Layer 2 Engineering – Spanning Tree

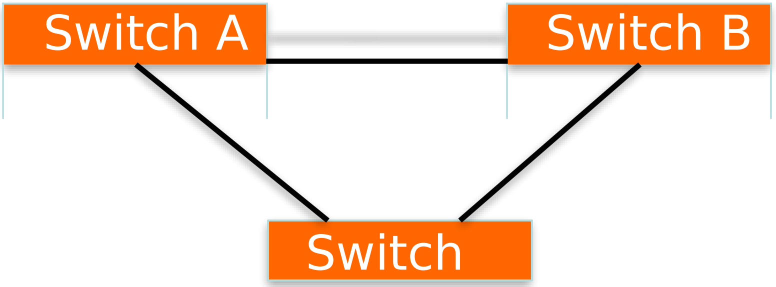
Campus Network Design & Operations Workshop

UNIVERSITY OF OREGON

Last updated 24th October 2022

Network Startup Resource Center

Switching Loop



When there is more than one path between two switches

What are the potential problems?

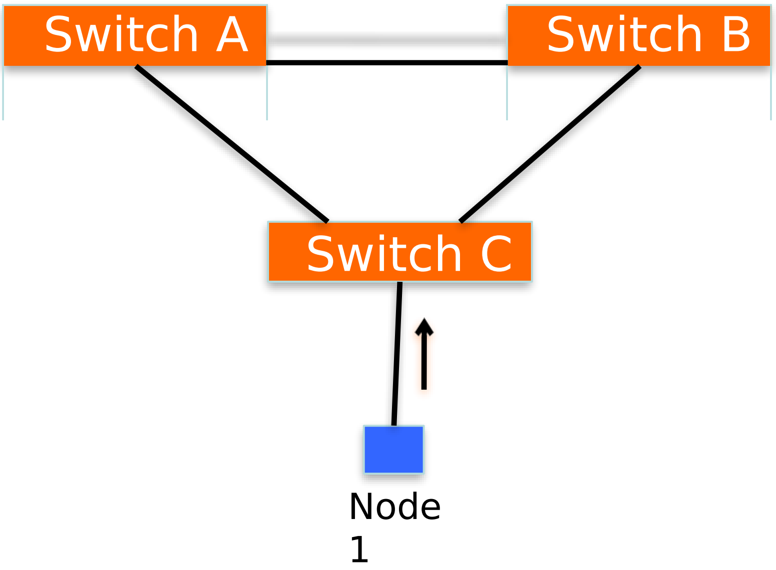
Switching Loop

- If there is more than one path between two switches:
  - Forwarding tables become unstable
    - Source MAC addresses are repeatedly seen coming from different ports

Switching Loop

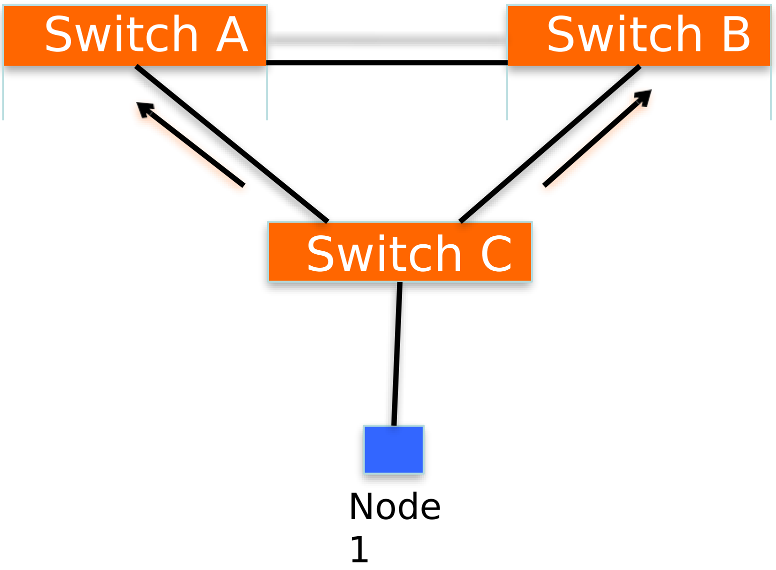
- If there is more than one path between two switches:
  - Forwarding tables become unstable
    - Source MAC addresses are repeatedly seen coming from different ports
  - Switches will broadcast each other’s broadcasts
    - All available bandwidth is utilized
    - Switch processors cannot handle the load

Switching Loop



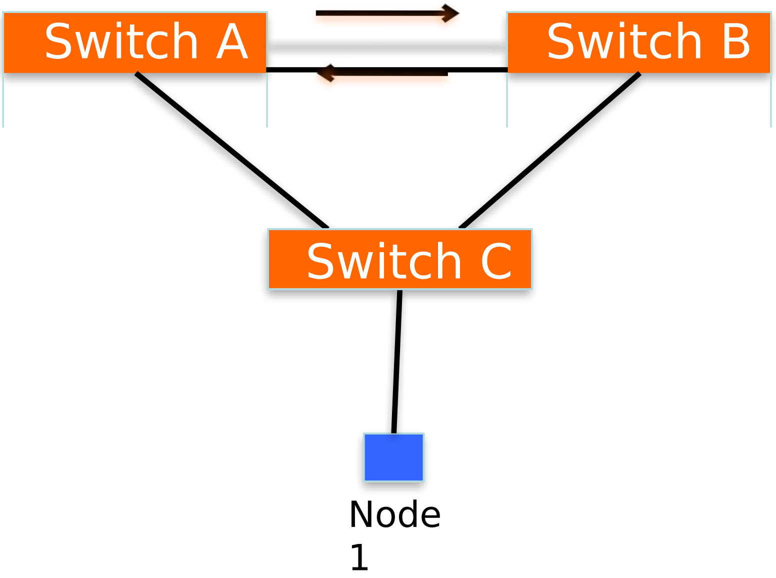
Node 1 sends a broadcast frame (e.g an ARP request)

Switching Loop



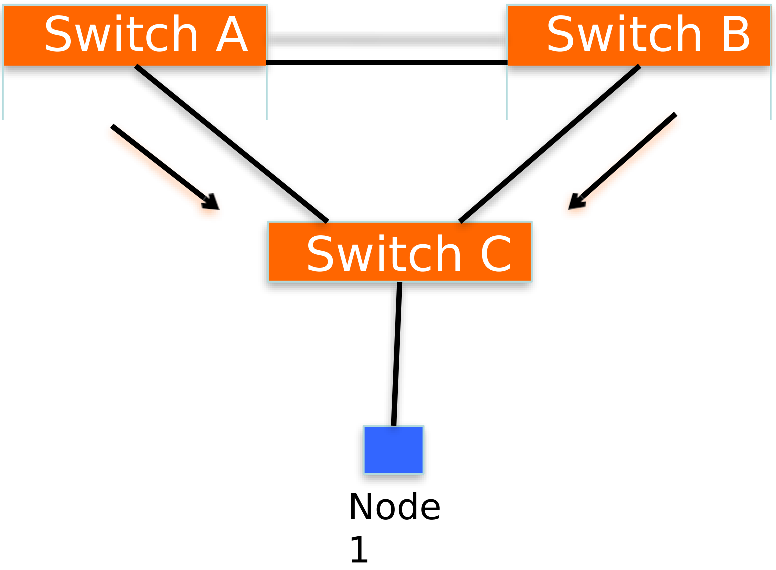
Switch C broadcasts node 1’s frame out every port

Switching Loop



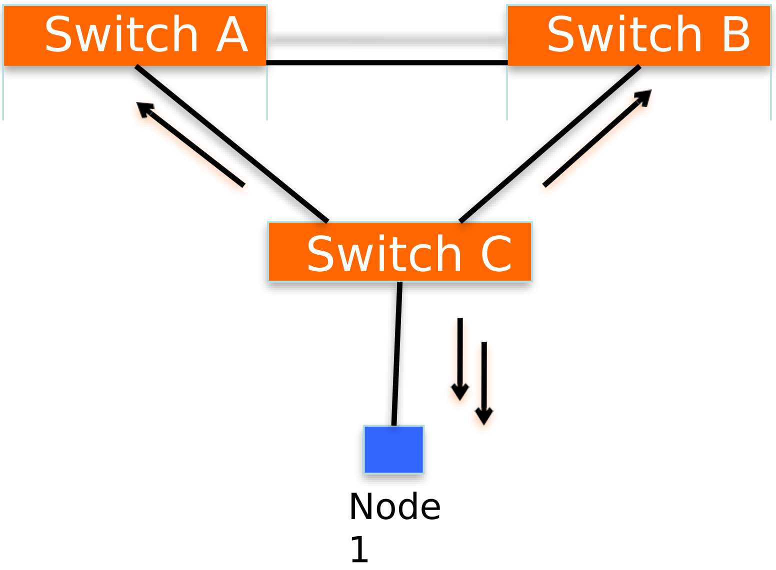
Switches A and B broadcast node 1’s frame out every port

Switching Loop



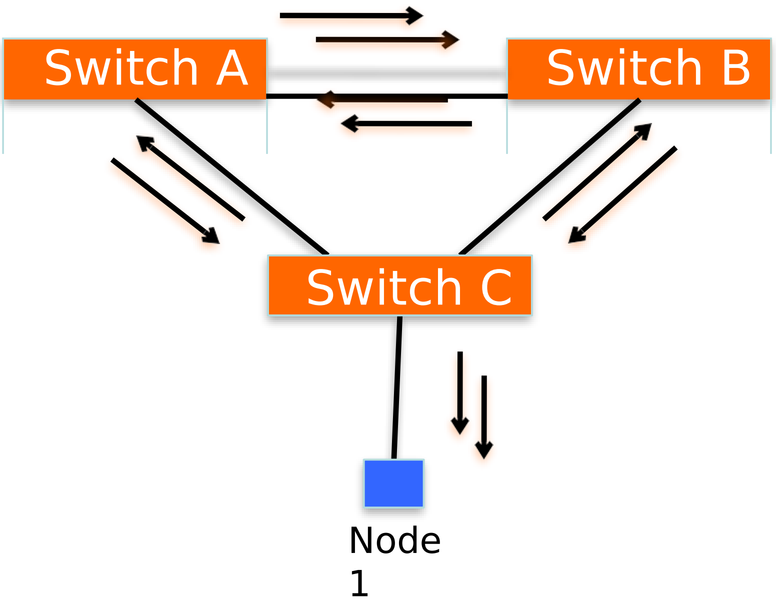
Switches A and B broadcast node 1's frame out every port

Switching Loop



Switch C broadcasts node 1's frame out every port

Switching Loop – End Result



They receive each other's broadcasts, which they then forward again out every port! There is now an infinite loop of broadcasts

This creates what is known as a ***broadcast storm***

Good Switching Loops

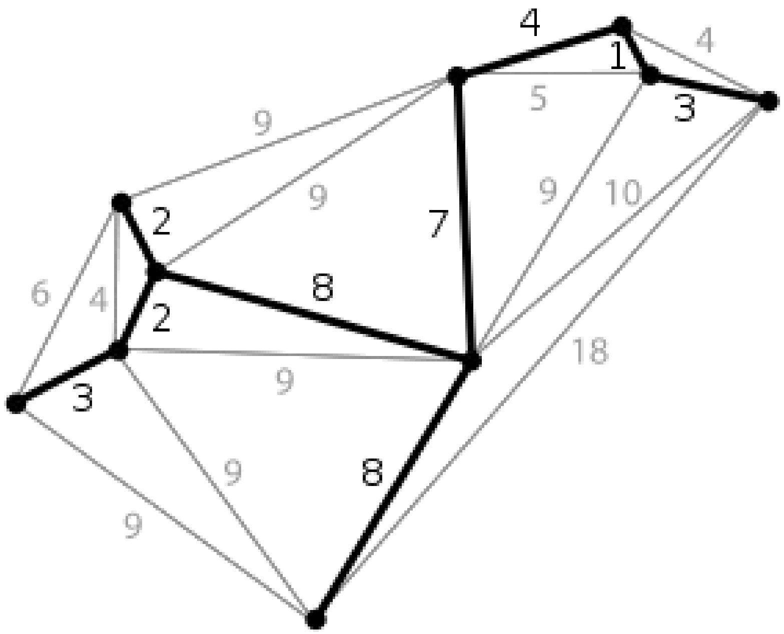
- But you can take advantage of loops!
  - Redundant paths improve resilience when:
    - A switch fails
    - Wiring breaks

Good Switching Loops

- But you can take advantage of loops!
  - Redundant paths improve resilience when:
    - A switch fails
    - Wiring breaks
- How to achieve redundancy without creating dangerous traffic loops?

What is a Spanning Tree?

- “Given a connected, undirected graph, a *spanning tree* of that graph is a subgraph which is a tree and connects all the vertices together”.
- A single graph can have many different spanning trees.



UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Spanning Tree Protocol

- *The purpose of the protocol is to have bridges dynamically discover a subset of the topology that is loop-free (a tree) and yet has just enough connectivity so that where physically possible, there is a path between every switch*

Spanning Tree Protocol

- Several standard flavors:
  - Traditional Spanning Tree (IEEE 802.1d)
  - Rapid Spanning Tree or RSTP (IEEE 802.1w)
  - Multiple Spanning Tree or MSTP (IEEE 802.1s)
- Proprietary flavors:
  - Per-VLAN Spanning Tree or PVST (Cisco)
  - Rapid Per-VLAN Spanning Tree or RPVST+ (Cisco)

Traditional Spanning Tree (802.1d)

- Switches exchange messages that allow them to compute the Spanning Tree
  - These messages are called BPDUs (Bridge Protocol Data Units)

Traditional Spanning Tree (802.1d)

- Switches exchange messages that allow them to compute the Spanning Tree
  - These messages are called BPDUs (Bridge Protocol Data Units)
- There are two types of BPDUs:
  - Configuration
  - Topology Change Notification (TCN)

Traditional Spanning Tree (802.1d)

- First Step:
  - Decide on a point of reference: the ***Root Bridge***

Traditional Spanning Tree (802.1d)

- First Step:
  - Decide on a point of reference: the ***Root Bridge***
  - The election process is based on the Bridge ID

Traditional Spanning Tree (802.1d)

- First Step:
  - Decide on a point of reference: the ***Root Bridge***
  - The election process is based on the Bridge ID
  - The Bridge ID is composed of:
    - The Bridge Priority: A two-byte value that is configurable
    - The MAC address: A unique, hardcoded address that cannot be changed.

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Root Bridge Selection (802.1d)

- Each switch starts by sending out BPDUs with a Root Bridge ID equal to its own Bridge ID
  - *I am the root!*

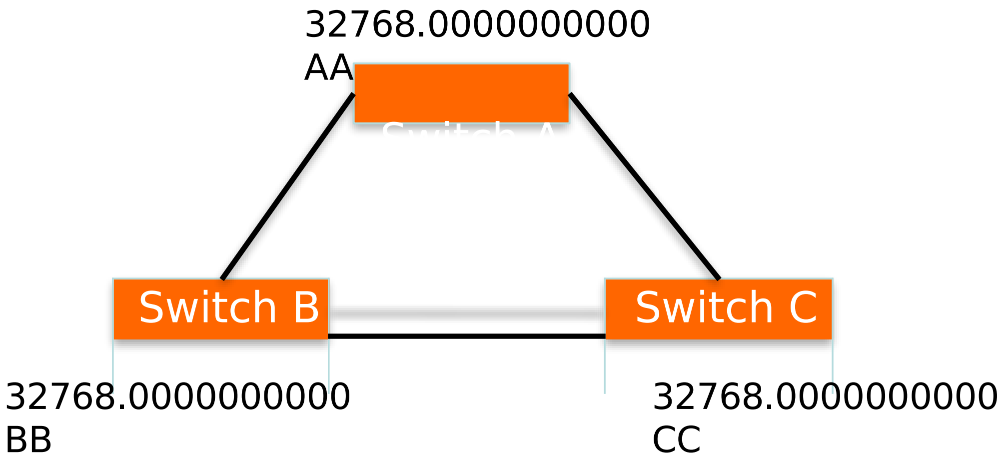
Root Bridge Selection (802.1d)

- Each switch starts by sending out BPDUs with a Root Bridge ID equal to its own Bridge ID
  - *I am the root!*
- Received BPDUs are analyzed to see if a <u>lower</u> Root Bridge ID is being announced
  - If so, each switch replaces the value of the advertised Root Bridge ID with this new lower ID

Root Bridge Selection (802.1d)

- Each switch starts by sending out BPDUs with a Root Bridge ID equal to its own Bridge ID
  - *I am the root!*
- Received BPDUs are analyzed to see if a <u>lower</u> Root Bridge ID is being announced
  - If so, each switch replaces the value of the advertised Root Bridge ID with this new lower ID
- Eventually, they all agree on who the Root Bridge is

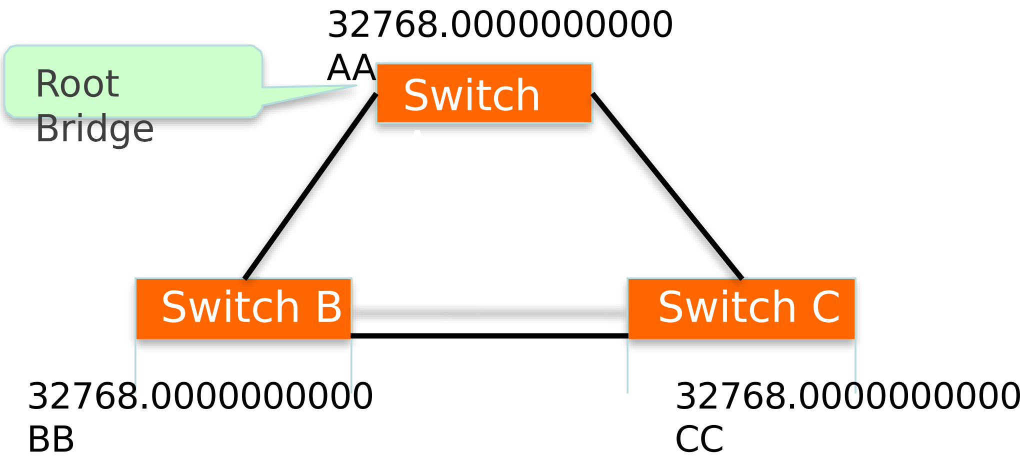
Root Bridge Selection (802.1d)



All switches have the same priority.

Which switch is the elected root bridge?

Root Bridge Selection (802.1d)



All switches have the same priority.

Which switch is the elected root bridge?

Root Port Selection (802.1d)

- Now each switch needs to figure out where it is in relation to the Root Bridge
  - Each switch needs to determine its ***Root Port***

Root Port Selection (802.1d)

- Now each switch needs to figure out where it is in relation to the Root Bridge
  - Each switch needs to determine its ***Root Port***
  - The key is to find the port with the <u>lowest</u> ***Root Path Cost***
    - The cumulative cost of all the links leading to the Root Bridge

Root Port Selection (802.1d)

- Each link on a switch has a ***Path Cost***
  - Inversely proportional to the link speed
  - The[illegible] cos[illegible]

Link Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	N/A	200
1 Tbps	N/A	20

Root Port Selection (802.1d)

- ***Root Path Cost*** is the accumulation of a link’s Path Cost and the Path Costs learned from neighboring Switches.
  - It answers the question: *How much does it cost to reach the Root Bridge through this port?*

Root Port Selection (802.1d)

1. Root Bridge sends out BPDUs with a Root Path Cost value of 0

Root Port Selection (802.1d)

1. Root Bridge sends out BPDUs with a Root Path Cost value of 0
2. Neighbor receives BPDU and adds port’s Path Cost to Root Path Cost received

Root Port Selection (802.1d)

1. Root Bridge sends out BPDUs with a Root Path Cost value of 0
2. Neighbor receives BPDU and adds port's Path Cost to Root Path Cost received
3. Neighbor sends out BPDUs with new cumulative value as Root Path Cost

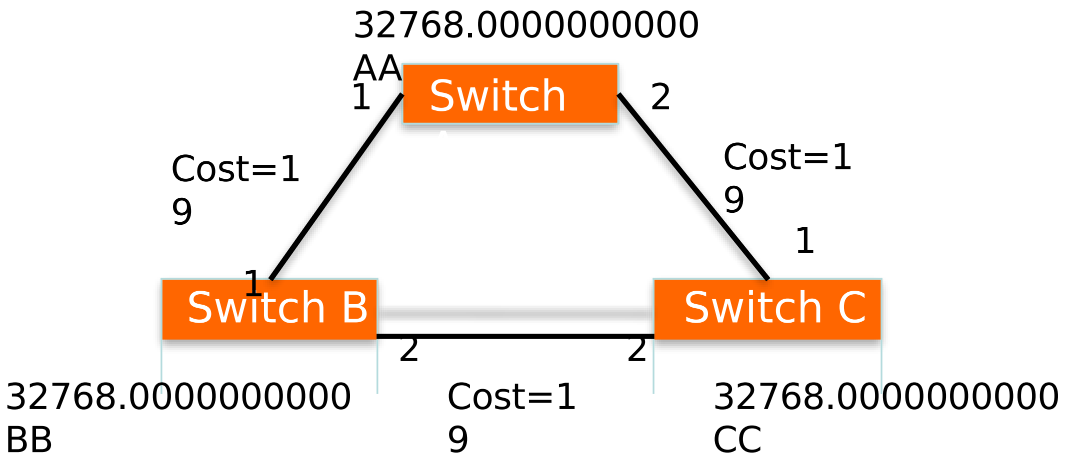
Root Port Selection (802.1d)

1. Root Bridge sends out BPDUs with a Root Path Cost value of 0
2. Neighbor receives BPDU and adds port’s Path Cost to Root Path Cost received
3. Neighbor sends out BPDUs with new cumulative value as Root Path Cost
4. Other neighbors down the line keep adding in the same fashion

Root Port Selection (802.1d)

- On each switch, the port which has the lowest Root Path Cost becomes the ***Root Port***
  - This is the port with the best path to the Root Bridge

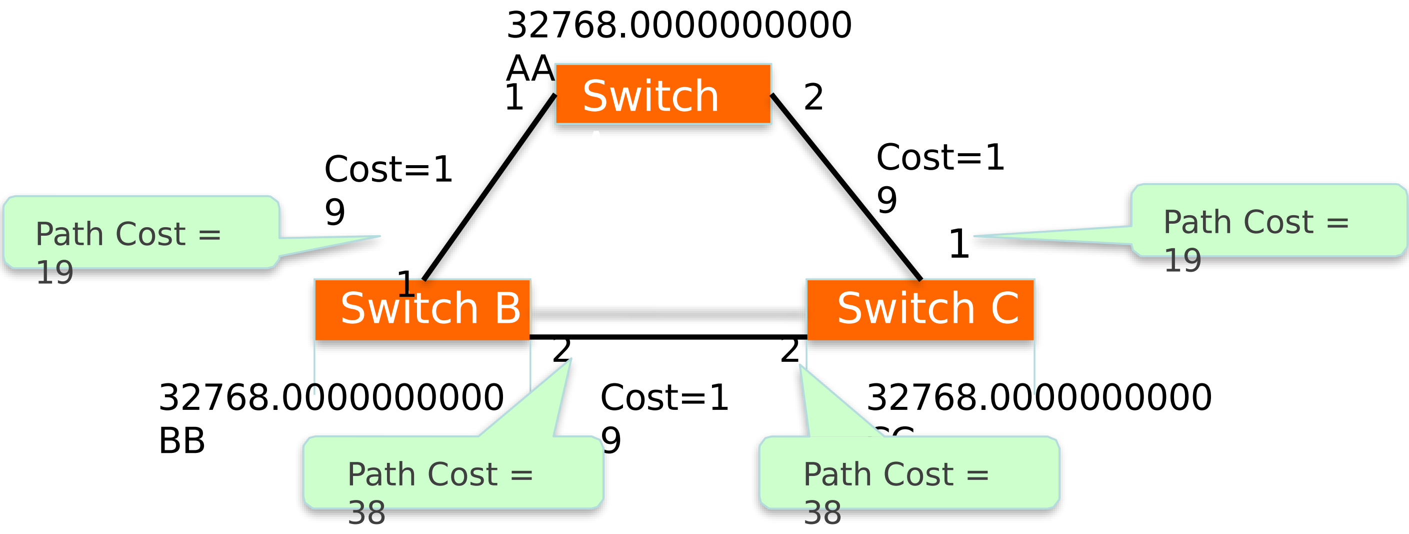
Root Port Selection (802.1d)



What is the Path Cost on each Port?

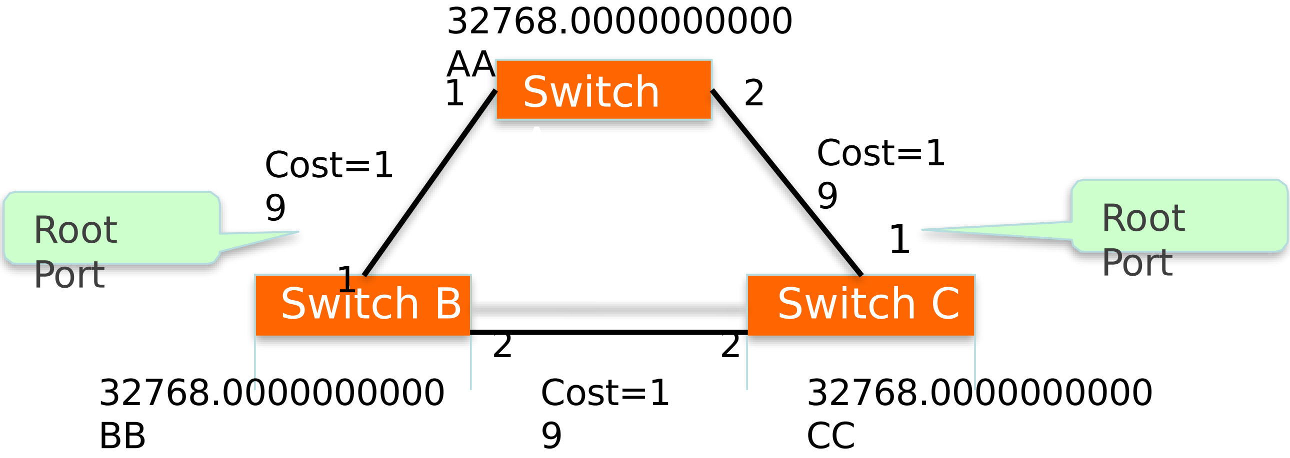
What is the Root Port on each switch?

Root Port Selection (802.1d)



Note: Path Cost is the sum of the value in the BPDU received from the neighbour plus the link cost

Root Port Selection (802.1d)



UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Electing Designated Ports (802.1d)

- OK, we now have selected the root ports, but we haven’t solved the loop problem yet:
  - *The links are still active!*

Electing Designated Ports (802.1d)

- OK, we now have selected root ports, but we haven’t solved the loop problem yet:
  - *The links are still active!*
- Each network segment needs to have only one switch forwarding traffic to and from that segment

Electing Designated Ports (802.1d)

- OK, we now have selected root ports, but we haven't solved the loop problem yet:
  - *The links are still active!*
- Each network segment needs to have only one switch forwarding traffic to and from that segment
- Switches then need to identify one **Designated Port** per network segment
  - The one with the lowest cumulative Root Path Cost to the Root Bridge

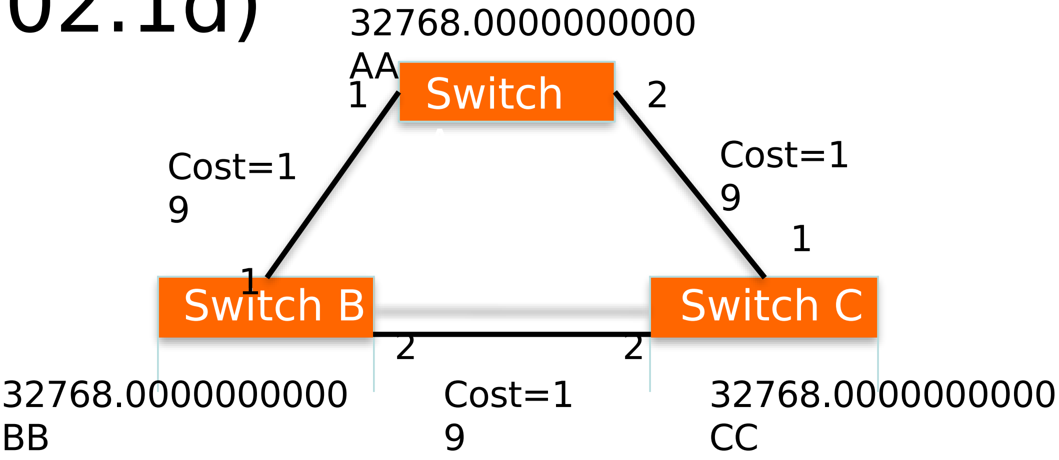
Electing Designated Ports (802.1d)

- Two or more ports in a segment having identical Root Path Costs is possible, which results in a tie condition

Electing Designated Ports (802.1d)

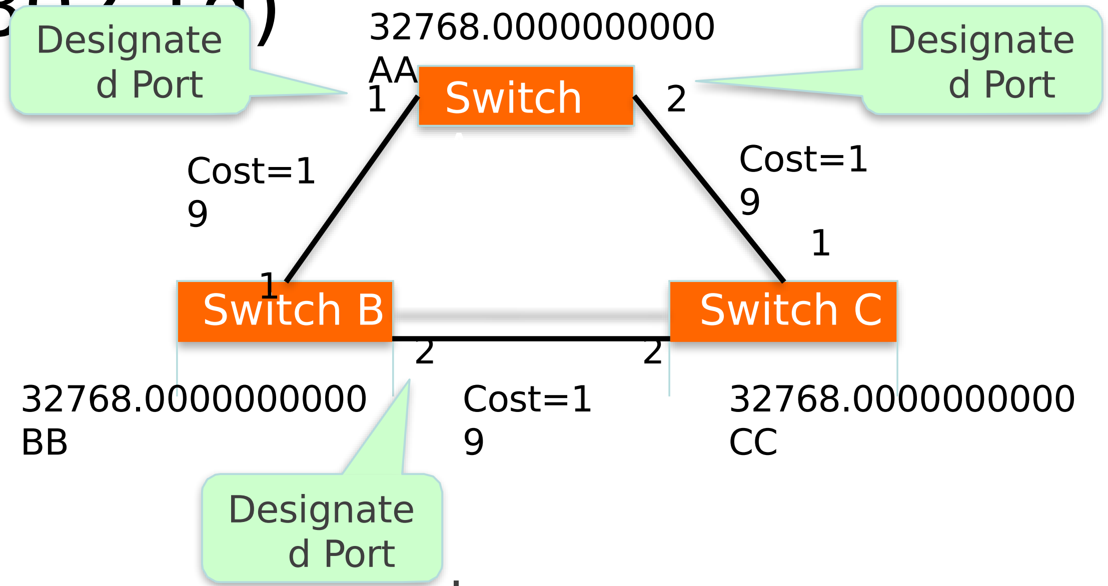
- Two or more ports in a segment having identical Root Path Costs is possible, which results in a tie condition
- All STP decisions are based on the following sequence of conditions:
    1. Lowest Root Bridge ID
    2. Lowest Root Path Cost to Root Bridge
    3. Lowest Sender Bridge ID
    4. Lowest Sender Port ID

Electing Designated Ports (802.1d)



Which port should be the Designated Port on each segment?

Electing Designated Ports (802.1d)



I
n
t
h
e

UNIVERSITY OF OREGON

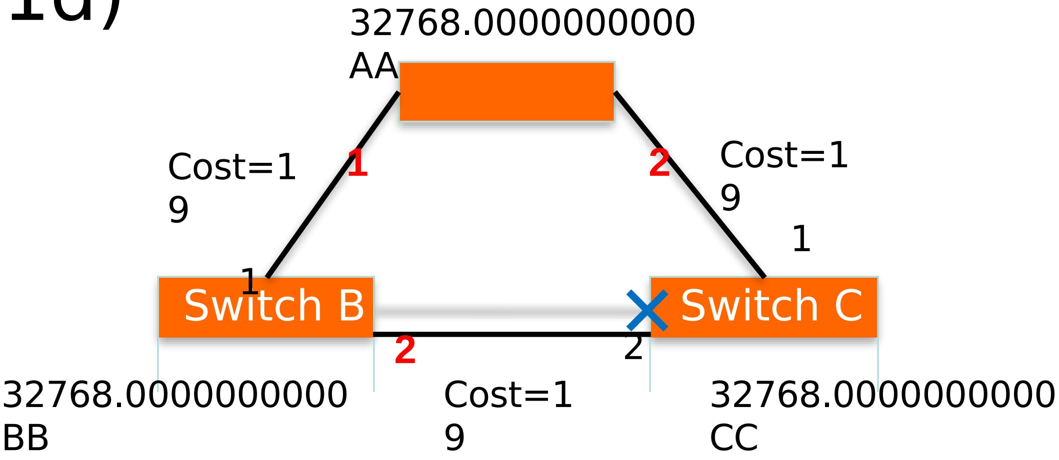
NSRC Network Startup Resource Center

Blocking a port

- Any port that is not elected as either a Root Port, nor a Designated Port is put into the
- **Blocking State**.

  This step effectively breaks the loop and completes the Spanning Tree.

Designated Ports on each segment (802.1d)



Port 2 in Switch C is then put into the ***Blocking State*** because it is ***neither a Root Port nor a Designated Port***

Spanning Tree Protocol States

- Disabled
  - Port is shut down
- Blocking
  - Not forwarding frames
  - Receiving BPDUs
- Listening
  - Not forwarding frames
  - Sending and receiving BPDUs

Spanning Tree Protocol States

- Learning
  - Not forwarding frames
  - Sending and receiving BPDUs
  - Learning new MAC addresses
- Forwarding
  - Forwarding frames
  - Sending and receiving BPDUs
  - Learning new MAC addresses
- *Once a link is detected on a port in a switch configured with spanning tree, it will typically go through all the states presented in order from disabled and stop at either* ***blocking*** *or* ***forwarding*** *depending on STP decisions.*

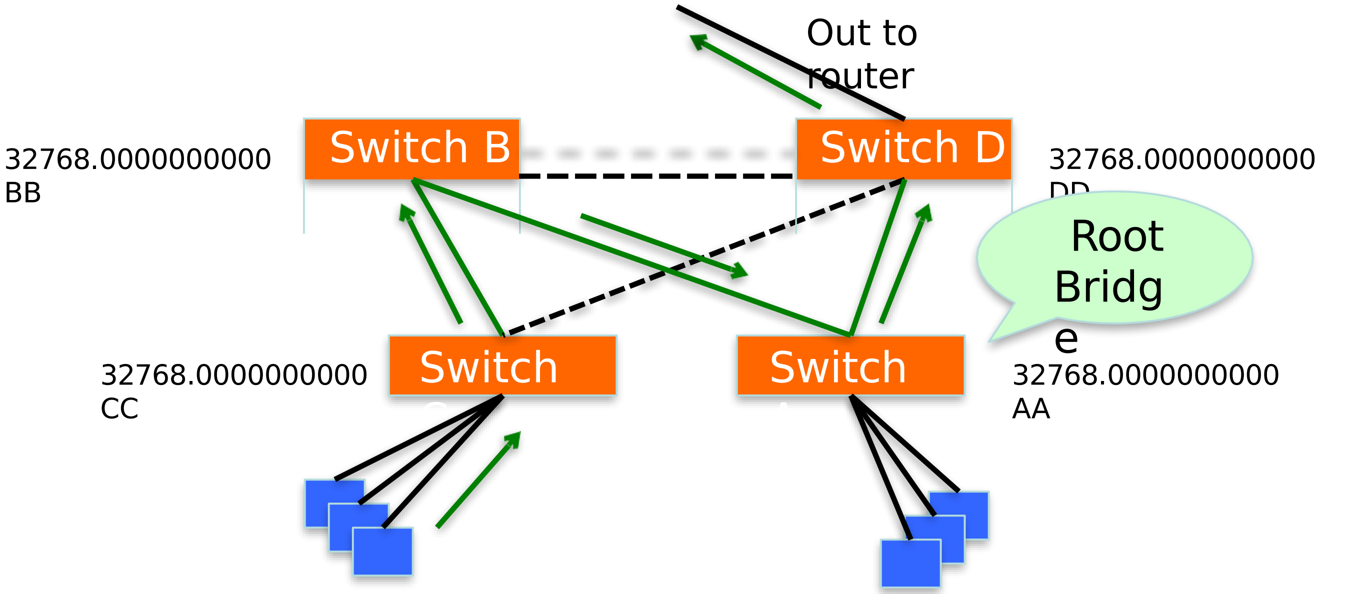
STP Topology Changes

- Switches will recalculate if:
  - A new switch is introduced
    - It could be the new Root Bridge!
  - A switch fails
  - A link fails
  - A link that failed comes back online
  - A new link is introduced

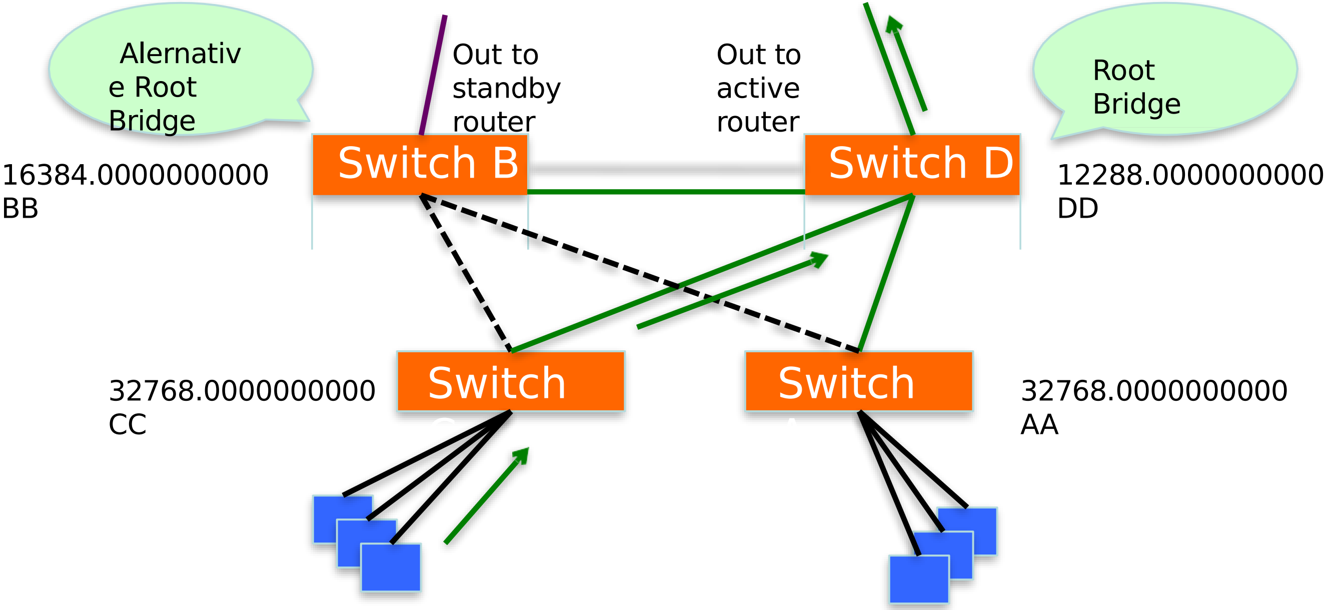
Root Bridge Placement

- Using default STP parameters might result in an undesired situation
  - Traffic will flow in non-optimal ways
  - An unstable or slow switch might become the root
- You need to plan your assignment of bridge priorities carefully

Bad Root Bridge Placement



Good Root Bridge Placement



Protecting the STP Topology

- Some vendors have included features that protect the STP topology:
  - Root Guard
  - BPDU Guard
  - Loop Guard
  - UDLD
  - Etc.
- These features will be discussed later

STP Design Guidelines

- Enable spanning tree even if you don’t have
- redundant paths Always plan and set bridge priorities
  - Make the root choice deterministic
  - Include an alternative root bridge
- If possible, do not accept BPDUs on end user ports
  - Apply BPDU Guard or similar where available

Bridge Priorities: Example Strategy

Priority	Description	Notes
0	Core Switch	
4096	Redundant Core Switch	For cases where there is a second core switch
8192	Reserved	
12288	Building Distribution Switch	
16384	Redundant Building Distribution Switch	For cases where buildings have redundant distribution switches
20480	Spare	
24576	Building Access Switch	
28672	Building Access Switch (Daisy Chain)	In rare cases where access devices have to be daisy-chained
32768	Default	No managed devices should have this priority

802.1d Convergence Speeds

- Moving from the Blocking state to the Forwarding State takes at least 2 x Forward Delay time units (~ 30 secs.)
  - This can be annoying when connecting end user stations
- Some vendors have added enhancements such as PortFast, which will reduce this time to a minimum for edge ports
- – Never use PortFast or similar in switch-to-switch links

Topology changes typically take 30 seconds too
  - This can be unacceptable in a production network

Rapid Spanning Tree (802.1w)

- Backwards-compatible with
- 802.1d Provides faster
- convergence Configure which ports are edge ports
  - i.e. for end users, not connections to other switches

Multiple Spanning Tree (802.1s)

- Again, backwards-compatible
- Includes the fast convergence from RSTP
- Also lets you configure multiple trees (with different roots) for different groups of VLANs
  - So that load is shared between links
  - Usually not worth the complexity

Configuration: Cisco

- STP enabled by default (PVST)
- Select standards-based STP *(recommended!)*
- `spanning-tree mode mst`

Set bridge priority:

- `spanning-tree mst 0 priority` ***12288***

For old switches which can only do PVST:

- `spanning-tree vlan 1 priority` ***12288***
  - *Repeat for all vlans!*

To enable portfast feature on all access ports:

`spanning-tree portfast default`

Configuration: HP

- Must enable STP explicitly!!

```
spanning-tree
```

- Set bridge priority:

```
spanning-tree priority 3
```

  - *Actual priority is 3 x 4096 = 12288*

- Disable portfast feature on each trunk port:

```
no spanning-tree <port> auto-edge-port
```

Configuration: Juniper

- Enable multiple spanning tree and set the bridge priority

```
set protocols mstp bridge-priority 12k
```

- – *(12k = 12x1024 = 12288)*

- Configure which ports participate in multiple spanning tree

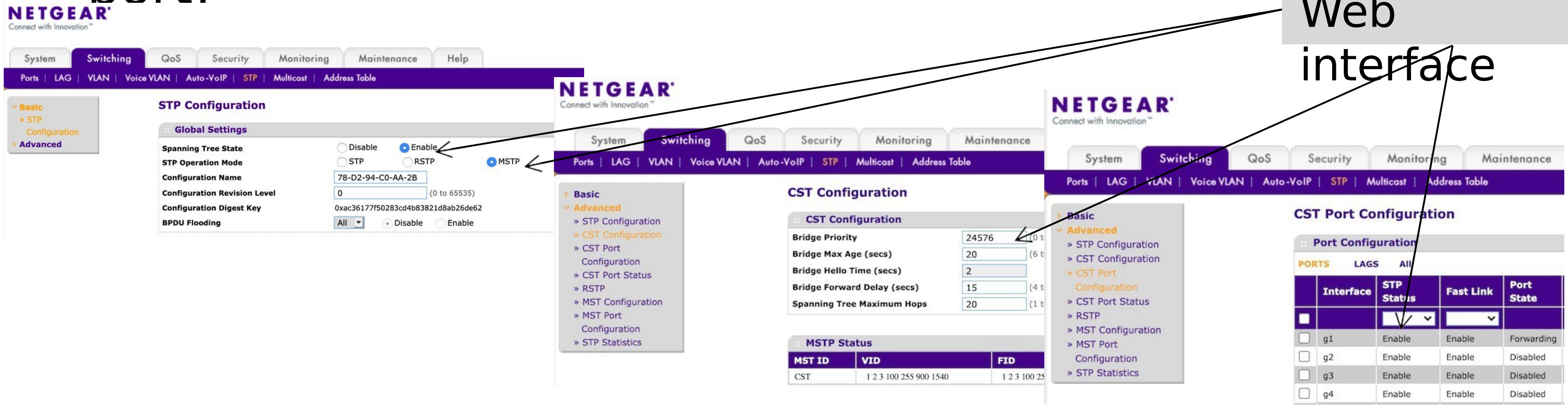
Alternatively for a group of ports:

```
wildcard range set protocols mstp interface ge-0/0/[0-3]
```

Configuration: Netgear

- Must enable STP globally, *and* on each port!

Web interface



Command line

```
spanning-tree
spanning-tree mst priority 0 24576
interface 0/1
  spanning-tree port mode
```

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Questions
?

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center