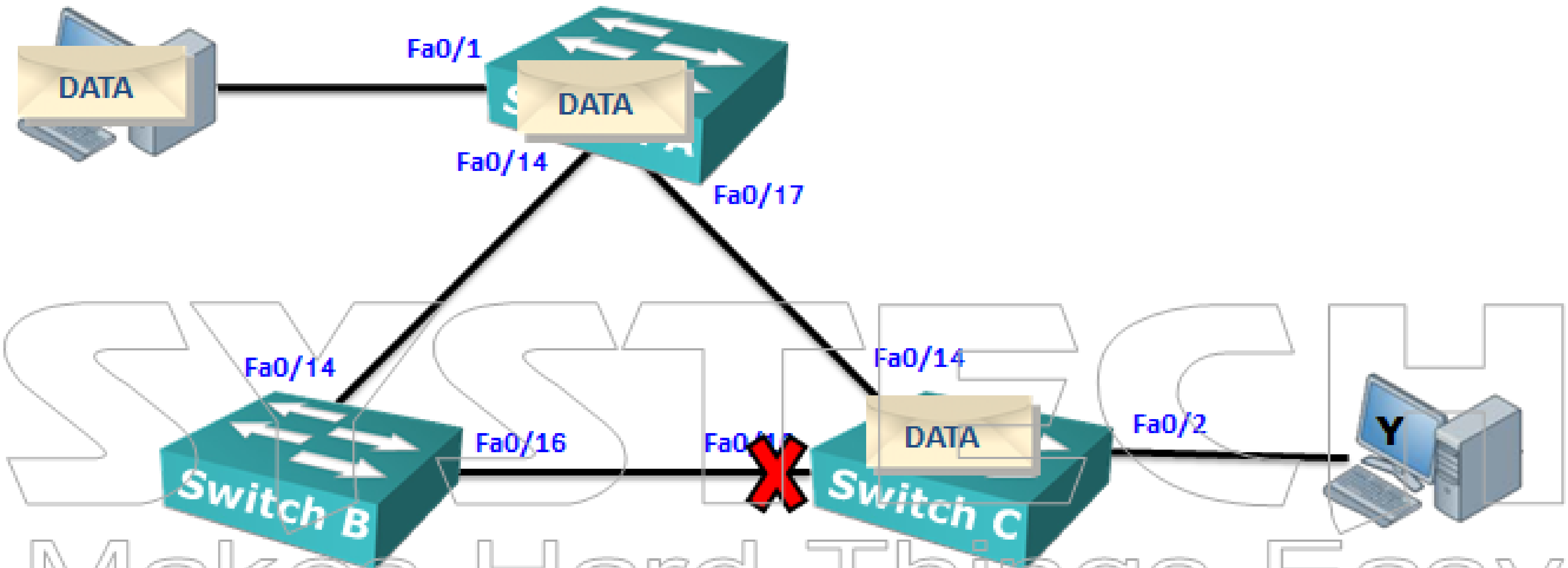
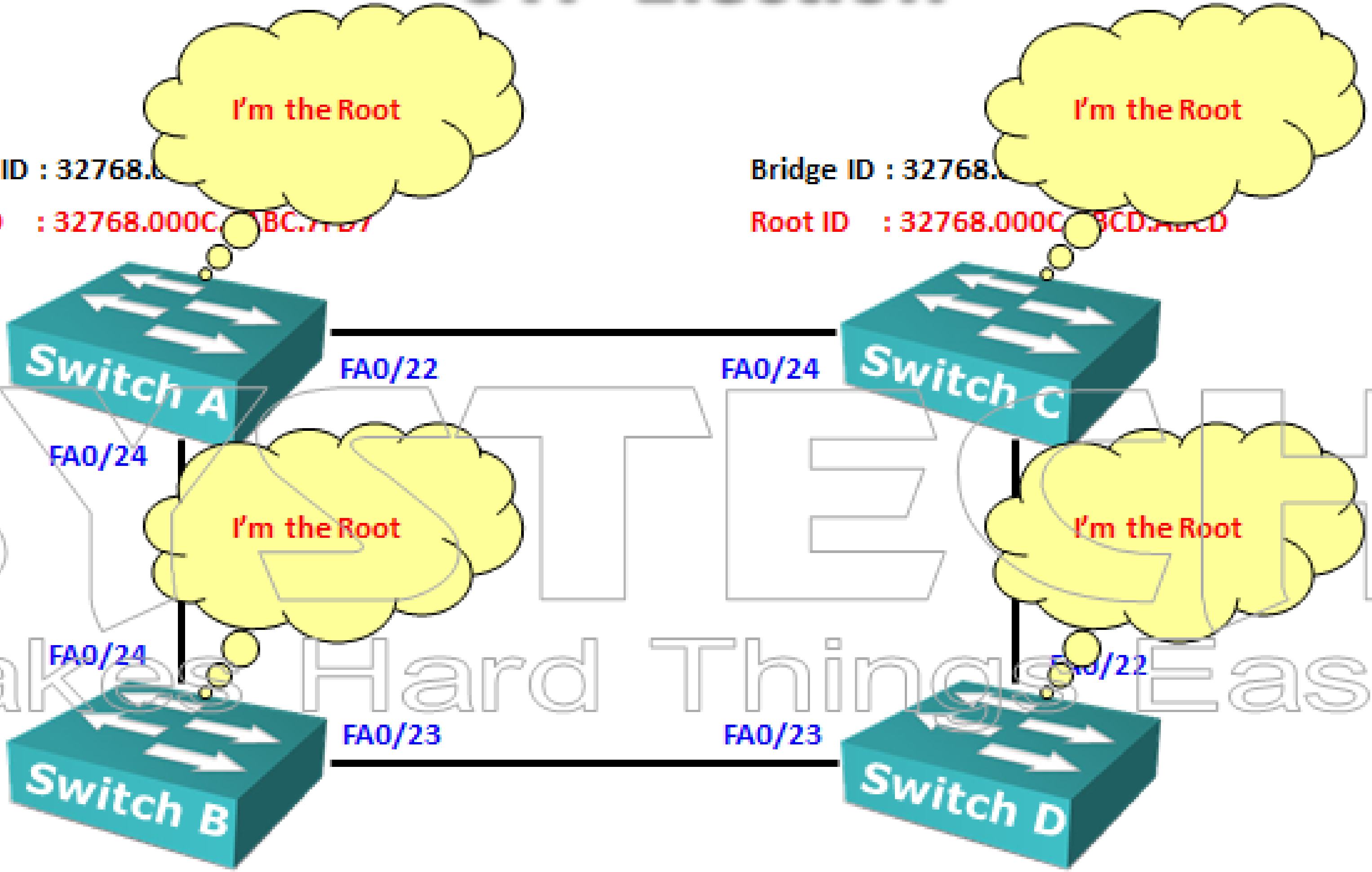
SPANNING TREE PROTOCOL(STP)



- ✓ Broadcast Frames will be forwarded on all interfaces, except the originated link
- ✓ No TTL (Time to Live) for Ethernet Frames, so loop forever
- ✓ Fix the loop by disconnecting cable between switch A & C , A & B or B & C
- ✓ Switch may crash because of overburden with traffic
- ✓ **Spanning Tree will block one or more interfaces and helps to create loop-free topology**
- ✓ **STP is open standard protocol (IEEE 802.1D)**
- ✓ **Enabled by default on all cisco switches**

STP Election



Bridge ID : 32768.0[illegible]

Root ID : 32768.000C.[illegible]BC.[illegible]

Bridge ID : 32768.[illegible]

Root ID : 32768.000C.[illegible]BCD.[illegible]CD

Bridge ID : 32768.0A07.D75F.0CAB

Root ID : 32768.0A07.D75F.0CAB

Bridge ID : 32768.0107.0001.0002

Root ID : 32768.0107.0001.0002

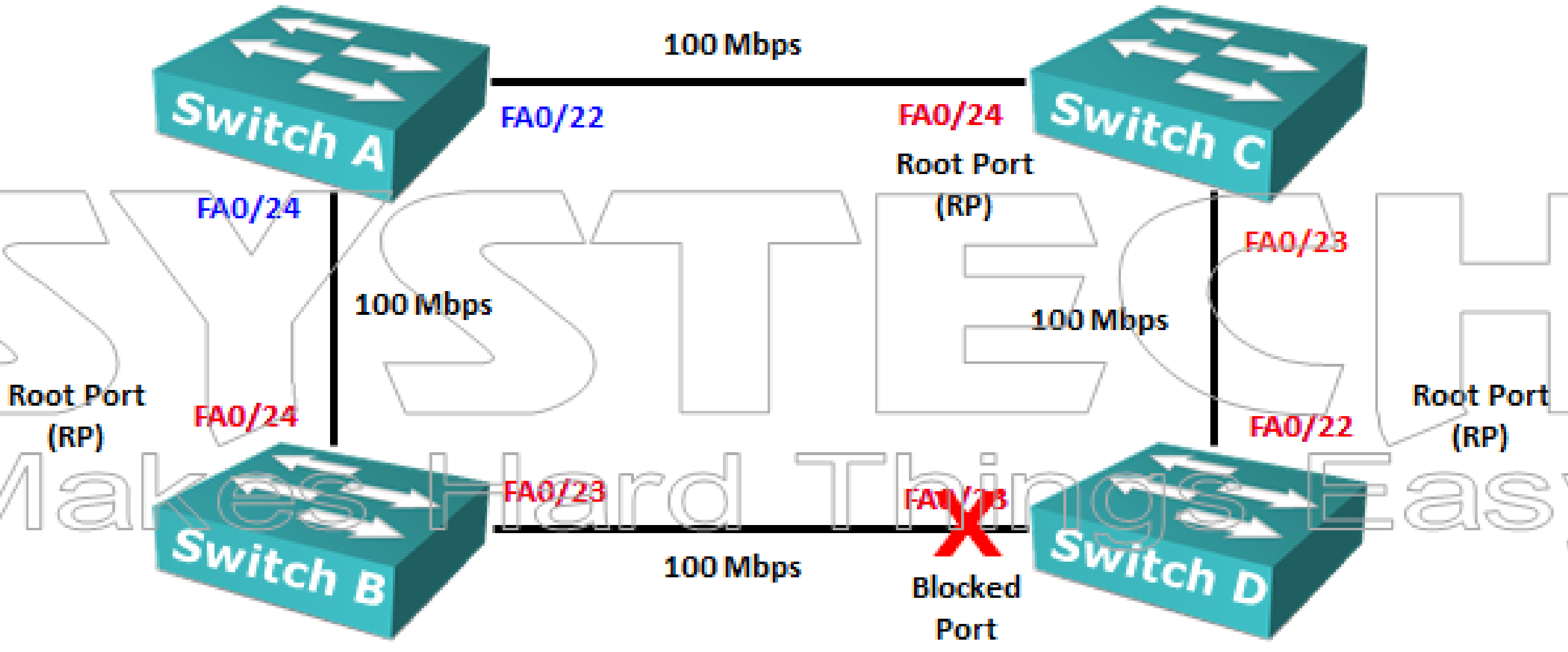
STP Election

Bridge ID : 32768.000C.4ABC.7FD7

Root ID : 32768.000C.4ABC.7FD7

Bridge ID : 32768.000C.ABCD.ABCD

Root ID : 32768.000C.4ABC.7FD7



Bridge ID : 32768.0A07.D75F.0CAB

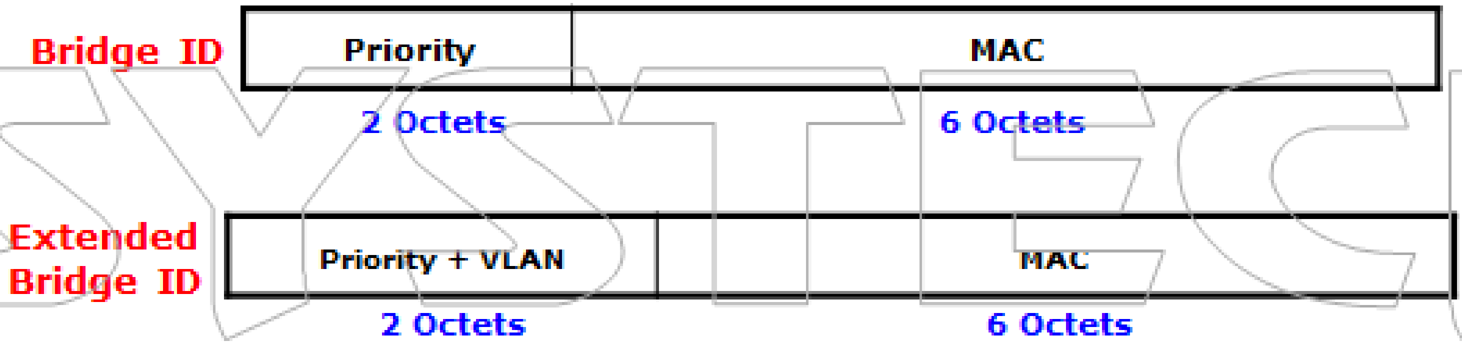
Root ID : 32768.000C.4ABC.7FD7

Bridge ID : 32768.0107.0001.0002

Root ID : 32768.000C.4ABC.7FD7

STP Election

- ✓ Elect one switch having best Bridge ID as Root Bridge.
- ✓ Other switches connect to Root Bridge with best cost.
- ✓ Select lowest Sender's Bridge ID.
- ✓ Select lowest Port ID



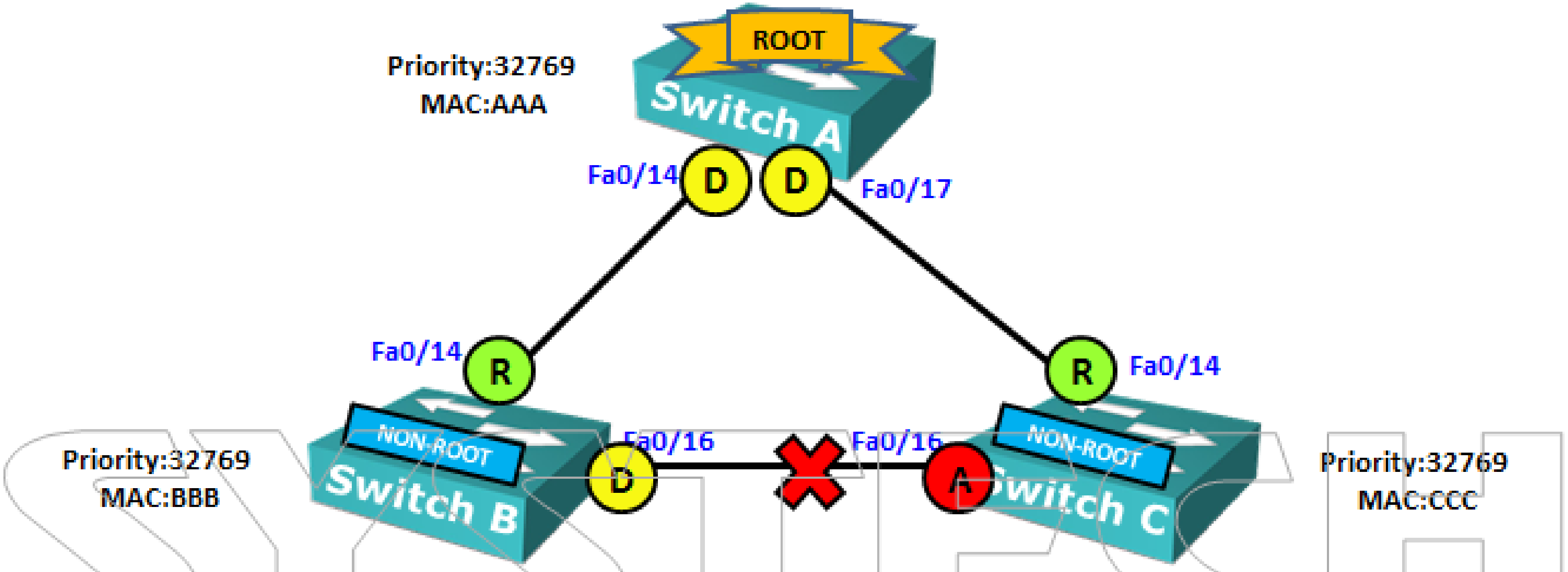
Bridge ID

Priority	MAC
2 Octets	6 Octets

Extended Bridge ID

Priority + VLAN	MAC
2 Octets	6 Octets

- ✓ Switches running spanning-tree exchange information with a special message called BPDU
- ✓ Bridge Protocol Data Unit (BPDU)
- ✓ All the information in BPDU is needed to create and maintain the spanning-tree topology
- ✓ Bridge identifier will have priority & MAC address
- ✓ Wireshark can capture a BPDU



✓ First SPT will detect a root bridge. (lowest bridge identifier)

✓ Switch A will become the root bridge.

✓ All other switches are non-root

✓ Interfaces that forward traffic are called designated ports

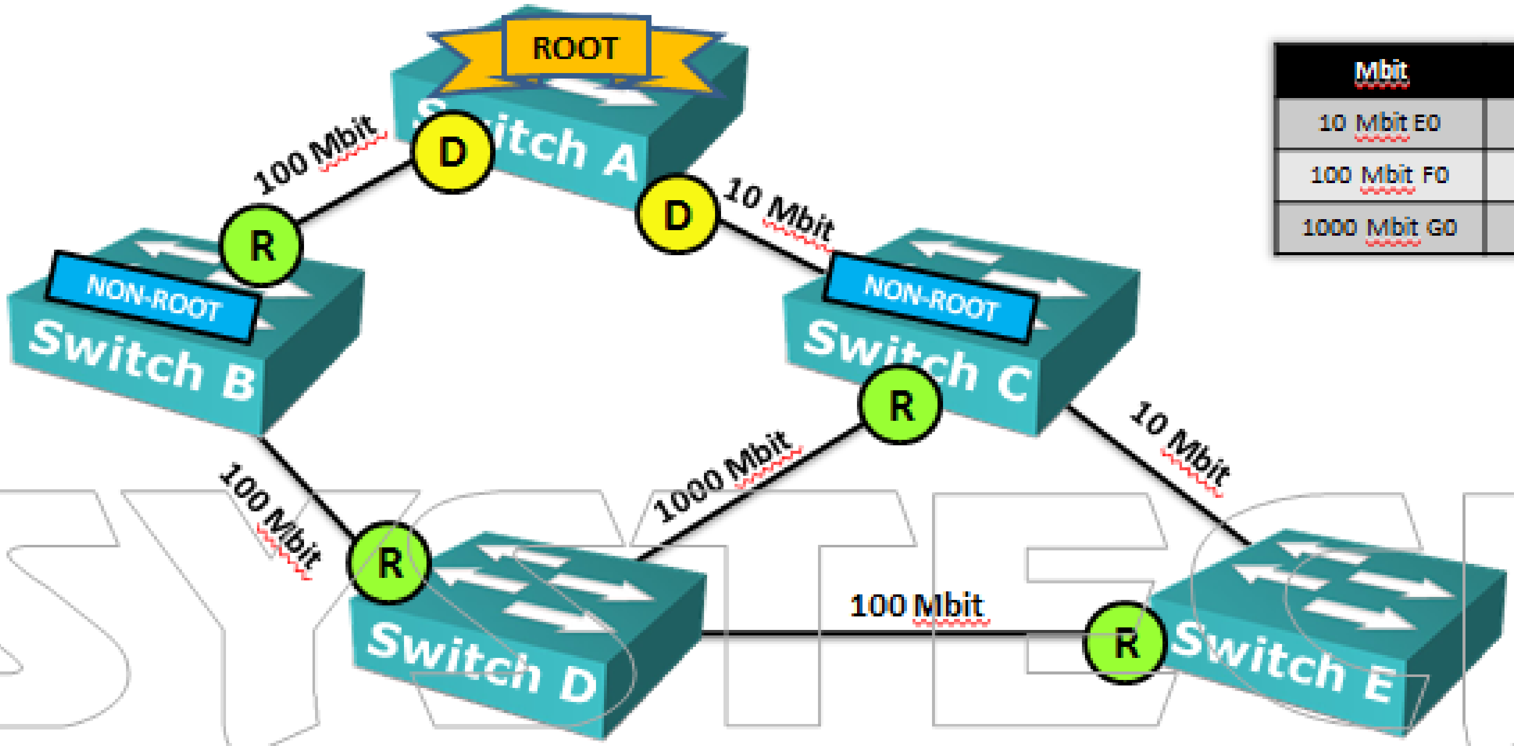
✓ On a root bridge the interfaces are always in forwarding mode because non root switches will need to find the root bridge

✓ All the non root switches has to find shortest path to the root bridge.

✓ Switch B (Fa0/14) Switch C (Fa0/14)

✓ Interfaces that leads to root bridge is called root port

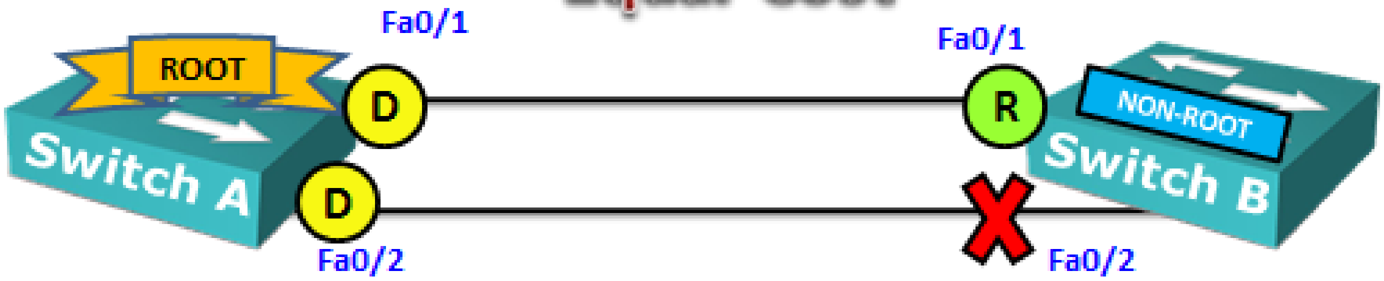
✓ To break the loop switch B&C will compare their identifiers & switch B has lower MAC address so switch C port (fa0/16) is blocked and that port is called an alternate port. Switch B fa0/16 port will be designated port



Mbit	COST
10 Mbit E0	100
100 Mbit F0	19
1000 Mbit G0	4

- ✓ Switch B will use the direct link to switch A as root port ,cost 19
- ✓ Switch C will use path through switch D , cost (19+19+4)
- ✓ Switch D will use path through switch B , cost (19+19)
- ✓ Switch E , BPDU from switch C: cost 42 | BPDU from switch D: cost 38
- ✓ Switch E will use path through switch D
- ✓ Switches only make decisions on BPDUs and they have no idea about topology
- ✓ Best BPDU is the one with shortest path to the root bridge.

Equal Cost



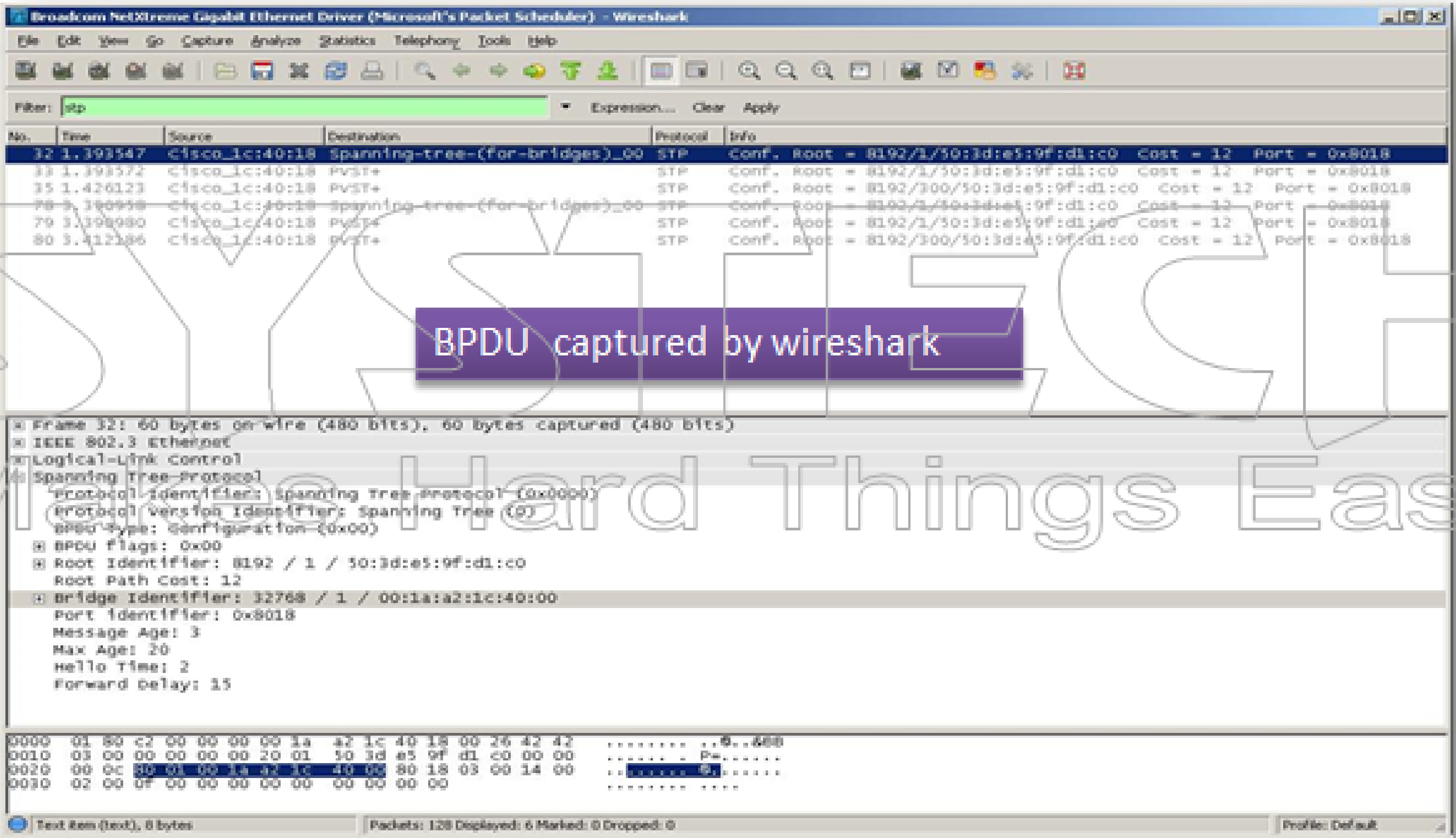
- ✓ We have redundancy between two switches means loops ,so spanning-tree will block one of the interfaces on switch B
- ✓ Switch B will receive BPDU on the both the interfaces but the root path cost field will be same
- ✓ When the cost is equal spanning tree looks at port priority which is by default same for all port
- ✓ When priority is equal spanning tree looks at lowest interface number and so fa0/2 will be blocked

STP Port Status

State	Forward Frames	Learn MAC address	Duration
Blocking	No	No	20 sec
Listening (Root & Designated)	No	No	15 sec
learning	No	yes	15 sec
forwarding	yes	yes	

BPDU

Protocol	Version	Message type	Root ID	Cost	Bridge ID	Port ID	Message Age	Max Time	Hello	Forward Delay


Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: stp

```
No.  Time      Source            Destination                       Protocol Info
  32 1.393547  Cisco_1c:40:18    Spanning-tree-(for-bridges)_00    STP      Conf. Root = 8192/1/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
  33 1.393572  Cisco_1c:40:18    PVST+                             STP      Conf. Root = 8192/1/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
  35 1.426123  Cisco_1c:40:18    PVST+                             STP      Conf. Root = 8192/300/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
  78 3.390958  Cisco_1c:40:18    Spanning-tree-(for-bridges)_00    STP      Conf. Root = 8192/1/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
  79 3.398980  Cisco_1c:40:18    PVST+                             STP      Conf. Root = 8192/1/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
  80 3.412186  Cisco_1c:40:18    PVST+                             STP      Conf. Root = 8192/300/50:3d:e5:9f:d1:c0   Cost = 12   Port = 0x8018
```

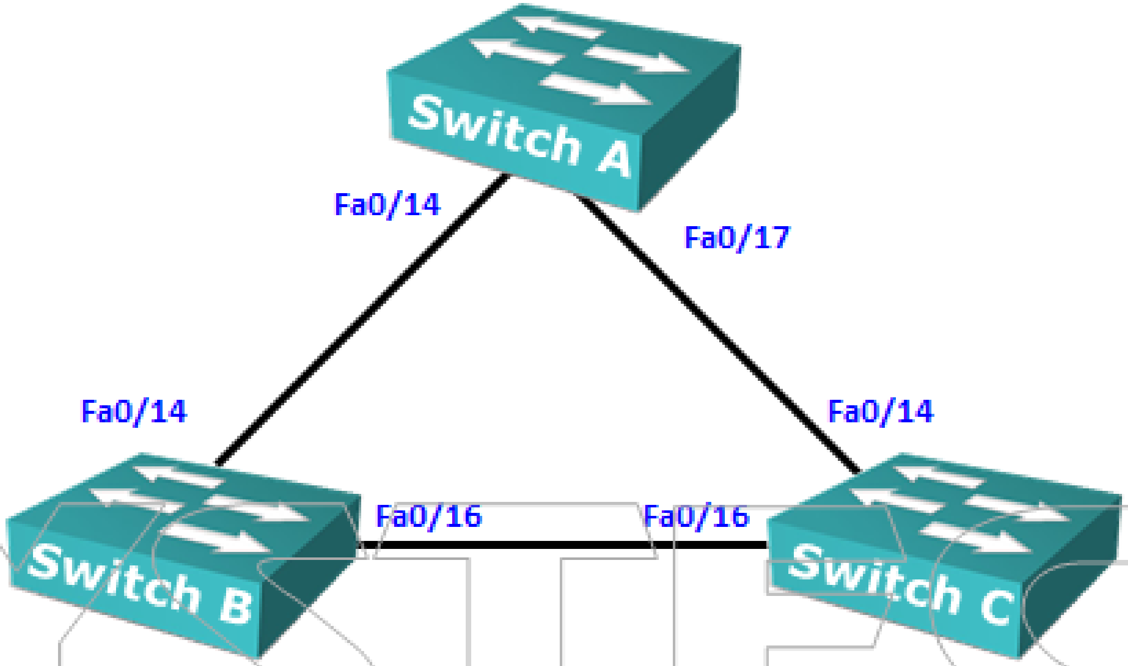
BPDU captured by wireshark

```
⊞ Frame 32: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ IEEE 802.3 Ethernet
⊞ Logical-Link Control
⊟ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
  ⊞ BPDU flags: 0x00
  ⊞ Root Identifier: 8192 / 1 / 50:3d:e5:9f:d1:c0
    Root Path Cost: 12
  ⊞ Bridge Identifier: 32768 / 1 / 00:1a:a2:1c:40:00
    Port identifier: 0x8018
    Message Age: 3
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
```

```
0000   01 80 c2 00 00 00 00 1a   a2 1c 40 18 00 26 42 42    ......... ..@..&BB
0010   03 00 00 00 00 00 20 01   50 3d e5 9f d1 c0 00 00    ...... . P=......
0020   00 0c 80 01 00 1a a2 1c   40 00 80 18 03 00 14 00    ........ @.......
0030   02 00 0f 00 00 00 00 00   00 00 00 00                ........ ....
```

Text item (text), 8 bytes | Packets: 128 Displayed: 6 Marked: 0 Dropped: 0 | Profile: Default

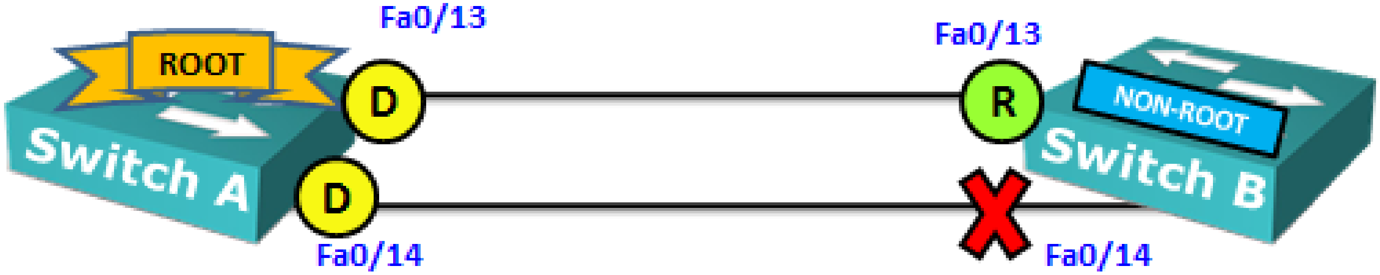
LAB



SWITCH A B C

```
#show spanning-Tree
#spanning-tree vlan 1 root primary
                or
#spanning-tree vlan 1 priority 4096
#show spanninng tree

#int fa0/14
#spanning-Tree cost 500
```



SWITCH A

```
# show spanning-Tree
```

Fa0/13 port Desg

Fa0/14 port - Desg

SWITCH B

```
# show spanning-Tree
```

Fa0/13 port - Root

Fa0/14 port - Block

SWITCH A

```
# int Fa0/14
# spanning-tree port-priority 16
```

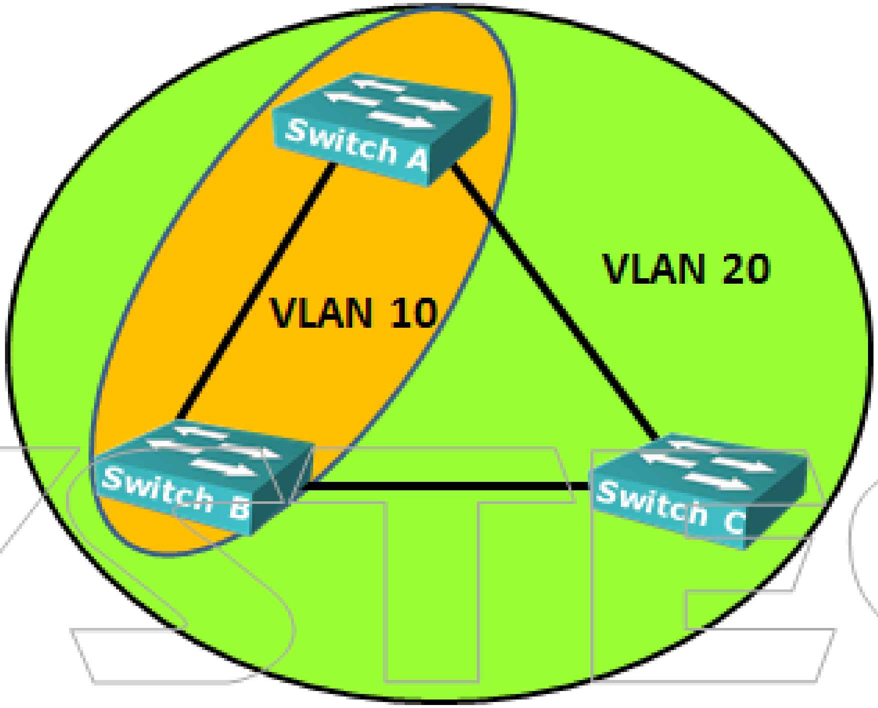
SWITCH B

```
# show spanning-Tree
```

Fa0/13 port - Block

Fa0/14 port - Root

PVST



DO WE HAVE LOOP ???

- ✓ Yes VLAN 20 will have loop
- ✓ CSPT(common Spanning-Tree 802.1D) will calculate single spanning tree for all VLAN
- ✓ PVST(Per VLAN spanning-tree) will create different root bridge for each vlan
- ✓ Multiple Root bridges can do Load balancing

LAB

SWITCH A B C

```
#vlan 10
#vlan 20
#vlan 30
```

SWITCH A

```
#int fa0/14
#switchport trunk encapsulation dot1q
#int fa0/17
#switchport trunk encapsulation dot1q
```

SWITCH B & C

```
#int fa0/14
#switchport trunk encapsulation dot1q
#int fa0/16
#switchport trunk encapsulation dot1q
```

SWITCH A B C

```
#show spanning-tree summary
```

SWITCH C

```
#show spanning-tree vlan 10
#show spanning-tree vlan 20
#show spanning–tree vlan 30
```

SWITCH A

```
#spanning-tree mode pvst
#spanning-tree vlan 10 priority 4096
```

SWITCH B

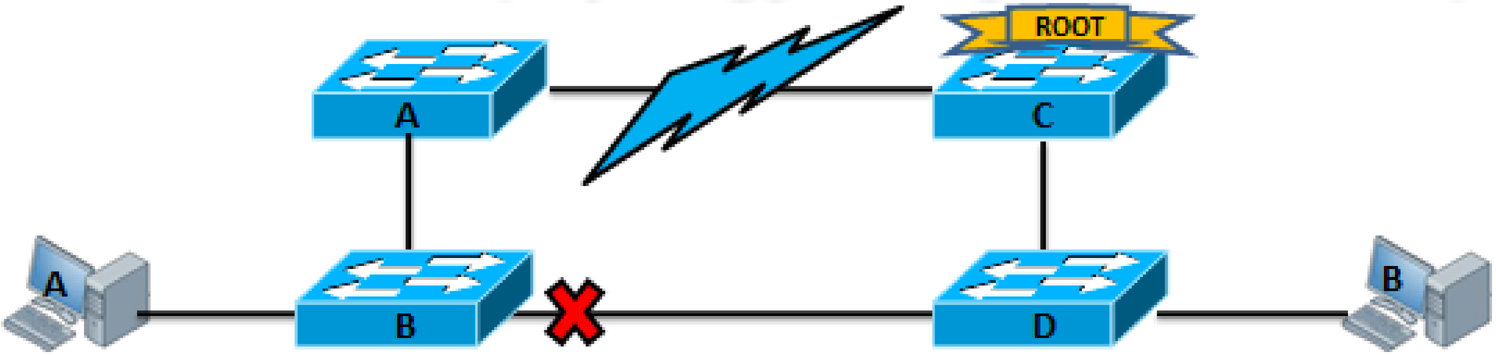
```
#spanning-tree mode pvst
#spanning-tree vlan 20 priority 4096
```

SWITCH C

```
#spanning-tree mode pvst
#spanning-tree vlan 30 priority 4096
```

```
#spanning-tree vlan 10 hello-time 1
(default is 2)
#spanning-tree vlan 20 max-age 6
 (default is 20)
#spanning-tree vlan 30 forward-time 4
(default is 15)
#debug spanning-tree
```

STP TCN (Topology Change Notification)



- ✓ If the link between switch A & C fails. Computer A & B will be unable to communicate with each other until interface of switch B goes into forwarding
- ✓ It will take maximum of 50 seconds
- ✓ But switch B still has MAC address of computer B in its MAC address table and will keep forwarding to switch A where it will be dropped.
- ✓ Computers will not communicate with each other for 300 seconds until MAC address tables age out
- ✓ Age out of MAC address works fine in stable network but not when topology changes occur
- ✓ Spanning tree has topology change mechanism
- ✓ When switch detect a change in the network it will advertise this event to the whole switched network
- ✓ When switches receive this message they will reduce the aging time of MAC address table from 300 to 15 seconds
- ✓ This message is called TCN

```
# show mac address-table aging-time
#debug spanning tree events
```

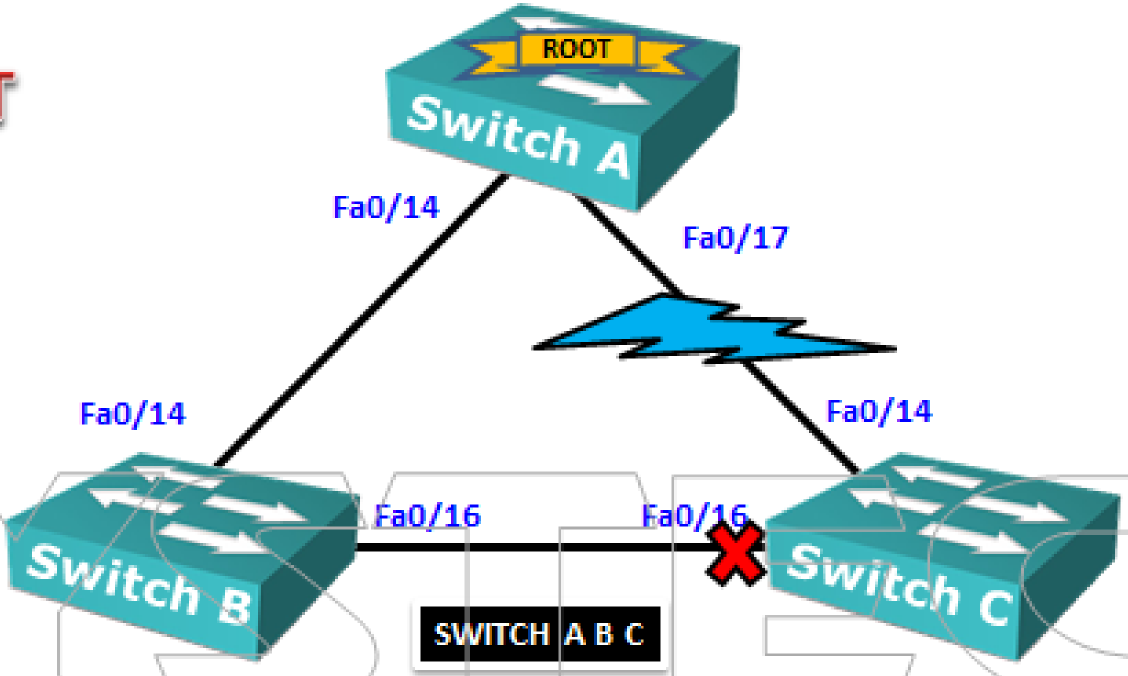
STP PORTFAST

- Each time an interface goes up or down TCN will be generated and all switches will set their aging time to 15 seconds
- If you have lot of hosts it is possible that you end up with a network that is in a constant state of "topology changes"
- To rescue from this problem cisco introduced portfast
- Portfast is cisco proprietary solution to deal with topology changes
- Interfaces with portfast enabled that come up will go to forwarding mode immediatly
- It will skip listening and learning state
- Switch will never generate TCN for an interface that has portfast enabled
- Enable portfast on interfaces connected to hosts because these interfaces are likely to go up and down all the time
- Dont enale portfast to interface connected with another switch

```
#int fa0/2
#spanning-tree portfast

#spanning-tree portfast disable
#show spanning-tree detail
(Number of topology changes 10 last change occured 00:43:29 ago)
```

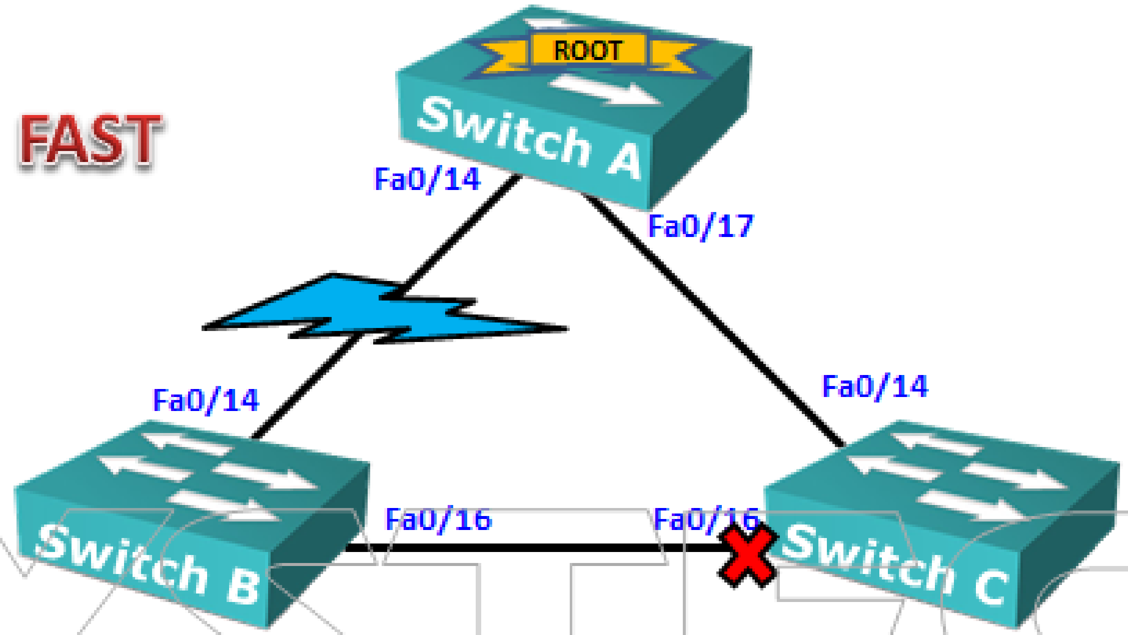
STP UPLINKFAST



- ✓ when fa0/14 interface on switch C fails we'll have to use fa0/16 to reach root bridge.
- ✓ Fa0/16 port will take 30 seconds to become forwarding state
- ✓ It have to cross listening and learning state to become forwarding state
- ✓ So to overcome this we have to use uplinkfast
- ✓ When uplinkfast is enabled an alternate port will go to forward state immediatly if the root port fails.

```
#spanning-tree uplinkfast
```

STP
BACKBONE FAST



- Backbone fast is used to recover from an indirect link failure.
- If connection between switch A & B fails then switch B will detect link failure immediately since it's a directely connected link
- It will not receive any BPDU from root bridge so it assumes itself as root brigde and will send BPDUs to switch C
- Switch C will not receive any BPDU from switch B,because it knows new BPDU is inferior compared to the old one
- When switch receive inferior BPDU it means that the neighbour switch has lost connection to root bridge
- After 20 sec (default timer) the max age timer will expire for old BPDU on fa0/16 on switch C. The interface will go from blocking to listeing state and will send BPDUs towards switch B
- Switch B will recieve this BPDU from switch C

✓ The fa0/16 port on switch C will continue from listening state (15 sec) to the learning state (15 sec) and then to the forwarding state

✓ It takes totally 50 sec (20 max age timer+15 sec listening+15 sec learning state)

✓ If we enable backbone fast it will skip the max age timer so we can save 20 seconds of time

```
#spanning-tree backbonefast
#debug spanning-tree backbonefast detail
```

✓ when switch B loses its connection to root bridge and assumes it as root bridge

✓ Switch B sends inferior BPDU to switch C

✓ When switch C recieves an inferior BPDU it will send root link query (RLQ) on its root port and alternate ports to check if the root bridge is still available

✓ Switch C will recieve a reply to its root link query on the fa0/14 interface to switch A

✓ Switch C recieved a response from the root bridge on its fa0/14 interface and it can now skip max age timer on fa0/16 interface and goes to listening and learning state

✓ We effectivly save 20 seconds (max age timer)

SPANNING TREE TOOLKIT

Tools used to protect spanning tree topology

✓ Portfast: It will configure an access port as edge port so it goes to forwarding mode immediately

✓ BPDUGuard: This will disable (err-disable) an interface that has portfast configured if it receives a BPDU

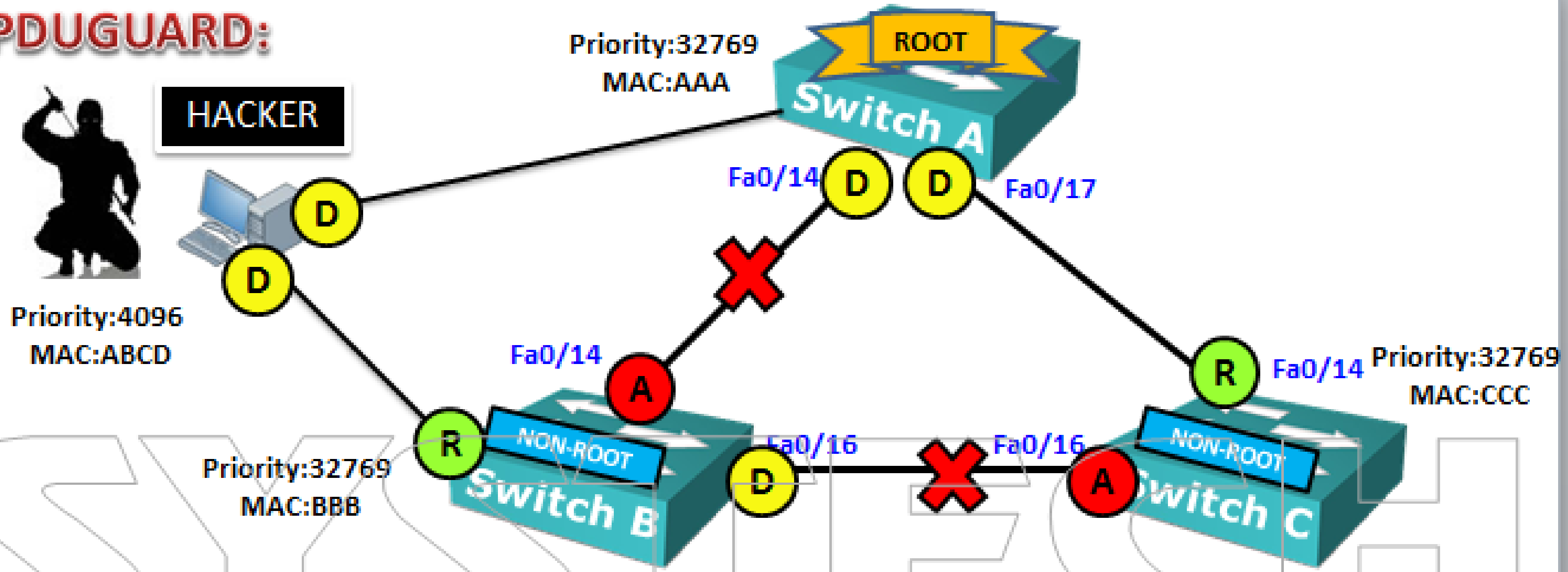
✓ BPDUFilter: This will suppress BPDUs on interfaces

✓ RootGuard: This will prevent a neighbor switch from becoming a root bridge,even if it has the best bridge ID

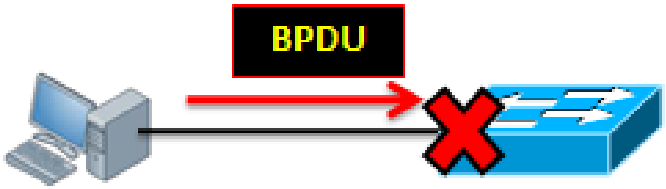
✓ Uplinkfast: it improves convergence time

✓ Backbonefast: it will improve convergence time if you have an indirect link failure

BPDUGUARD:



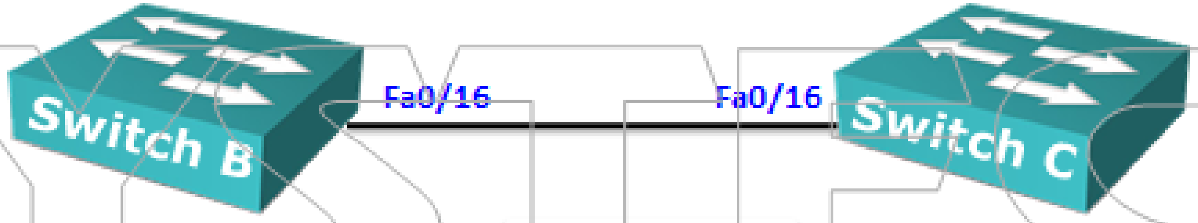
- ✓ If a hacker connect his computer to two switches and make his computer the root bridge, all traffic from switch A or C towards switch B will flow through Hacker's PC
- ✓ Hacker will run wireshark and wait till he captures all !!!!!!!!!



```
# interface fa0/16
#spanning-tree  bpduguard  enable
#spanning-tree  portfast  bpduguard
(globally  activate  BPDUguard  on all  portfast  enabled  interfaces)
#spanning-tree  portfast  default
(portfast  can be enabled globally  for all access  mode  interfaces)
#show  spanning-tree  summary
```

BPDUFILTER:

- ✓ If BPDUfilter is configured globally any interfaces with portfast enable will become a standard port
- ✓ If BPDUfilter is configured on the interface it will ignore incoming BPDUs and will not send any BPDUs
- ✓ BPDUfilter should be configured on access mode port that connect to computers but not on port connected to other switch ;if it is configured then it end up with loop!!!!



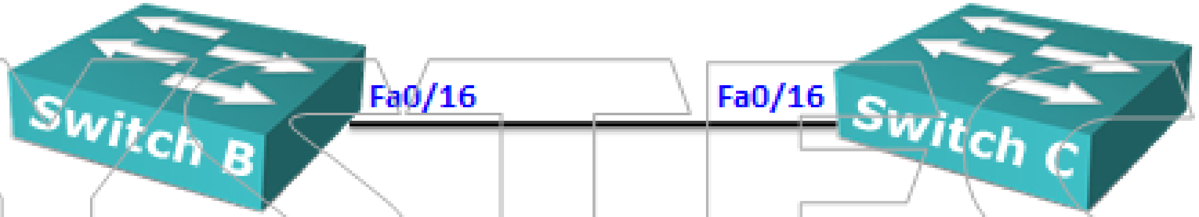
SWITCH C

```
# interface fa0/16
#spanning-tree portfast trunk
#spanning-tree bpdufilter enable
#debug spanning-tree bpdu

#spanning-tree portfast bpdufilter default
```

RootGuard:

- ✓Rootguard will make sure not to accept a certain switch as root bridge
- ✓If a switch suddenly sends superior bridge ID it wont accept it as root bridge



SWITCH B

```
#spanning-tree vlan 1 priority 4096
# interface fa0/16
#spanning-tree guard root
#debug spanning-tree events
```

SWITCH C

```
#spanning-tree vlan 1 priority 0
```

(now switch C will not become root switch)