

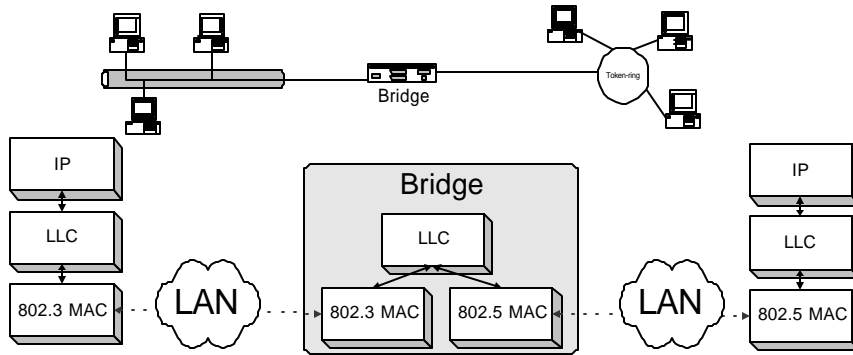
Ethernet (LAN switching)

Outline

- Interconnection devices
- Bridges/LAN switches vs. Routers
- Bridges
- Learning Bridges
- Transparent bridges

Bridges/LAN switches

- Interconnect multiple LAN, possibly with different type
- Bridges operate at the Data Link Layer (Layer 2)
- The term **LAN switches** and **Bridges** are synonymous

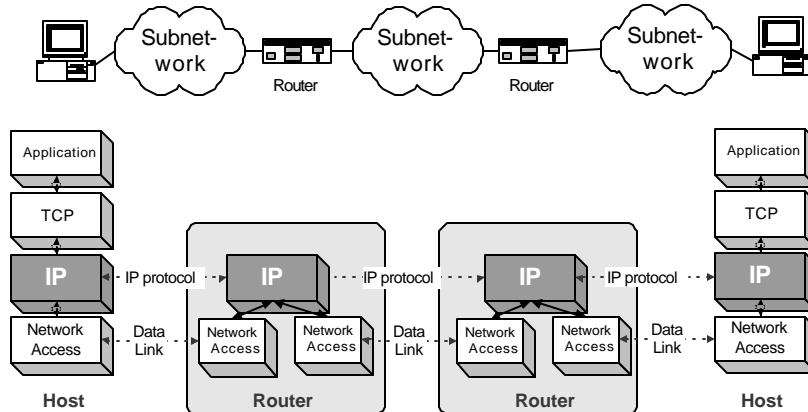


© Jörg Liebeherr 1998-2000

3

Routers

- Routers operate at the Network Layer (Layer 3)
- Interconnect different subnetworks

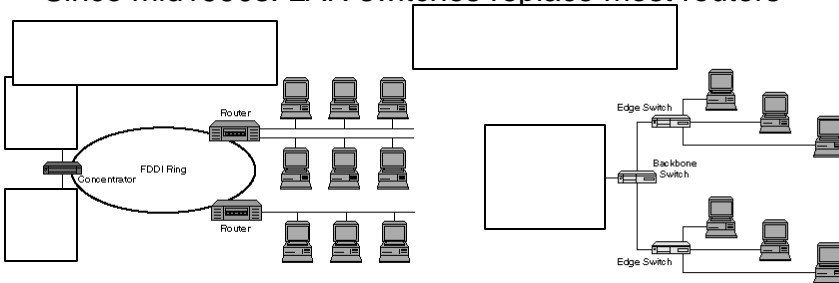


© Jörg Liebeherr 1998-2000

4

Bridges versus Routers

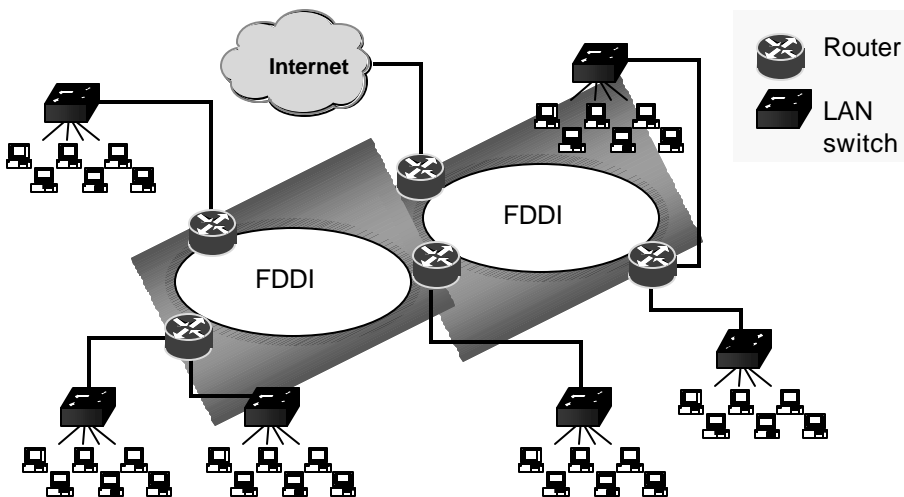
- An enterprise network (e.g., university network) with a large number of local area networks (LANs) can use routers or bridges
- Until early 1990s: most LANs were interconnected by routers
- Since mid-1990s: LAN switches replace most routers



© Jörg Liebeherr 1998-2000

5

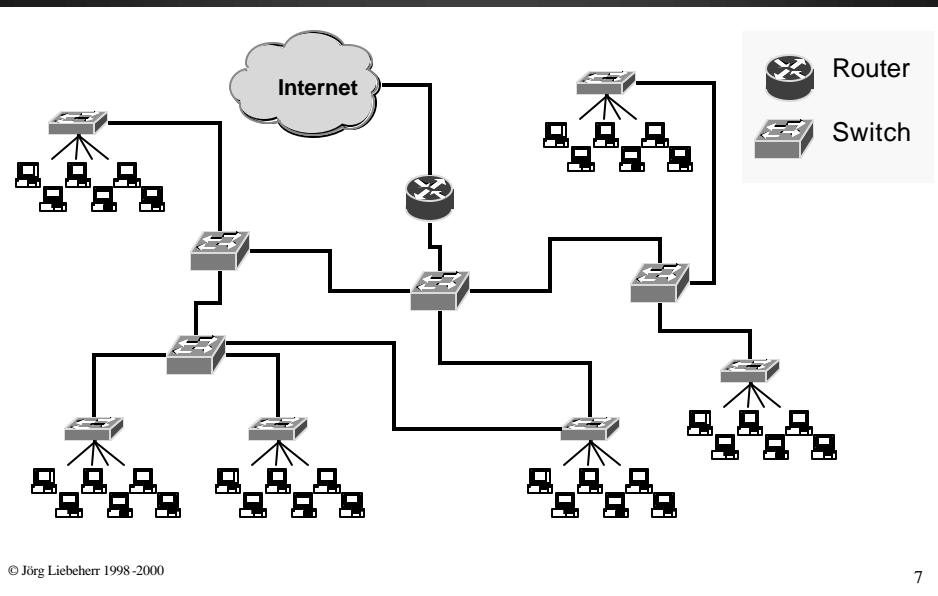
A Routed Enterprise Network



© Jörg Liebeherr 1998-2000

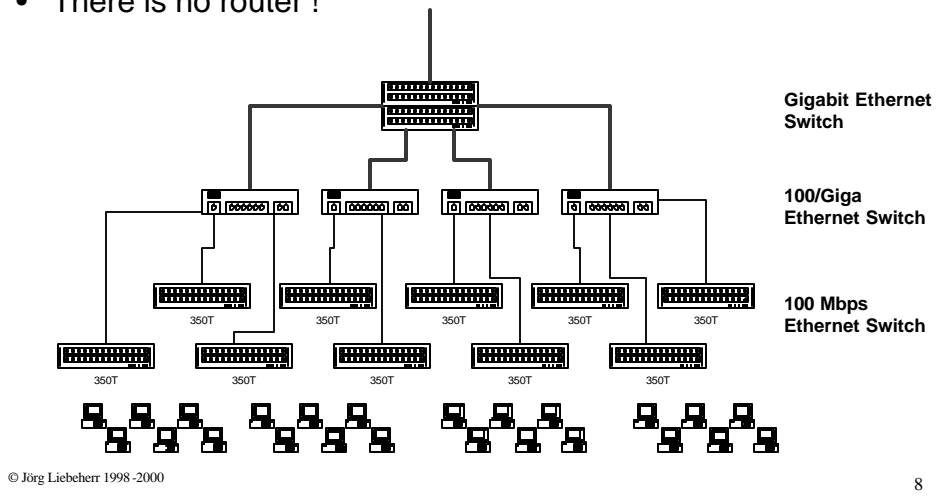
6

A Switched Enterprise Network



Example: Univ. of Virginia CS Department Network

- Design of the network architecture (Spring 2000)
- There is no router !



Bridges versus Routers

Routers

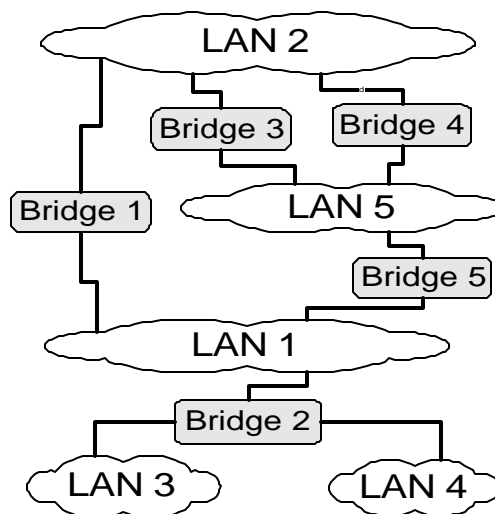
- Each host's IP address must be configured
- If network is reconfigured, IP addresses may need to be reassigned
- Routing done via RIP or OSPF
- Each router manipulates packet header (e.g., reduces TTL field)

Bridges

- MAC addresses are hardwired
- No network configuration needed
- No routing protocol needed (sort of)
 - **learning bridge algorithm**
 - **spanning tree algorithm**
- Bridges do not manipulate frames

Need for Routing

- What do bridges do if some LANs are reachable only in multiple hops ?
- What do bridges do if the path between two LANs is not unique ?



Routing for Bridges

Overall design goal: **Complete transparency**
“Plug-and-play”
Self-configuring without hardware or software changes
Bridges should not impact operation of existing LANs

Three parts to transparent bridges:
(1) Forwarding of Frames
(2) Learning of Addresses
(3) Spanning Tree Algorithm

(1) Frame Forwarding

- Each bridge maintains a **forwarding database** with entries
< **MAC address, port, age** >

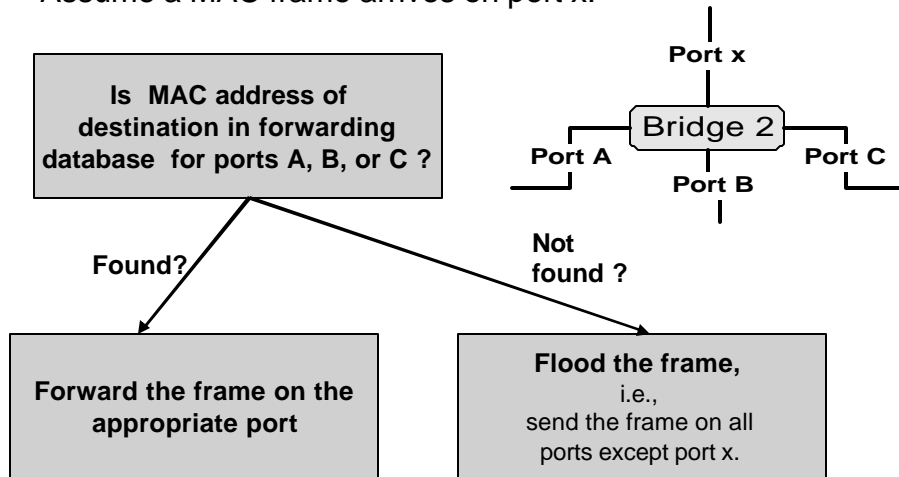
MAC address:	host name or group address
port:	port number of bridge
age:	aging time of entry

with interpretation:

- a machine with **MAC address** lies in direction of the **port** number from the bridge. The entry is **age** time units old.

(1) Frame Forwarding

- Assume a MAC frame arrives on port x.



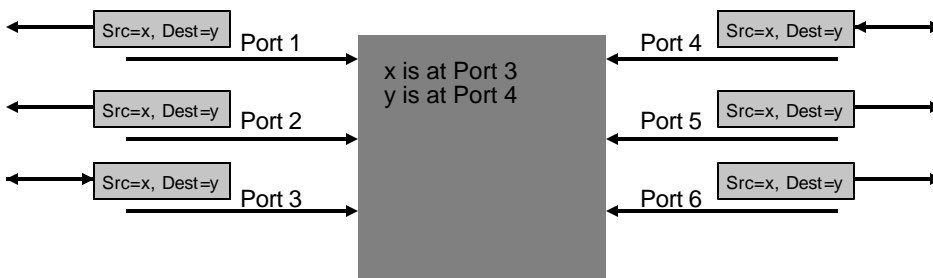
© Jörg Liebeherr 1998-2000

13

(2) Address Learning (Learning Bridges)

- Routing tables entries are set automatically with a simple heuristic:

The source field of a frame that arrives on a port tells which hosts are reachable from this port.



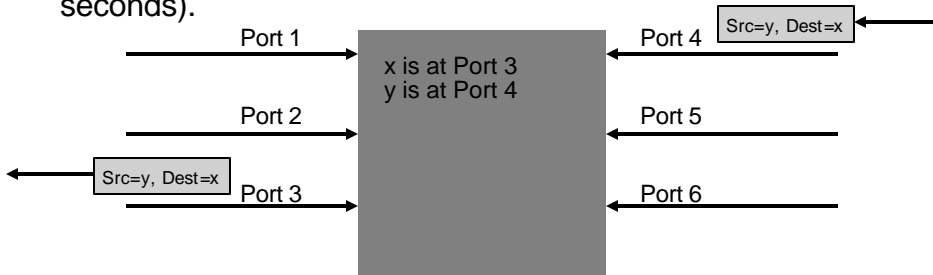
© Jörg Liebeherr 1998-2000

14

(2) Address Learning (Learning Bridges)

Algorithm:

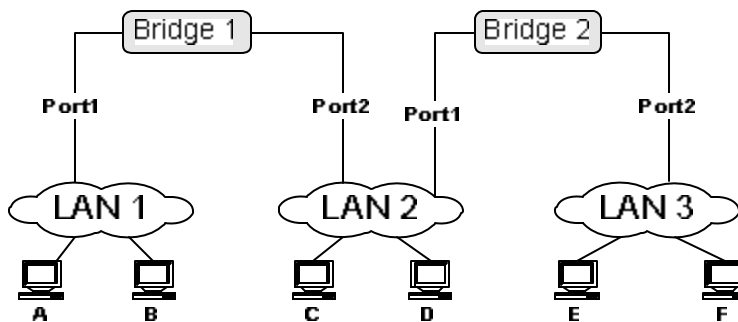
- For each frame received, the bridge stores the source field in the forwarding database together with the port where the frame was received.
- All entries are deleted after some time (default is 15 seconds).



© Jörg Liebeherr 1998-2000

15

Example



© Jörg Liebeherr 1998-2000

16

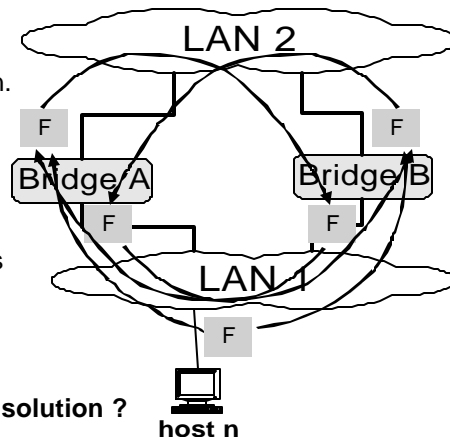
Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume *host n* is transmitting a frame *F* with unknown destination.

What is happening?

- Bridges A and B flood the frame to LAN 2.
- Bridge B sees *F* on LAN 2 (with unknown destination), and copies the frame back to LAN 1
- Bridge A does the same.
- The copying continues

Where's the problem? What's the solution ?



Spanning Trees / Transparent Bridges

- A solution is to prevent loops in the topology
- IEEE 802.1d has an algorithm that builds and maintains a spanning tree in a dynamic environment using a distributed spanning tree (SPT) algorithm
- Bridges that run 802.1d are called transparent bridges
- Bridges exchange messages to configure the bridge (Configuration Bridge Protocol Data Unit, Configuration BPDUs) to build the tree.

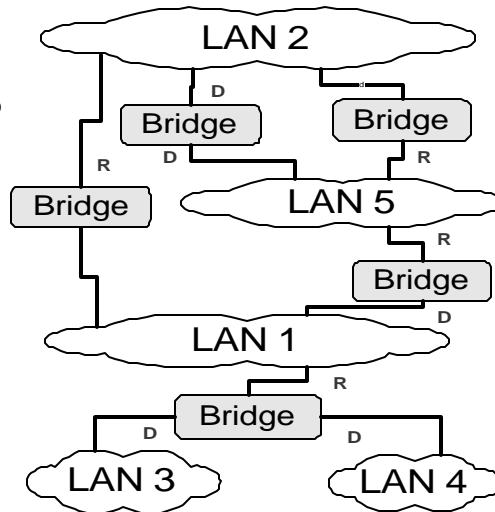
Building the Spanning Tree

Result of the SPT algorithm:

- One bridge is elected as the **root bridge**.
- Each bridge selects a **root port ("R")**, which is the port that leads to the shortest path to the root bridge
- For each LAN there is one bridge that has a **designated port ("D")** for this LAN
- All port of a bridge that are neither the root port, nor a designated port, are blocked.
- Bridges never forward traffic on blocked ports.

Question:

- What is the spanning tree in the figure?
- How does this algorithm prevent loops?



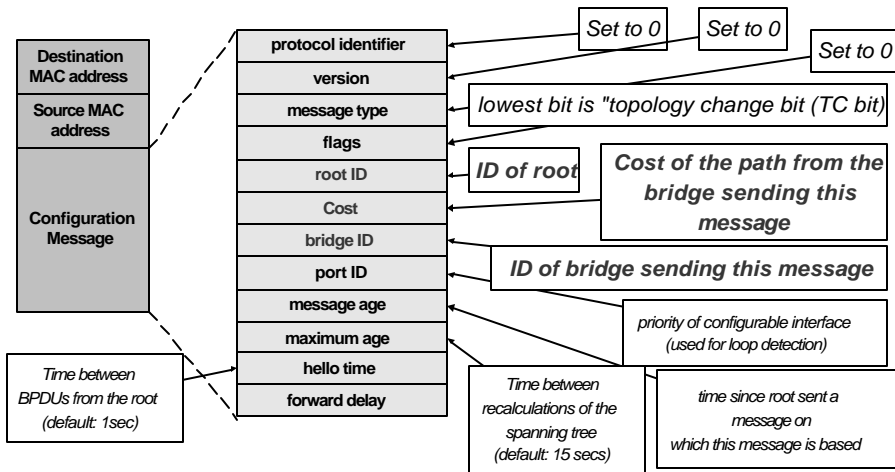
Details of the IEEE 802.1d SPT algorithm

- Bridges connected to the same LAN exchange messages to configure the bridge (Configuration Bridge Protocol Data Unit, Configuration BPDUs) to build the tree.

With the help of the BPDUs, bridges can:

- Elect a single bridge as the **root bridge**.
- Calculate the distance of the shortest path to the root bridge
- Each LAN can determine a **designated bridge**, which is the bridge closest to the root. The designated bridge will forward packets towards the root bridge.
- Each bridge can determine a **root port**, the port that gives the best path to the root.
- Select ports to be included in the spanning tree.

Configuration BPDUs



© Jörg Liebeherr 1998-2000

21

Concepts

- Each bridge as a unique identifier:

$$\text{Bridge ID} = \langle \text{MAC address} + \text{priority level} \rangle$$

Note that a bridge has several MAC addresses (one for each port), but only one ID
- Each port within a bridge has a unique identifier (port ID).
- **Root Bridge:** The bridge with the lowest identifier is the root of the spanning tree.
- **Root Port:** Each bridge has a root port which identifies the next hop from a bridge to the root.

© Jörg Liebeherr 1998-2000

22

Concepts

- **Root Path Cost:** For each bridge, the cost of the min-cost path to the root.
Assume it is measured in #hops to the root
- **Designated Bridge, Designated Port:** Single bridge on a LAN that provides the minimal cost path to the root for this LAN:
 - if two bridges have the same cost, select the one with highest priority
 - if the min-cost bridge has two or more ports on the LAN, select the port with the lowest identifier
- **Note:** We assume that “cost” of a path is the number of “hops”.

© Jörg Liebeherr 1998-2000

23

Steps of Spanning Tree Algorithm

1. Determine the root bridge
 2. Determine the root port on all other bridges
 3. Determine the designated port on each LAN
- Each bridge is sending out BPDUs that contain the following information:

root ID	cost	bridge ID/port ID
---------	------	-------------------

root bridge (what the sender thinks it is)
root path cost for sending bridge
Identifies sending bridge

© Jörg Liebeherr 1998-2000

24

Ordering of Messages

- We can order BPDU messages with the following ordering relation “ \ll ”:

M1

ID R1	C1	ID B1
-------	----	-------

 <

ID R2	C2	ID B2
-------	----	-------

M2

If $(R1 < R2)$

M1 \ll M2

elseif $((R1 == R2) \text{ and } (C1 < C2))$

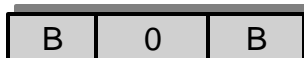
M1 \ll M2

elseif $((R1 == R2) \text{ and } (C1 == C2) \text{ and } (B1 < B2))$

M1 \ll M2

Determine the Root Bridge

- Initially, all bridges assume they are the root bridge.
- Each bridge B sends BPDUs of this form on its LANs:



- Each bridge looks at the BPDUs received on all its ports and its own transmitted BPDUs.
- Root bridge is the smallest received root ID that has been received so far (Whenever a smaller ID arrives, the root is updated)

Calculate the Root Path Cost Determine the Root Port

- At this time: A bridge B has a belief of who the root is, say R.
- Bridge B determines the Root Path Cost (Cost) as follows:
 - If $B = R$: Cost = 0.
 - If $B \neq R$: Cost = {Smallest Cost in any of BPDUs that contains root ID R} + 1
- **B's root port** is the port from which B received the lowest cost path to R (in terms of relation " $<<$ ").
- Knowing R and Cost, B can generate its BPDU (but will not necessarily send it out):



© Jörg Liebeherr 1998-2000

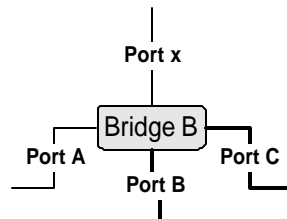
27

Calculate the Designated Bridge Determine the Designated Port

- At this time: B has generated its BPDU



- B will send this BPDU on one of its ports, say **port x**, only if its BPDU is lower (via relation " $<<$ ") than any BPDU that B received from port x.
- In this case, B also assumes that it is the **designated bridge** for the LAN to which the port connects.



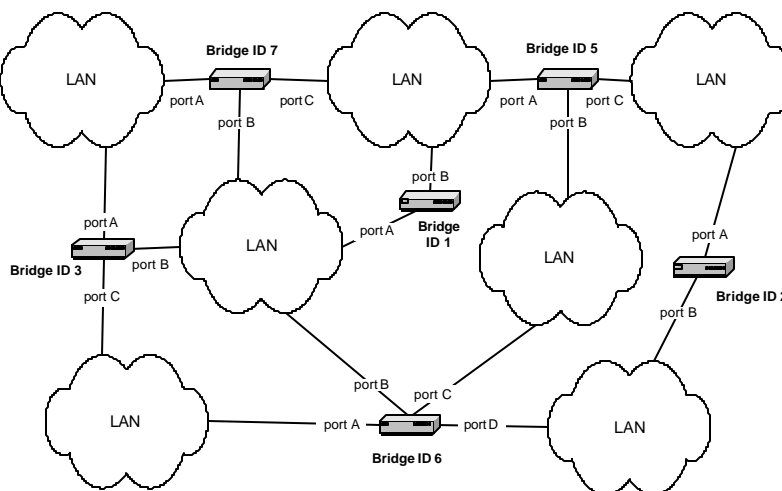
© Jörg Liebeherr 1998-2000

28

Selecting the Ports for the Spanning Tree

- At this time: Bridge B has calculated the root, the root path cost, and the designated bridge for each LAN.
- Now **B can decide which ports are in the spanning tree**:
 - B's root port is part of the spanning tree
 - All ports for which B is the designated bridge are part of the spanning tree.
- B's ports that are in the spanning tree will forward packets **(=forwarding state)**
- B's ports that are not in the spanning tree will not forward packets **(=blocking state)**

Example network



Example

- Assume that all bridges send out their BPDU's once per second, and assume that all bridges send their BPDUs at the same time. Assume that all bridges are turned on simultaneously at time T=0 sec.
- Use the following table to write the BPDUs which are sent by the bridges. For each cell, include:
 - the BPDU (For each BPDU, only provide the <root, cost, bridge> fields)
 - the ports on which the BPDUs is transmitted (write "none" if the BPDU is not sent on any of the ports)
- See next page.

Table for example

	Bridge 1	Bridge 2	Bridge 3	Bridge 5	Bridge 6	Bridge 7
T=0sec	1,0,1	2,0,2	3,0,3	5,0,5	6,0,6	7,0,7
T=1sec	1,0,1	2,0,2	1,1,3	1,1,5	1,1,6	1,1,7
T=2sec	1,0,1	1,2,2	1,1,3	1,1,5	1,1,6	1,1,7

Results for example

	Bridge 1	Bridge 2	Bridge 3	Bridge 5	Bridge 6	Bridge 7
Root Port	/	A	B	A	B	B
Designated Ports	A,B	/	A,C	B,C	D	/
Blocked ports	/	B	/	/	A,C	A,C