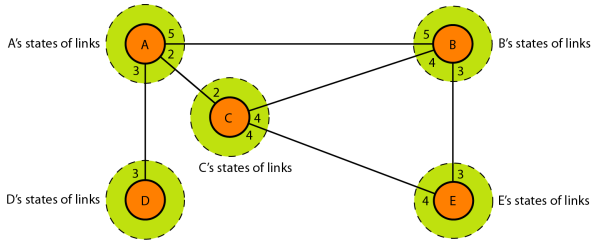
2011-10-24

The Dijkstra Algorithm

1. Identify the root (the node itself)
2. Attach all neighbour nodes temporarily
3. Make arc and node with least cumulative cost permanent
4. Choose this node
5. Repeat 2 and 3 until all nodes are permanent

2011-10-24 37

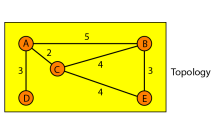
Initial link state knowledge



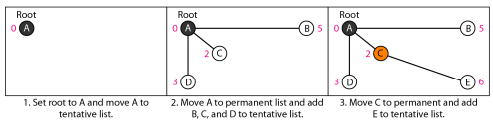
2011-10-24 38

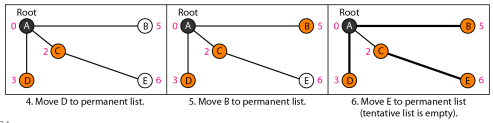
Shortest path tree generation

- **A**



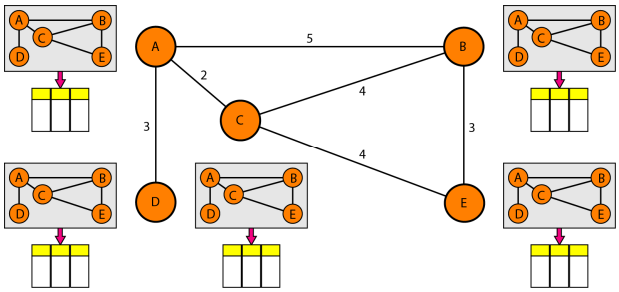
Node	Cost	Next Router
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C





2011-10-24 39

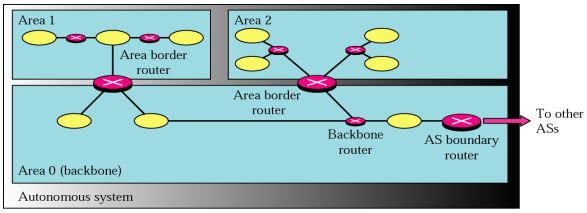
Link state concept



2011-10-24 40

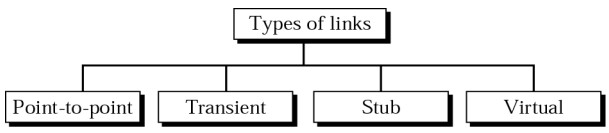
OSPF (Open Shortest Path First)

- Divides AS into areas for efficiency
- Networks represented as nodes



2011-10-24 41

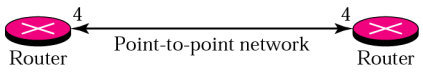
Types of Links



2011-10-24 42

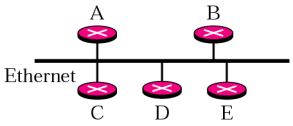
Point-to-Point Link

- Connects two routers
- No need for addresses

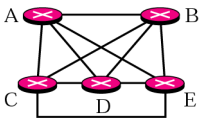


2011-10-24 43

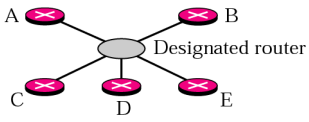
Transient Link



a. Transient network



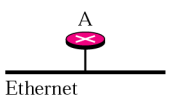
b. Unrealistic representation



c. Realistic representation

2011-10-24 44

Stub Link

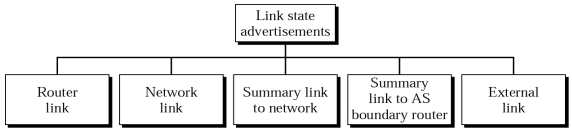


a. Stub network

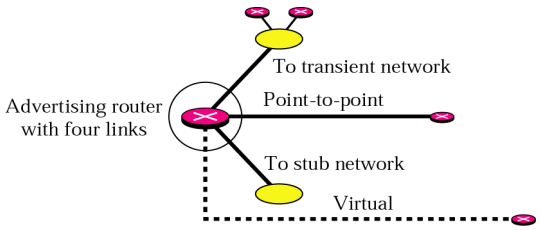
b. Representation

2011-10-24 45

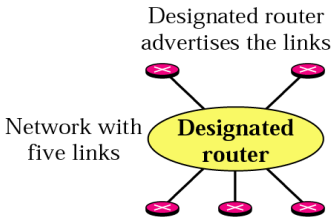
Link State Advertisements



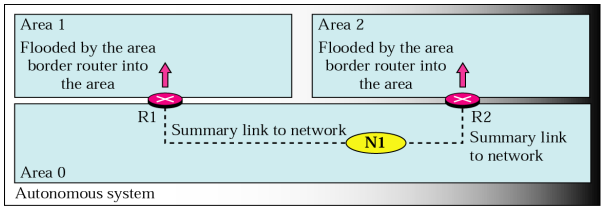
Router Link Advertisement



Network Link Advertisement

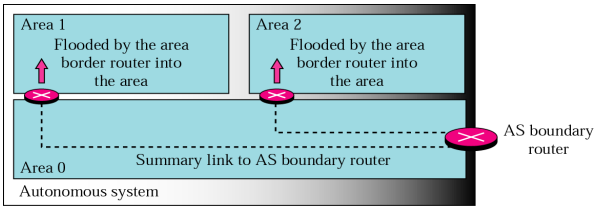


Summary Link to Network



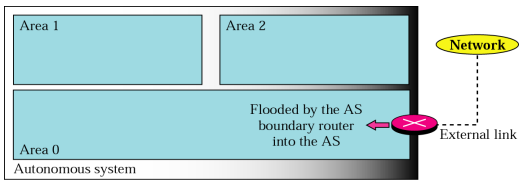
2011-10-24 49

Summary Link to AS Boundary Router



2011-10-24 50

External Link Advertisement



2011-10-24 51