

Transparent Bridging and VLAN

Plug and Play Networking

(C) Herbert Haas 2005/03/11

Algorhyme



*I think that I shall never see
a graph more lovely than a tree
a graph whose crucial property
is loop-free connectivity.
A tree which must be sure to span
so packets can reach every lan.
first the root must be selected
by ID it is elected.
least cost paths to root are traced,
and in the tree these paths are place.
mesh is made by folks like me;
bridges find a spanning tree.*

Radia Perlman

Radia Perlman, PhD computer science 1988, MIT * MS math 1976, MIT * BA math 1973, MIT

Radia Perlman specializes in network and security protocols. She is the inventor of the spanning tree algorithm used by bridges, and the mechanisms that make modern link state protocols efficient and robust. She is the author of two textbooks, and has a PhD from MIT in computer science.

Her thesis on routing in the presence of malicious failures remains the most important work in routing security. She has made contributions in diverse areas such as, in network security, credentials download, strong password protocols, analysis and redesign of IPsec's IKE protocols, PKI models, efficient certificate revocation, and distributed authorization. In routing, her contributions include making link state protocols robust and scalable, simplifying the IP multicast model, and routing with policies.

Bridge History



- Bridges came **after** routers!
- First bridge designed by **Radia Perlman**
 - ◆ Ethernet has size limitations
 - ◆ Routers were single protocol and expensive
- **Spanning Tree because Ethernet had no hop count**
- **IEEE 802.1D**

Bridging is a fundamental part of the IEEE LAN standard. Actually bridges were invented relatively late—routers were invented a bit earlier. Radia Perlman, a pioneer in data communication designed the first bridge. The main reason was to extend the total network diameter of Ethernet and to provide a transport technique which supports multiple layer 3 technologies. She also invented the Spanning Tree Protocol (STP) because Ethernet had no hop count, thus any store and forwarding technology would suffer from broadcast storms, when broadcast destination addresses are used. But this issue is discussed in more detail later in this chapter.

The IEEE standard 802.1D specifies bridging and spanning tree (and more).

What is Bridging?



- **Layer 2 packet forwarding principle**
- **Separate two (or more) shared-media LAN segments with a bridge**
 - ◆ Only frames destined to the other LAN segment are forwarded
 - ◆ **Number of collisions reduced (!)**
- **Different bridging principles**
 - ◆ **Ethernet: Transparent Bridging**
 - ◆ **Token Ring: Source Route Bridging**

Bridges forward layer 2 packets (frames) according to their destination address. Hereby, those frames are filtered whose destination is not reachable on another port of the bridge. This filtering capability significantly enhances the total performance of a LAN as it is divided into multiple segments—multiple broadcast domains: The number of collisions is reduced!

IEEE defined bridges for all kind of LAN technologies. For example a Token Ring network relies on so-called source route bridging, while Ethernet uses "Transparent Bridging".

This chapter only discusses Transparent Bridging.

Bridging vs Routing



- **Bridging works on OSI layer 2**
 - ♦ Forwarding of **frames**
 - ♦ Use **MAC** addresses only
 - ♦ Termination of physical layer (!)
- **Routing works on OSI layer 3**
 - ♦ Forwarding of **packets**
 - ♦ Use **routable** addresses only (e.g. IP)
 - ♦ Termination of both layer 1 and 2

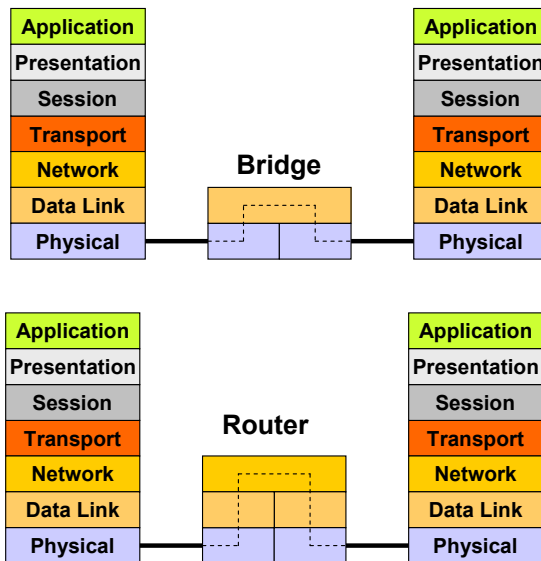
There are many differences between bridging and routing! The only thing in common is the store and forwarding principle, based on some sort of destination address.

But a bridge forwards layer 2 frames while a router forwards layer 3 packets. Layer 2 frames use simple MAC addresses, having no logical structure, while layer 3 packets use structured addresses, revealing topology information. Only layer 3 addresses are routable. In order to understand the latter statement, it is important to understand the principles of routing and how a routing table works. We will discuss this soon.

Bridges terminate physical links. Thus, one port of the same bridge might support optical fiber transmission and another port might support twisted pair copper cabling.

On the other hand, routers terminate layer 2 links. That is, one interface might utilize Ethernet as link layer technology, another interface Frame Relay, and a third interface might run ATM. A router only forwards the packet—the layer 3 information—carried inside a frame.

OSI Comparison



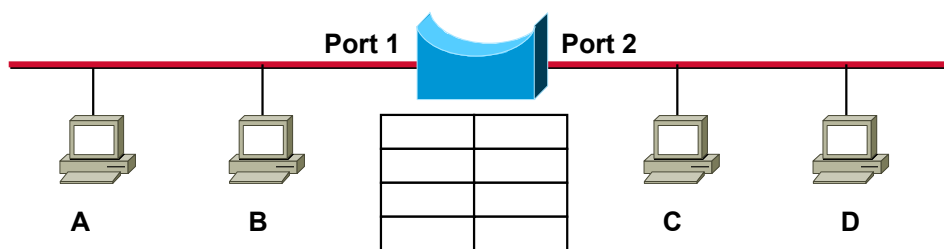
- **MAC addresses not routable**
 - ♦ NetBios over NetBEUI not routable (no L3)
- **Bridge supports different physical media on each port**
 - ♦ E.g. 10Mbit/s to 100Mbit/s
- **Router supports different layer-2 technologies**
 - ♦ E.g. Ethernet to Frame Relay

It is very important to understand the differences between bridges and routers. There are many implications related to the operating layer these devices support. As a rule of thumb any device is able to terminate all layers below the highest layer implemented.

How does it work?



- Transparent bridging is like "plug & play"
- Upon startup a bridge knows nothing
- Bridge is in **learning mode**



(C) Herbert Haas 2005/03/11

7

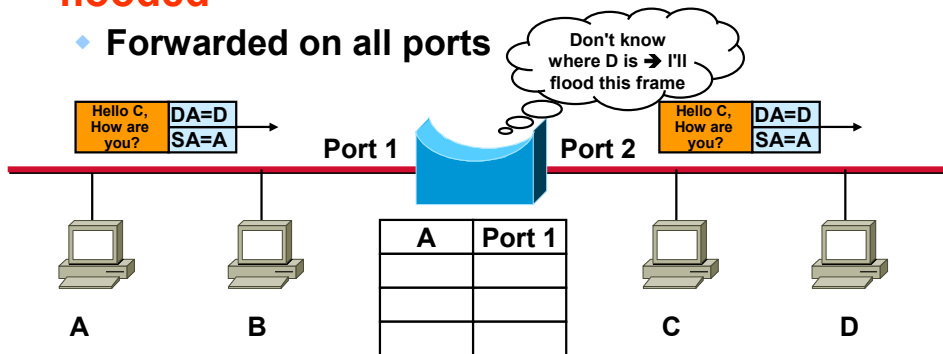
The main advantage of transparent bridging over source route bridging (token ring) is the transparency or "plug & play" capability. No end station notices the presence of bridges.

In order to be invisible, bridges must learn somehow where end stations are located. Upon startup, a bridge knows nothing and the bridging table is empty. At this time the bridge is in learning mode.

Learning



- Once stations send frames the bridge notices the **source** MAC address
 - ♦ Entered in bridging table
- Frames for unknown destinations are **flooded**
 - ♦ Forwarded on all ports



(C) Herbert Haas 2005/03/11

8

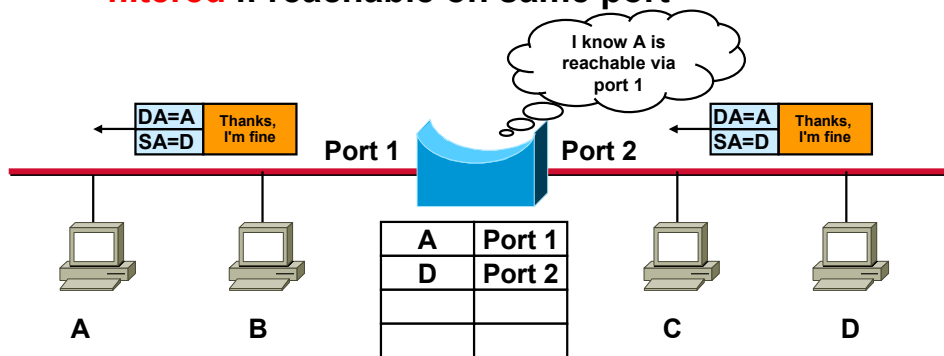
Assume we have a bridge with only two ports, each attached at one Ethernet segment. Assume the left station "A" sends one frame to "D" on the right side. Obviously the bridge learns the location of A but has no idea where D is. Thus the MAC address of A is entered in the bridging table and also the port number "1", on which A is reachable. Since the location of D is unknown, the bridge floods this frame over all ports, in our case only to port two (as there are no other ports).

This way, connectivity is granted even if there is no entry in the bridging table.

Learning → Table Filling



- If the destination address matches a bridging table entry, this frame can be actively
 - ♦ **forwarded** if reachable via other port
 - ♦ **filtered** if reachable on same port



(C) Herbert Haas 2005/03/11

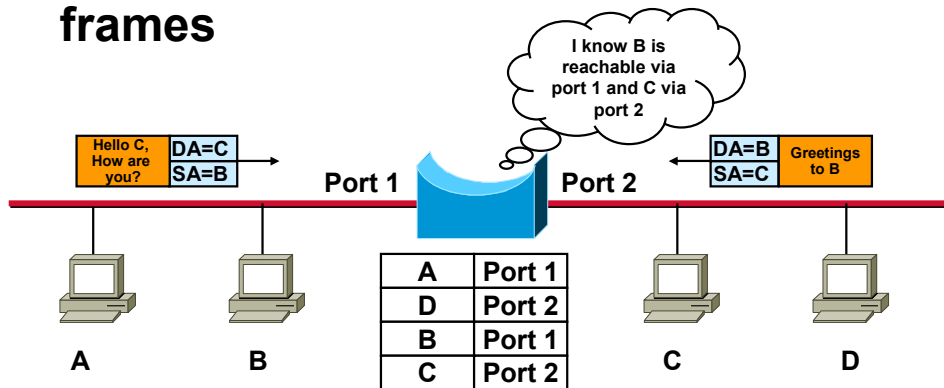
9

Now assume D replies to the message which has been received from A. The bridge knows already the port number over which A can be reached and forwards the frame accordingly. If A would be located on the same port as D then this frame would be filtered.

Learning → Table Filling



- After some time the location of every station is known – simply by listening!
- Now only **forwarding** and **filtering** of frames



(C) Herbert Haas 2005/03/11

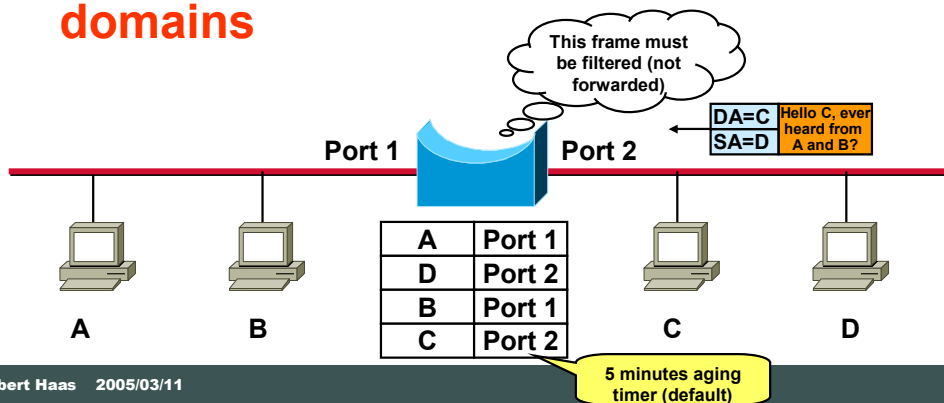
10

After some traffic observing time, the bridging table contains all host locations (addresses and port numbers). At this time the bridge enters the forwarding and filtering mode.

Forwarding and Filtering



- Frames whose source and destination address are reachable over the same bridge port are filtered
- LAN separated into **two collision domains**



Since only frames are forwarded to other ports whose destination is really located there, the LAN is separated into as many collision domains as ports are available (and attached to a LAN segment).

What if a host is removed from its location and attached at another place in the LAN? Obviously frames could be forwarded to the wrong port. Therefore each entry in the bridging table ages out after some time. The default aging time is 300 seconds or 5 minutes.

Most Important !



- Bridge separates LAN into **multiple collision domains** !
- A bridged network is still **one broadcast domain** !
 - ◆ Broadcast frames are always flooded
- A **router** separates the whole LAN into **multiple broadcast domains**

It is very important to understand the basic message which is given here:

The use of bridges results in a separation of **multiple collision domains** of the LAN. Still we have **one single broadcast domain!** That is, broadcast frames are always flooded throughout the network.

Only the use of routers results in a separation of multiple (layer 2) broadcast domains—or the use of VLANs, which will be discussed soon in this chapter.

What is a Switch?



- A switch *is* basically a bridge, differences are only:
 - ♦ **Faster** because implemented in **HW**
 - ♦ **Multiple ports**
 - ♦ **Improved functionality**
- **Don't confuse it with WAN Switching!**
 - ♦ **Completely different !**
 - ♦ **Connection oriented (stateful) VCs**



Now what is the difference between a bridge and a switch? Logically there is no difference. Technically there are major differences, leading marketing folks to define a new term—the switch. Switches typically employ more than two ports, and the bridging functionality is implemented in hardware. Additionally other features are added, depending on the vendor. These will be discussed next.

Note: Don't confuse LAN switching with WAN switching. Unfortunately modern bridging is called switching but logically it is still bridging. The term "bridging" was originally defined to differentiate this technique strictly from WAN switching. The main characteristic of WAN switching is its connection oriented behavior—WAN switches are never transparent! In order to connect to a WAN switch the end system must comply to some specific User to Network Interface (UNI).



Bridge = Switch

Since we use only switches today, let's talk about them...

Modern Switching Features



- **Different data rates supported simultaneously**
 - ◆ 10, 100, 1000, 10000 Mbit/s depending on switch
- **Full duplex operation**
- **QoS**
 - ◆ Queuing mechanisms
 - ◆ Flow control
- **Security features**
 - ◆ Restricted static mappings (DA associated with source port)
 - ◆ Port secure (Limited number of predefined users per port)
- **Different forwarding**
 - ◆ Store & Forward
 - ◆ Cut-through
 - ◆ Fragment-Free
- **VLAN support (Trunking)**
- **Spanning Tree**

Today most switches support different data rates at each interface or at selected interfaces. Also full duplex operation is standard today. QoS might be supported by using sophisticated queuing techniques, 802.1p priority tags, and flow control features, such as the pause MAC control frame.

Security is provided by statically entered switching tables and port locking (port secure), that is only a limited number or predefined users are allowed at some designated ports.

Forwarding of frames can be significantly enhanced using cut through switching: the processor immediately forwards the frame when the destination is determined. The switching latency is constant and very short for all length of packets but the CRC is not checked. In the Fragment-Free switching mode, the switch waits for the collision window (64 bytes) to pass before forwarding. If a packet has an error or better explained, a collision, it almost always occurs within the first 64 bytes. Fragment-Free mode provides better error checking than the Cut through mode with practically no increase in latency. The store and forward mode is the classical forwarding mode.

VLAN support allows to separate the whole LAN into multiple broadcast domains, hereby improving performance and security.

The spanning tree protocol (STP) avoids broadcast storms in a LAN. It is described on the next slides.

Bridging Problems



- **Redundant paths lead to**
 - ♦ Broadcast storms
 - ♦ Endless cycling
 - ♦ Continuous table rewriting
- **No load sharing possible**
- **No ability to select best path**
- **Frame may be stored for 4 seconds (!)**
 - ♦ Although rare cases
 - ♦ But only little acceptance for realtime and isochronous traffic – might change!

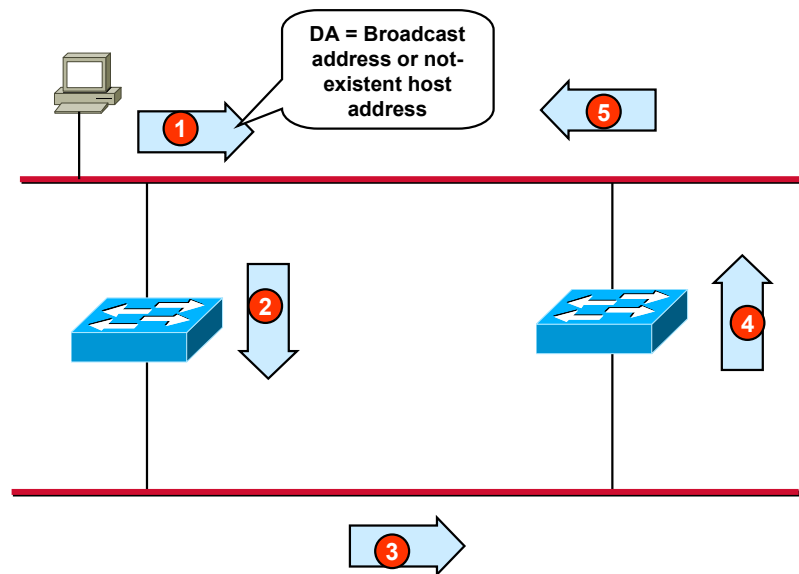
You might have noticed that bridges do not really learn the network topology. They only learn a simple destination to port association! Because of this there is no means to determine the best path, and furthermore frames might be caught in a loop.

Especially broadcast frames have no defined destination and would be forwarded over all parallel paths—endlessly! This results in endless circling of frames, or more dangerous, in a so-called "broadcast storm".

Also a continuous table rewriting might occur (this is not so widely known but also explained in the next pages).

Most people are not aware that frames might be stored up to 4 seconds inside the buffer of a switch—and it still complies to the IEEE standard. Although this would happen only in rare cases of congestion, transparent bridging is not suitable for hard realtime applications. Today the situation has changed, QoS features are included to assure bounded delays.

Endless Circling



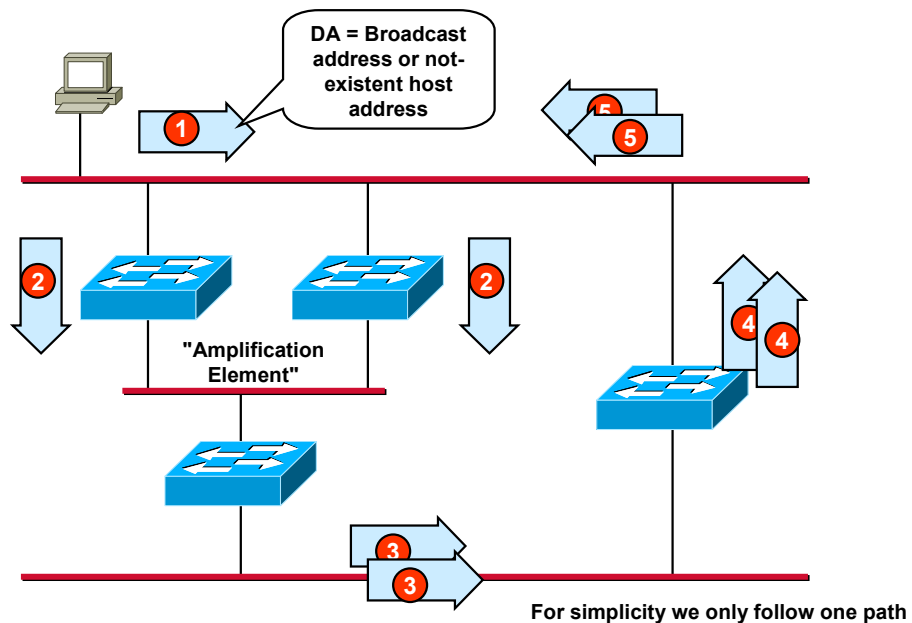
For simplicity we only follow one path

The picture above illustrates the endless circling phenomena. Assume a network with parallel paths between two LAN segments, realized by two bridges. Any frame with a broadcast destination address would be forwarded by both bridges to the other segment and back and forth and so on.

Obviously endless circling leads to congestion problems and is not desired. Remember that there is not hop count or time-to-live number within the Ethernet header.

But endless circling is not the main problem... (see next slide)

Broadcast Storm (1)



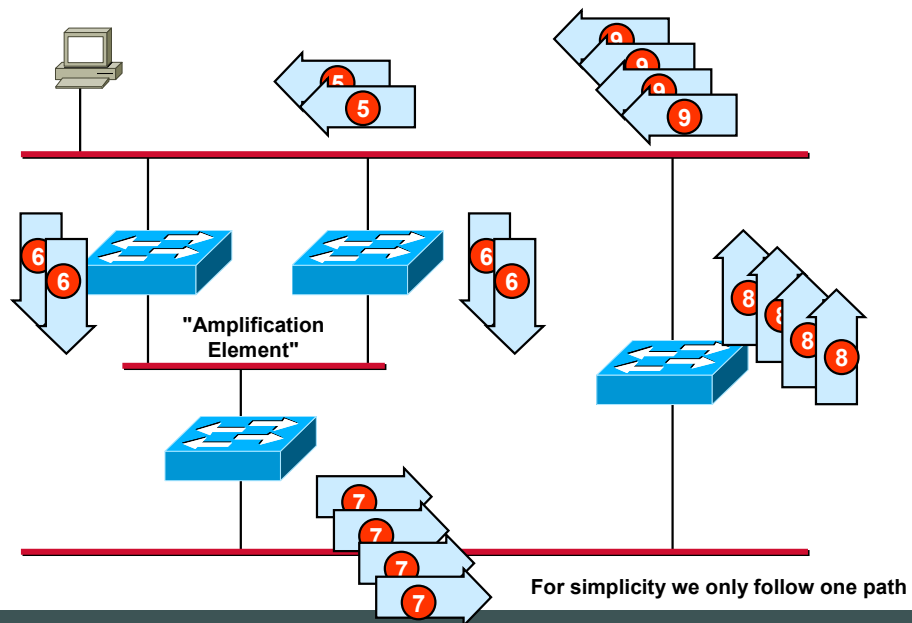
(C) Herbert Haas 2005/03/11

18

The most feared issue with bridging are broadcast storms. Broadcast storms can be considered as a dramatically "enhanced" endless circling problem. Broadcast storms appear when there is an "amplification" element within the network, such as those threefold parallel paths in the diagram above.

Within a very short time (e.g. 1 second) the whole LAN is overloaded with broadcast frames and nobody could transmit any useful frame anymore.

Broadcast Storm (2)

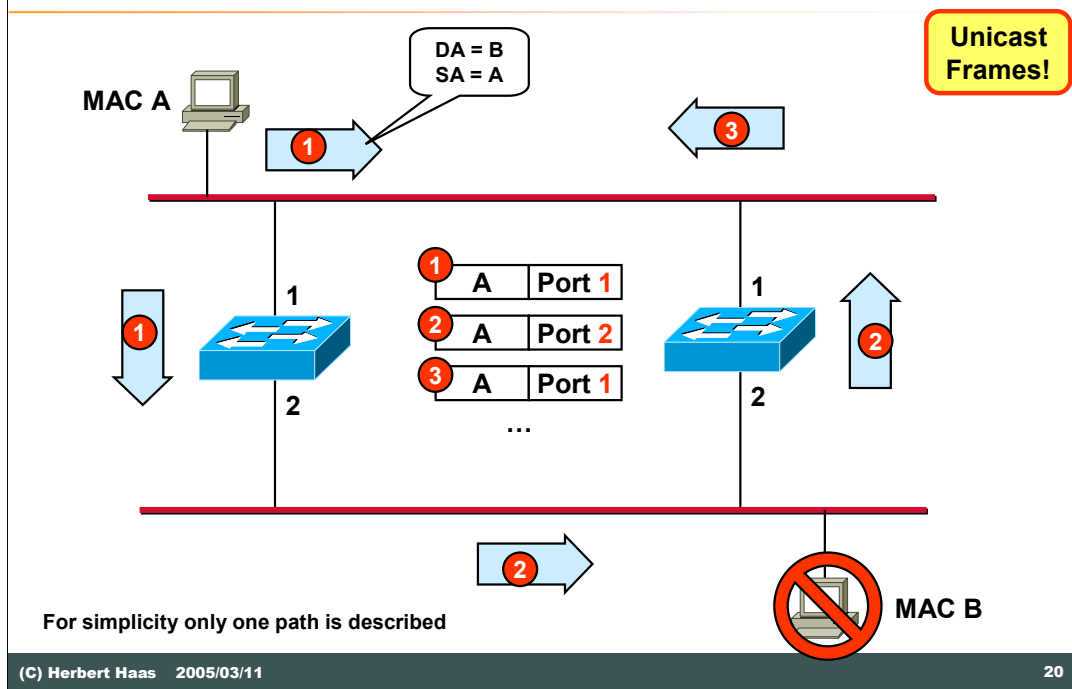


(C) Herbert Haas 2005/03/11

19

The picture above shows the amplification effect mentioned on the previous page.

Mutual Table Rewriting



A relatively seldom known problem is the mutual table rewriting phenomena.

This problem occurs with unicast frames!

Assume that host A sends an unicast frame to destination B, both bridges learn the location of host A and host B, but suddenly B is detached. However, both bridges keep the entry for B for five minutes.

During this time the following happens:

- 1) After the bridges forward the frame from the above segment to the bottom segment this frame is not consumed by any host B, and therefore the bridges forward this frame back to the top segment.
- 2) At this moment the bridges rewrites their table as host A appears to be located on the bottom segment.
- 3) Again the bridge forward the frame to the bottom segment, hereby rewriting the port address for this source address...ad infinitum!

Spanning Tree



- Invented by *Radia Perlman* as general "mesh-to-tree" algorithm
- A must in bridged networks with redundant paths
- Only one purpose:
cut off redundant paths with highest costs

Now we have learned that active parallel paths lead to severe problems in a switched (i.e. bridged) network. Therefore we can only overcome this problem by deactivating any redundant path. This should be performed automatically in order to call Ethernet bridging still "Transparent" bridging.

The inventor of bridging, Radia Perlman, also created an easy solution for the redundancy problem: The Spanning Tree Protocol (STP).

The STP is implemented in bridges only (not in hosts) and has only one purpose: To determine any redundant paths and cut them off! Hereby cost values are considered for each path in order to maintain the best paths.

STP Ingredients



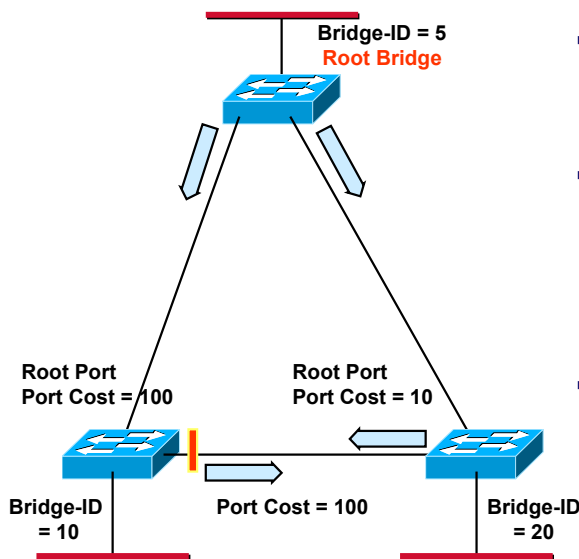
- **Special STP frames: "Bridge Protocol Data Units" (BPDUs)**
- **A Bridge-ID for each bridge**
 - ◆ Priority value (16 bit, default 32768)
 - ◆ (Lowest) MAC address
 - ◆ Cisco uses only 4 most-significant bits
 - Remaining 12 bits used as "Extended System-ID" (actually the VLAN number)
- **A Port Cost for each port**
 - ◆ Initially 1000/Mbits now *nonlinear* table
 - 10 Mbit/s → C=100
 - 100 Mbit/s → C=19

What do we need for STP to work? First of all this protocol needs a special messaging means, realized in so-called **Bridge Protocol Data Units (BPDUs)**. BPDUs are simple messages contained in Ethernet frames containing several parameters described below.

Each bridge is assigned one unique **Bridge-ID** which is a combination of a 16 bit priority number and the lowest MAC address found on any port on this bridge. The Bridge-ID is determined automatically using the default priority 32768.

Each port is assigned a **Port Cost**. Again this value is determined automatically using the simple formula $\text{Port Cost} = 1000 / \text{BW}$, where BW is the bandwidth in Mbit/s. Of course the Port Cost can be configured manually.

STP Principle



- First a **Root Bridge** is determined
 - ♦ Initially every bridge assumes itself as root
 - ♦ The bridge with lowest Bridge-ID wins
- Then the root bridge triggers BDPU sending (hello time intervals)
 - ♦ Received at "**Root Ports**" by other bridges
 - ♦ Every bridge adds its own port cost to the advertised cost and forwards the BPDU
- On each LAN segment one bridge becomes **Designated Bridge**
 - ♦ Having lowest total root path cost
 - ♦ Other bridges set redundant ports in **blocking state**

(C) Herbert Haas 2005/03/11

23

We give only a basic explanation here of how the STP works. First a **Root Bridge** is determined by choosing the bridge with the **lowest** Bridge-ID. This is simply done by sending BDUs containing the presumed Root Bridge. At first each bridge assumes to be the Root Bridge itself. After any bridge has sent his "opinion" the root bridge is determined.

Then the **Root Ports** are determined by each bridge. The Root Bridge sends BPDUs periodically (every 2 seconds by default) "downstream" to the "leaves" of the tree which is currently created. Each bridge adds its own port costs to the Root Path Cost parameter in the BPDU and forwards this BPDU over all other ports. This way each bridge learns the best path to the root.

Finally on each LAN segment the bridge having best Root Port becomes **Designated Bridge**. Its port on this LAN segment is called Designated Port (DP). Root Ports and Designated Ports are in a forwarding state. All other ports are in a blocking state.

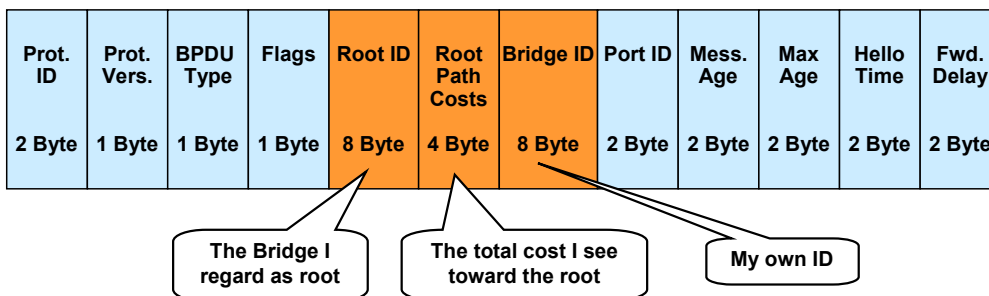
But the best (and shortest) description comes from Radia Perlman's poem:

*First the root must be selected
by ID it is elected.
least cost paths to root are traced,
and in the tree these paths are place.*

BPDU Format



- Each bridge sends periodically BPDUs carried in Ethernet multicast frames
 - ◆ Hello time default: 2 seconds
- Contains all information necessary for building Spanning Tree



Just for your interest, the above picture shows the structure of BPDUs. You see, there is no magic in here, and the protocol is very simple. There are no complicated protocol procedures. BPDUs are sent periodically and contain all involved parameters. Each bridge enters its own "opinion" there or adds its root path costs to the appropriate field. Note that some parameters are transient and others are not.

The other parameters not explained here are not so important to understand the basic principle.

Note



- **Redundant links remain in active stand-by mode**
 - ◆ If root port fails, other root port becomes active
- **Low-price switches might not support STP**
 - ◆ Don't use them in meshed configurations
- **Only 7 bridges per path allowed according standard (!)**
 - ◆ Use more recent Rapid STP instead...

Still it is reasonable to establish parallel paths in a switched network in order to utilize this redundancy in an event of failure. The STP automatically activates redundant paths if the active path is broken. Note that BPDUs are always sent or received on blocking ports.

Note that (very-) low price switches might not support the STP and should not be used in high performance and redundant configurations.

For performance reasons the IEEE standard 802.1d only allows 7 bridges for each path. Some vendors allow to change this value.

Only for your interest, here are the Ethernet parameters for BPDUs:

Multicast address 0180 C200 0000 hex

LLC DSAP=SSAP= 42 hex

STP More Detailed...



Typical loop problems:

- Broadcast Storm
- Bridge Table Corruption, endless unicast circulation

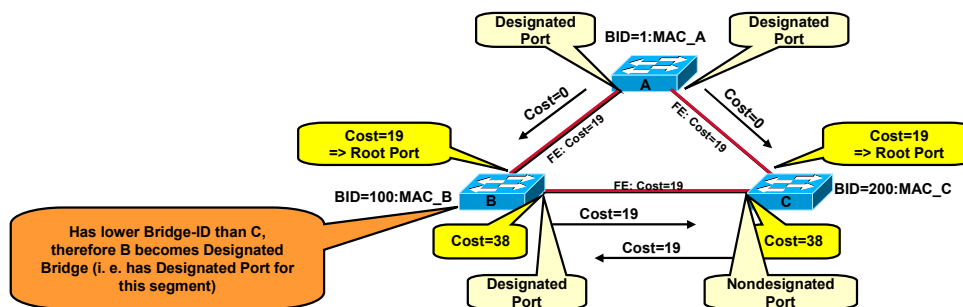
Three steps to create spanning tree:

1. Elect Root Bridge (Each L2-network has exactly one Root Bridge)
2. Elect Root Ports (Each non-root bridge has exactly one Root Port)
3. Elect Designated Ports (Each segment has exactly one Designated Port)

The "better" BPDU contains

1. Lowest Bridge ID
2. Lowest Path Cost to Root Bridge
3. Lowest Sender Bridge ID
4. Lowest Port ID

} Only used as tie breaker to determine Designated Port



(C) Herbert Haas 2005/03/11

26

Each segment has exactly one Designated Port. This simple rule actually breaks any loops.

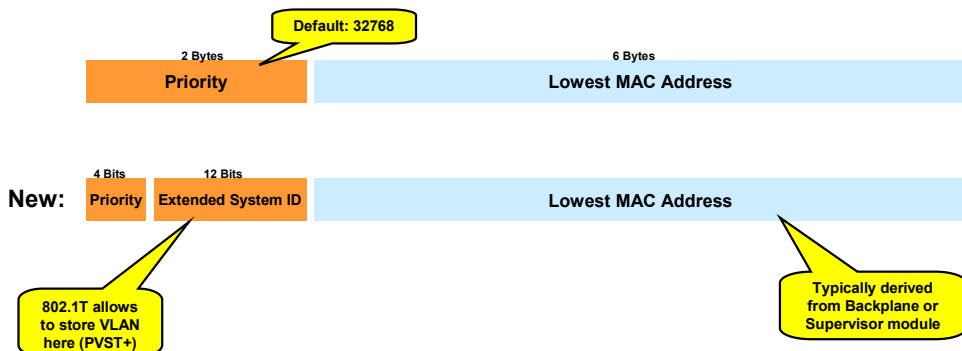
A nondesignated port receives a more useful BPDU than the one it would send out on its segment. Therefore it remains in the so-called blocking state.

Port ID - Contains a unique value for every port. Port 1/1 contains the value 0x8001, whereas Port 1/2 contains 0x8002. (Or in decimal: 128.1, 128.2, ...)

From the 802.1D-1998 standard:

Each Configuration BPDU contains, among other parameters, the unique identifier of the Bridge that the transmitting Bridge believes to be the Root, the cost of the path to the Root from the transmitting Port, the identifier of the transmitting Bridge, and the identifier of the transmitting Port. This information is sufficient to allow a receiving Bridge to determine whether the transmitting Port has a better claim to be the Designated Port on the LAN on which the Configuration BPDU was received than the Port currently believed to be the Designated Port, and to determine whether the receiving Port should become the Root Port for the Bridge if it is not already.

Parameter Details: Bridge-ID



- **The recent 802.1D-2004 standard describes how the modern RSTP is backwards compatible to the STP**
 - ◆ Now the configurable priority part is only 4 bits

802.1T spanning-tree extensions, and some of the bits previously used for the switch priority are now used for the extended system ID (VLAN identifier for the per-VLAN spanning-tree plus [PVST+] and for rapid PVST+ or an instance identifier for the multiple spanning tree [MST]).

Before this, spanning tree used one MAC address per VLAN to make the bridge ID unique for each VLAN.

Extended system IDs are VLAN IDs between 1025 and 4096. Releases 12.1(14)E1 and later releases support a 12-bit extended system ID field as part of the bridge ID.

```
Switch(config)# spanning-tree extend system-id
```

STP Path Cost



Speed [Mbit/s]	Old Cost (1000/Speed)	New Cost
10	100	100
100	10	19
155	6	14
622	1	6
1000	1	4
10000	1	2

- **Also different cost values might be used**
 - ♦ See recommendations in the IEEE 802.1D-2004 standard to comply with RSTP and MSTP

BPDU Format



		Bytes	
<p>A TCN-BPDU only consists of these 3 fields !!!</p>	Protocol ID	2	Always zero
	Version	1	Always zero
	Message Type	1	Configuration (0x00) or TCN BPDU (0x80)
<p>When first booted, Root-ID == BID</p>	Flags	1	LSB = Topology change flag (TC), MSB = TC Ack flag (TCA)
	Root ID	8	Who is Root Bridge?
<p>If value increases, then the originating bridge lost connectivity to Root Bridge</p>	Root Path Cost	4	How far away is Root Bridge?
	Bridge ID	8	ID of bridge that sent this BPDU
	Port ID	2	Port-ID of sending bridge (unique: Port1/1=0x8001, 1/2=0x8002, ...)
<p>• Predetermined by root bridge • Affect convergence time • Misconfigurations cause loops</p>	Message Age	2	Time since Root generated this BPDU
	Maximum Age = 20	2	BPDU is discarded if older than this value (default: 20 seconds)
	Hello Time = 2	2	Broadcast interval of BPDUs (default: 2 seconds)
	Forward Delay = 15	2	Time spent in learning and listening states (default: 15 seconds)

- BPDUs are sent in 802.3 frames
 - ♦ DA = 01-80-C2-00-00-00
 - ♦ LLC has DSAP=SSAP = 0x42
- Configuration BPDUs
 - ♦ Originated by Root Bridge periodically (2 sec Hello Time), flow downstream

In normal stable operation, the regular transmission of Configuration Messages by the Root ensures that topology information is not timed out. To allow for reconfiguration of the Bridged LAN when components are removed or when management changes are made to parameters determining the topology, the topology information propagated throughout the Bridged LAN has a limited lifetime. This is effected by transmitting the age of the information conveyed (the time elapsed since the Configuration Message originated from the Root) in each Configuration BPDU. Every Bridge stores the information from the Designated Port on each of the LANs to which its Ports are connected, and monitors the age of that information.

Protocol Details



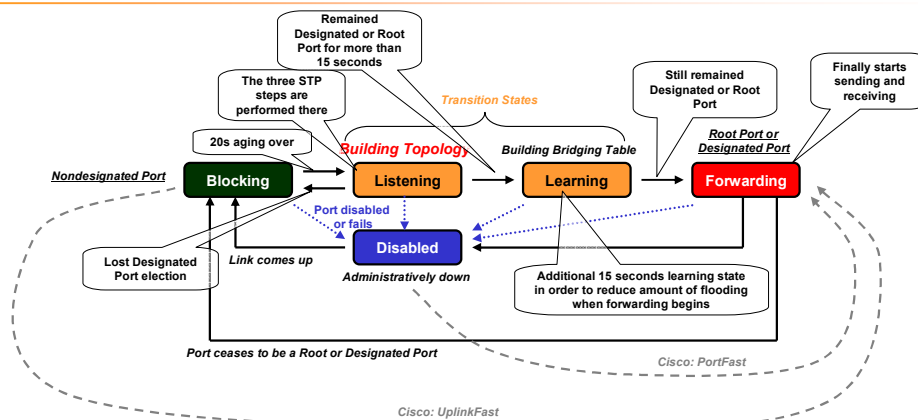
- BPDUs are sent in 802.3 frames
 - ♦ DA=01-80-C2-00-00-00
 - ♦ LLC has DSAP=SSAP=0x42
- Configuration BPDUs
 - ♦ Originated by Root Bridge periodically (2 sec Hello Time), **flow downstream**
- Topology Change Notification (TCN) BPDUs
 - ♦ Used as alert by any bridge, **flow upstream** (through Root Port)
 - ♦ Only consists of first three fields!
 - ♦ Sent upon transition of a port into Forwarding state and at least one Designated Port exists
 - ♦ Sent upon transition of a port into Blocking state (from either Forwarding or Learning state)
 - ♦ **Sent until acknowledged by TC Acknowledge (TCA)**
 - ♦ Only the Designated Ports of upstream bridges processes TCN-BPDUs and sent TCA downstream
 - ♦ Finally Root Bridge receive it and sends Configuration BPDU with TC=TCA=1 downstream for (Forward Delay + Max Age = 35) seconds => instructs all bridges to reduce default bridging table aging (300s) to the current Forward Delay value (15s)

Protocol Details



- **Root switch has only Designated Ports (=forwarding)**
- **Other switches have**
 - ♦ Exactly one Root Port (upstream)
 - ♦ Zero or more Designated Ports (downstream)
 - ♦ Zero or more Nondesignated Ports (blocked)
- **Additionally, each of these port is in one of the following states**
 - ♦ Blocking, Listening, Learning, Forwarding, or Disabled

Port Transition Rules



- **STP is completely performed in the Listening state**
 - Blocking ports still receive BPDUs (but don't send)
- **Default convergence time is 30-50 s**
 - 20s aging, (15+15)s transition time
- **Timer tuning: Better don't do it !**
 - Only modify timers of the root bridge
 - Don't forget values on supposed backup root bridge

802.1d defines port **roles** and **states**:

Port Roles	Port States
Root	Disabled
Designated	Blocking
Nondesignated	Listening
	Learning
	Forwarding

A specific port role is a long-term "destiny" for a port, while port states denote transient situations. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

From the 802.1D-1998 standard:

If the Bridge times out the information held for a Port, it will attempt to become the Designated Bridge for the LAN to which that Port is attached, and will transmit protocol information received from the Root on its Root Port on to that LAN.

If the Root Port of the Bridge is timed out, then another Port may be selected as the Root Port. The information transmitted on LANs for which the Bridge is the Designated Bridge will then be calculated on the basis of information received on the new Root Port.

Configuration



```
Switch(config)# spanning-tree vlan 200
```

Enable SPT on a specific VLAN

```
Switch(config)# spanning-tree vlan 200 priority 0
```

Enforcing Root Bridge

```
Switch(config-if)# spanning-tree cost 18
```

Manipulate Port Costs

```
Switch(config-if)# spanning-tree vlan 200 cost 15
```

Manipulate Port Costs for a specific VLAN

```
Switch# show spanning-tree vlan 200

VLAN0200
Spanning tree enabled protocol ieee
Root ID    Priority    49352
          Address    0008.2199.2bc0
          This bridge is the root
          Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority    49352 (priority 49152 sys-id-ext 200)
          Address    0008.2199.2bc0
          Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
          Aging Time 300
          Uplinkfast enabled

Interface  Port ID      Designated      Port ID
Name      Prio.Nbr    Cost Sts       Cost Bridge ID   Prio.Nbr
-----
Fa0/1     128.1       3019 LIS       0 49352 0008.2199.2bc0 128.1
Fa0/2     128.2       3019 LIS       0 49352 0008.2199.2bc0 128.2
```

Enable spanning tree on a per-VLAN basis.

Old commands:

set spantree priority

set spantree root

show spantree

AND THE ANSWER IS ... FORTY-TWO!

The choice of 0x42 as the LLC SAP value for BPDUs has an interesting history. First, the chair and editor of the IEEE 802.1D Task Force (Mick Seaman) was British, and 42 is "The Answer to the Ultimate Question of Life, the Universe, and Everything" in *The Hitchhiker's Guide to the Galaxy*, a popular British book, radio, and television series [by Douglas Adams] at the time of the development of the original standard.

Even in the United States, the series was so popular that the original Digital Equipment Corp. bridge architecture specification was titled eXtended LAN Interface Interconnect, or XLII, the Roman representation of 42.

From Rich Seifert's Switch Book

(C) Herbert Haas 2005/03/11

34

Rich Seifert also continues:

"Finally, 0x42 is a palindrome; it has the same binary pattern regardless of whether one transmits the most-significant bit first or the least-significant bit first—01000010. This eliminates any confusion regarding bit ordering of the field when transmitted on Little Endian (e. g. Ethernet) versus Big Endian (e. g. Token Ring) networks, although this side benefit was not recognized until after the value was assigned."

Bridging versus Routing



Bridging		Routing	
	Depends on MAC addresses only		Requires structured addresses (must be configured)
	Invisible for end-systems; transparent for higher layers		End system must know its default-router
	Must process every frame		Processes only frames addressed to it
	Number of table-entries = number of all devices in the whole network		Number of table-entries = number of subnets only
	Spanning Tree eliminates redundant lines; no load balance		Redundant lines and load balance possible
	No flow control		Flow control is possible (router is seen by end systems)

The list shown above summaries all pro and cons of bridging (switching) and routing.

Bridging versus Routing



Bridging

- No LAN/WAN coupling because of high traffic (broadcast domain!)
- Paths selected by STP may not match communication behaviour/needs of end systems
- + Faster, because implemented in HW; no address resolution
- + Location change of an end-system does not require updating any addresses
- Spanning tree necessary against endless circling of frames and broadcast storms

Routing

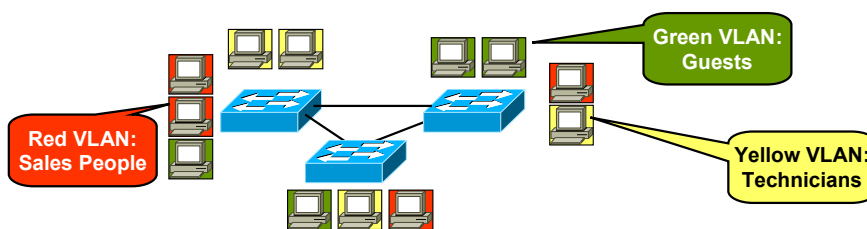
- + Does not stress WAN with subnet's broadcast or multicasts; commonly used as "gateway"
- + Router knows best way for each frame
- Slower, because usually implemented in SW; address resolution (ARP) necessary
- Location change of an end-system requires adjustment of layer 3 address
- Routing-protocols necessary to determine network topology

The list shown above summarizes all pro and cons of bridging (switching) and routing (continued from previous slide).

Virtual LANs



- **Separate LAN into multiple broadcast domains**
 - ◆ No global broadcasts anymore
 - ◆ For security reasons
- **Assign users to "VLANs"**



(C) Herbert Haas 2005/03/11

37

Since most organizations consist of multiple "working groups" it is reasonable to confine their produced traffic somehow. This is achieved using Virtual LANs (VLANs). Switches configured for VLANing consist logically of multiple virtual switches inside.

Users are assigned to dedicated VLANs and there is no communication possible between different VLANs—even broadcasts are blocked! This significantly enhances security.

On a switch each VLAN is identified by a number and a name (optionally) but in our example we also use colors to differentiate them.

Host to VLAN Assignment



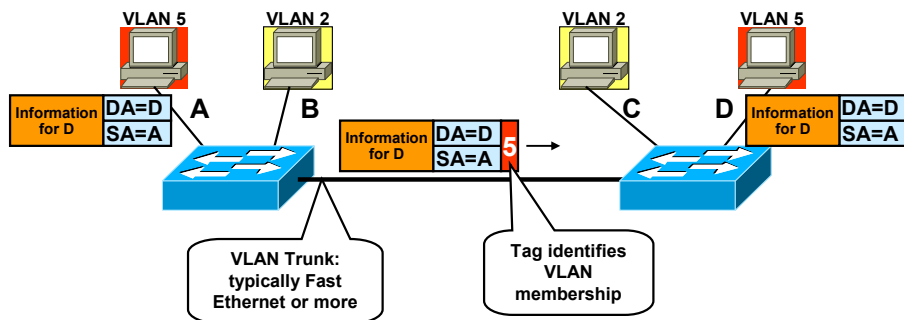
- **Different solutions**
 - ◆ **Port** based assignment
 - ◆ **Source address** assignment
 - ◆ Protocol based
 - ◆ Complex rule based
- **Bridges are interconnected via **VLAN trunks****
 - ◆ **IEEE 802.1q** (New: 802.1w, 802.1s)
 - ◆ **ISL (Cisco)**

There are different ways to assign hosts (users) to VLANs. The most common is the port-based assignment, meaning that each port has been configured to be member of a VLAN. Simply attach a host there and its user belongs to that VLAN specified.

Hosts can also be assigned to VLANs by their MAC address. Also special protocols can be assigned to dedicated VLANs, for example management traffic. Furthermore, some devices allow complex rules to be defined for VLAN assignment, for example a combination of address, protocol, etc.

Of course VLANs should span over several bridges. This is supported by special VLAN trunking protocols, which are only used on the trunk between two switches. Two important protocols are commonly used: the IEEE 802.1q protocol and the Cisco Inter-Switch Link (ISL) protocol. Both protocols basically attach a "tag" at each frame which is sent over the trunk.

VLAN Trunking Example



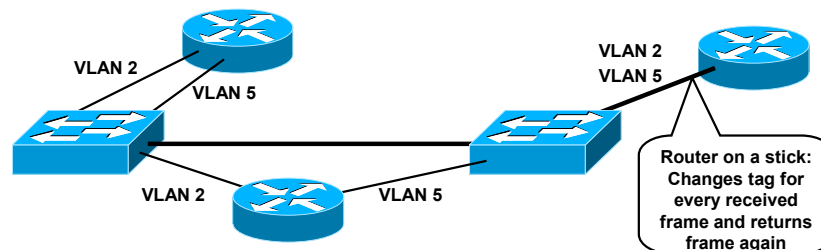
- Inter-VLAN communication not possible
- Packets across the VLAN trunk are **tagged**
 - ♦ Either using 802.1q or ISL tag
 - ♦ So next bridge is able to constrain frame to same VLAN as the source

By using VLAN tagging the "next" bridge knows whether the source address is also member of the same VLAN.

Inter-VLAN Traffic



- **Router can forward inter-VLAN traffic**
 - ♦ Terminates Ethernet links
 - ♦ Requirement: **Each VLAN in other IP subnet !**
- **Two possibilities**
 - ♦ Router is member of every VLAN with one link each
 - ♦ Router attached on VLAN trunk port ("Router on a stick")



(C) Herbert Haas 2005/03/11

40

Now we admit the wholly truth: of course it is possible to communicate between different VLANs—using a router! A router terminates layer 2 and is not interested in VLAN constraints. Of course this requires that each VLAN uses another subnet IP address since the router needs to make a routing decision.

There are two possible configurations: The straightforward solution is to attach a router to several ports on one or more switches, provided that each port is member of another VLAN.

Another method is the "Router on a stick" configuration, employing only a single attachment to a trunk port of a switch. This method saves ports (and cables) but requires trunking functionality on the router. Here the router simply changes the tag of each frame (after making a routing decision) and sends the frame back to the switch.

Summary



- **Ethernet Bridging is "Transparent Bridging"**
 - ♦ Hosts do not "see" bridges
 - ♦ Plug & Play
- **1 Collision domain → 1 Broadcast domain**
- **Switches increase network performance !**
- **Redundant paths are dangerous**
 - ♦ Broadcast storm is most feared
 - ♦ Solution: **Spanning Tree Protocol**
- **VLANs create separated broadcast domains**
 - ♦ Port based or address based VLANing
 - ♦ Routers allow inter-VLAN traffic

Quiz



- **Can I bridge from Ethernet to Token Ring?**
- **How is flow control implemented?**
- **Which bridge should be root bridge?**
- **What are main differences between 802.1q and ISL?**
- **What are Layer-3, Layer-4, and Layer-7 switches ?**

Hints



- **Q1: Yes, using translational bridges, problem with different MAC-address styles, increased delay due to higher processing demand, forget it!**
- **Q2: Half duplex: Backpressure (preamble jamming) or reduced interframe-gap; Full duplex: Pause frame (special MAC control frame)**
- **Q3: Root bridge should be point of high load**
- **Q4: 802.1q only allows one Spanning Tree for all VLANs, ISL allows multiple.**
- **Q5: HW-routers, QoS-Support, Application awareness (server load)**