Campus Networks: Layer 2 switching

VLANs / STP / LACP / LLDP / Security

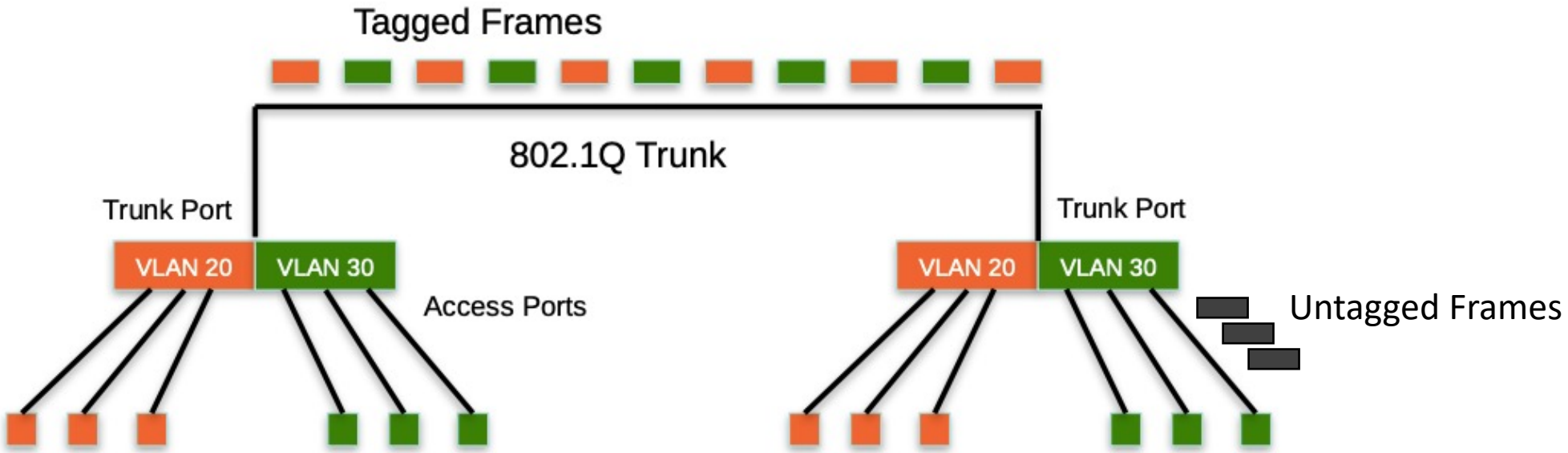
Author:
Sami Ait Ali Oulahcen

Nouakchott, Mauritania
17-22 February 2025

Intro

- A switch maps MAC addresses to ports and stores this information in the mac address table (aka forwarding table)
- A switch learns MAC addresses and corresponding ports from received frames
- A switch broadcasts the frame if MAC address not in forwarding table
- By default all switch interfaces connected to VLAN1

```
show mac address-table
```


Tagged Frames
802.1Q Trunk
Trunk Port
Trunk Port
VLAN 20
VLAN 30
VLAN 20
VLAN 30
Access Ports
Untagged Frames

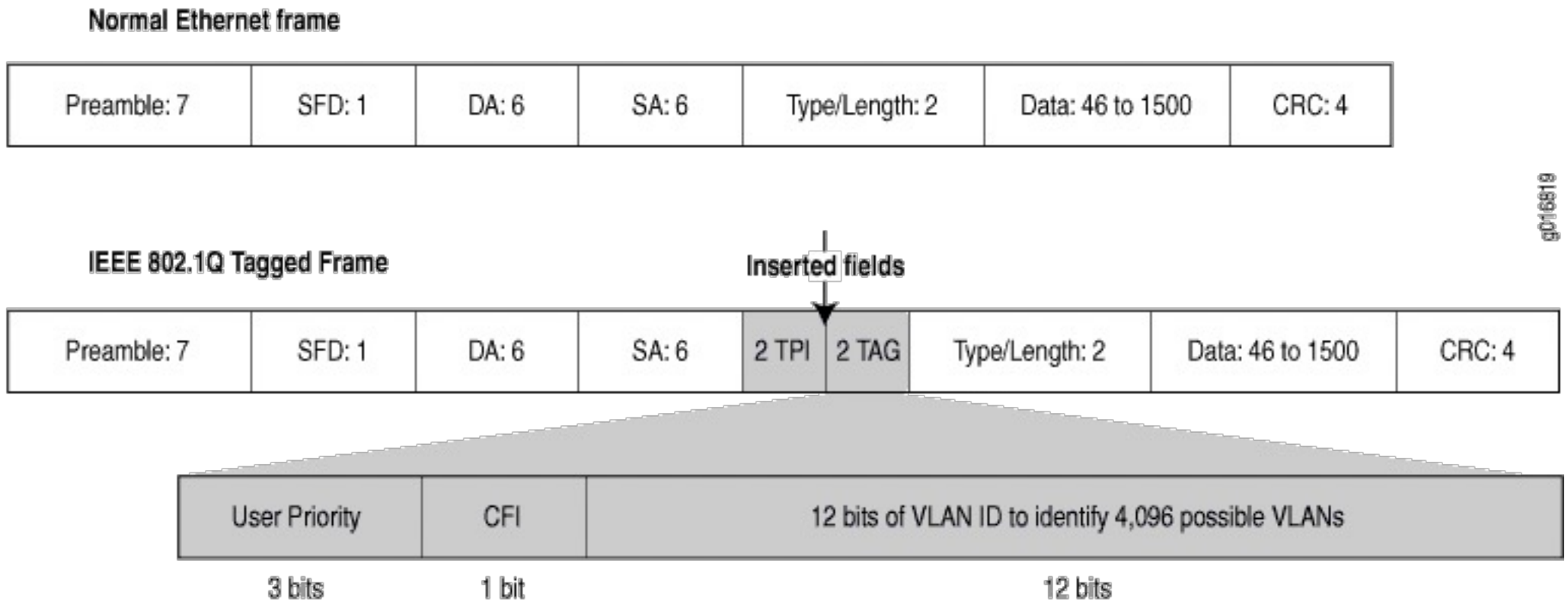
VLANs

VLANs

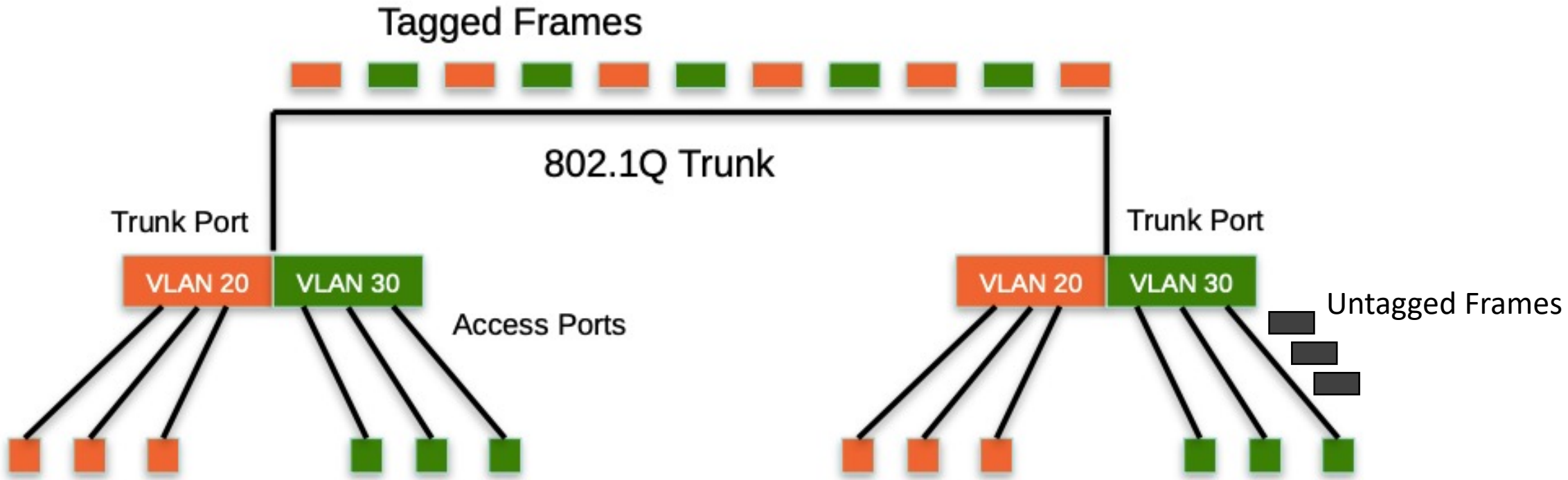
- Allow us to segregate the network into many Virtual LANs
- Only members of a VLAN can see that VLAN’s traffic
- Access ports for members of a single VLAN (untagged traffic)
- Trunk ports for carrying multiple VLANs (tagged traffic)
- Limits the broadcasting domain, frames only broadcasted inside the VLAN they belong to

VLAN tagged vs untagged frames

- IEEE standard 802.1q defines how ethernet frames should be tagged



VLAN tagged vs untagged frames (illustration)



VLAN tagged vs untagged frames (native VLAN)

- A trunk port can have one and only one untagged VLAN, this is called a native VLAN

```
interface GigabitEthernet1/0/1
 description Trunk Port to Switch2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport trunk native vlan 99
 switchport trunk allowed vlan 10,20,30,99
 no shutdown
```

Inter-VLAN routing

- All traffic between VLANs must be routed:
  - through a router
  - OR a layer 3 switch

Configuration example

<> bash

```
vlan 10
 name Building 1, ground floor
exit

vlan 11
 name Building 1, 1st floor
exit

interface GigabitEthernet0/1
 switchport mode access
 switchport access vlan 10
exit

interface GigabitEthernet0/2
 switchport mode access
 switchport access vlan 11
exit
```

Arista/Cisco/Dell/SONiC

<> bash

```
/interface bridge
add name=bridge1 protocol-mode=none

/interface vlan
add interface=bridge1 name=vlan10 vlan-id=10

/interface vlan
add interface=bridge1 name=vlan11 vlan-id=11

/interface bridge port
add bridge=bridge1 interface=ether1 pvid=10
add bridge=bridge1 interface=ether2 pvid=11
```

Mikrotik

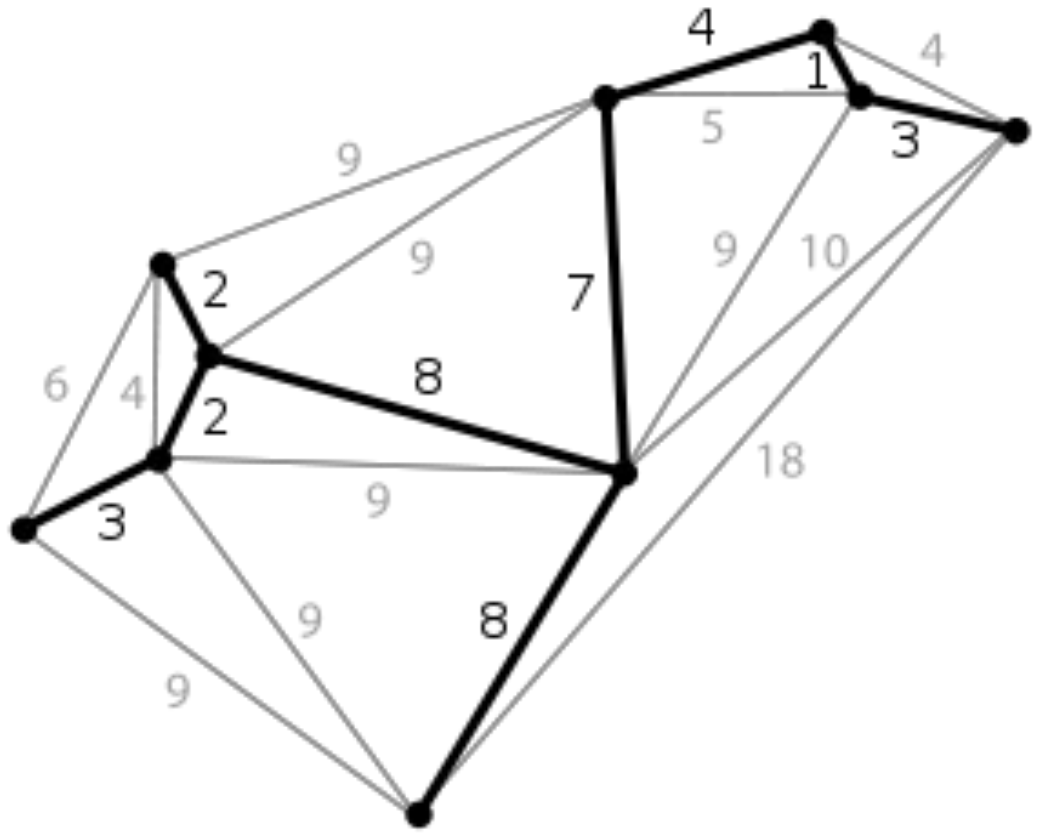
<> bash

```
vlan 10
 name Building 1, ground floor
exit

vlan 11
 name Building 1, 1st floor
exit

interface ethernet-1/1
 vlan access 10
exit

interface ethernet-1/2
 vlan access 11
exit
```

Nokia

Recommendations for VLANs

- Avoid using VLAN1
- Avoid using proprietary protocols like VTP or PVST
- Create only necessary VLANs on each switch
- Avoid using native VLAN unless necessary
- Restrict access to management VLAN



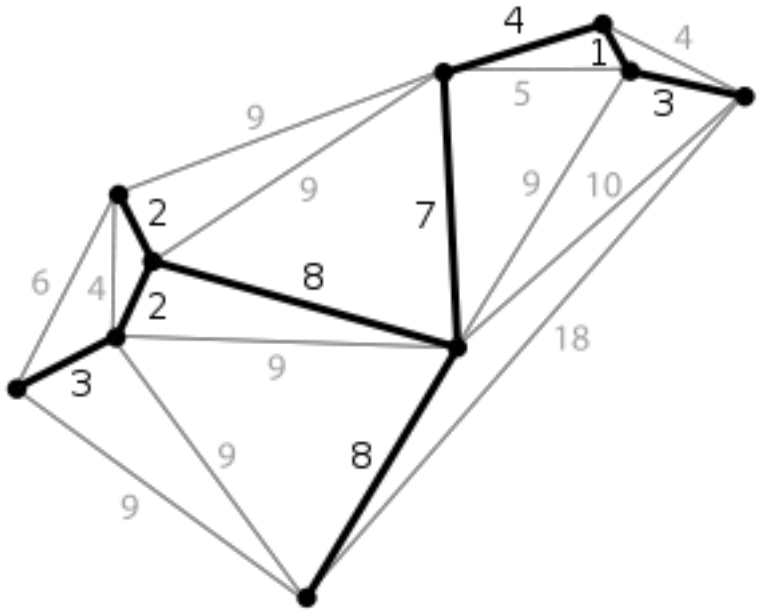
Spanning Tree Protocol

Switching loops

- When there is more than one path between 2 switches
- Good for redundancy / Bad for the forwarding table
- MAC address seen coming from different ports
- Each switch re-Broadcasts their own broadcast => **broadcast storm**
- All available bandwidth is used and network is congested

Switching loops

- Enter 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”.



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
- 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)
- We sometimes say Spanning Tree or STP to refer to any flavor of STP

How Spanning Tree works (1)

- **Root bridge** selection

  - smallest (root bridge id + mac address) is selected

- **Root port** selection for each switch

  - the port with the lowest **root path cost**

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

How Spanning Tree works (2)

- **Designated Port** selection per network segment

  - Each network segment needs to have only one switch that forwards traffic to solve the loop

  - The following sequence of conditions is followed for selection:

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

When a port is neither a Root Port nor a Designated Port is goes into the **Blocking state**

Spanning Tree states

- Blocking:
  - Receiving BPDUs
- Listening
  - Sending and receiving BPDUs
- Learning:
  - Sending and receiving BPDUs
  - Learning new MAC addresses
- Forwarding:
  - Sending and receiving BPDUs
  - Learning new MAC addresses
  - Forwarding frames

Spanning Tree topology changes

- Switches will recalculate Spanning Tree topology if:
  - A new switch is introduced
  - A switch fails
  - A link fails
  - A link that failed comes back online
  - A new link is introduced

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

Configuration examples

<> bash

```
enable
configure terminal
spanning-tree mst configuration
 name RIMER-LAB
 revision 1
 instance 1 vlan 12, 15, 20
 instance 1 priority 12288
exit
spanning-tree mode mst
end
write memory
```

Nokia/Cisco/Arista/Dell

<> bash

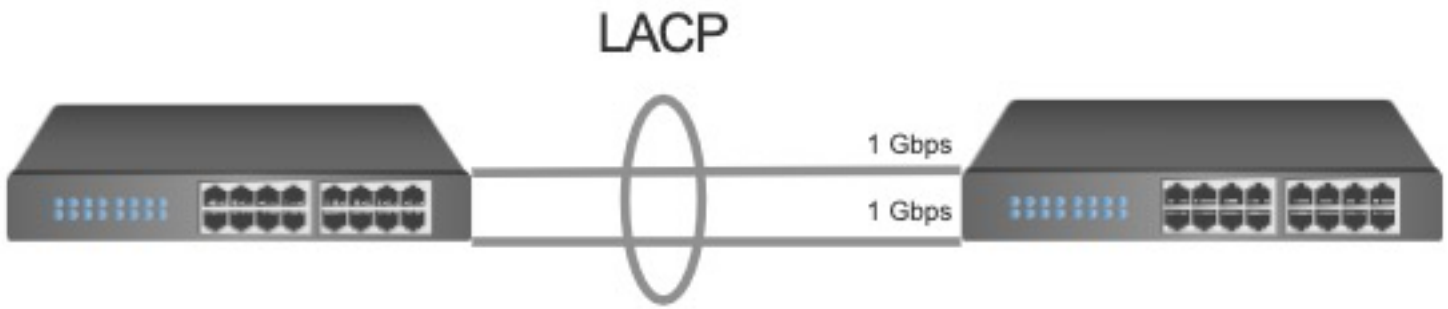
```
# Enter the configuration mode
sudo config terminal

# Configure MSTP
spanning-tree mst configuration
 name RIMER-LAB
 revision 1
 instance 1 vlan 12, 15, 20
 instance 1 priority 12288
exit

# Enable MSTP globally
spanning-tree mode mst

# Save the configuration
write memory
```

SONiC

<> bash

```
/interface bridge
add name=bridge1 protocol-mode=mstp
/interface bridge port
add bridge=bridge1 interface=ether1
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
/interface bridge mstp
set bridge=bridge1 name=RIMER-LAB revision-id=1
/interface bridge mstp instance
add bridge=bridge1 instance=1 vlan=12,15,20 prio
```

Mikrotik



Link Aggregation (LACP)

Link Aggregation

- Also known as port bundling, link bundling
- Uses multiple links in parallel as a single, logical link:
    - For increased capacity
    - For redundancy (fault tolerance)
- The current standard is LACP or Link Aggregation Control Protocol (802.1AX)

LACP Operation

- Two devices connected via multiple links will send LACPDU packets, identifying themselves and the port capabilities
- Then automatically build the logical aggregated links, and pass traffic
- Ports configured as active or passive
- Bundled links distribute frames using a hashing algorithm, based on:
    - Source and/or Destination MAC address
    - Source and/or Destination IP address
    - Source and/or Destination Port numbers

Configuration examples

Configuration on Nokia/Cisco switch:

```
! Configure LACP port-channel
interface Port-channel70
 switchport
 switchport access vlan 160
 switchport mode access
! Add an interface to LACP
interface TenGigabitEthernet5/6
 switchport access vlan 160
 switchport mode access
 channel-group 70 mode active
 spanning-tree portfast edge
! Add other interfaces as needed
```

Configuration on Linux Network Manager:

```
# Add and configure LACP interface
nmcli connection add type bond con-
name bond0 ifname bond0 bond.options
'mode=4,miimon=100,lacp_rate=fast,xmit
_hash_policy=layer2+3'
nmcli con mod id bond0 ipv6.addresses
'2001:db8:f1::71/64' ipv6.gateway
'2001:db8:f1::4' ipv6.dns
'2001:db8:f1:d::160,2001:db8:f1:d::170
' ipv6.method manual
nmcli conn up bond0
# Add interface ens1f0np0 to LACP
nmcli con add type ether slave-type
bond con-name bond0-eth0 ifname
ens1f0np0 master bond0
# Add other interfaces as needed then
restart NetworkManager
systemctl restart NetworkManager
```

Linux bond mode 4 is 802.3ad (lacp)

LLDP

Link Layer Discovery Protocol

LLDP

- LLDP or Link Layer Discovery Protocol (802.1 AB) is a layer 2 standard used by network devices for advertising their identity, capabilities and neighbors on a LAN segment
- Can be used to quickly identify directly connected equipment
- Implementation on Cisco is called **CDP**
- Enable on trunk ports
- Disable on access ports

Layer 2 Network Security

Root Guard / BPDU guard / DHCP snooping / RA guard

Problem 1: rogue switches

- A colleague introduces a small unmanaged switch to the network
- The rogue switch participates in spanning tree and has a lower bridge ID + MAC address than the actual root bridge
- The rogue switch becomes the root bridge and legitimate switches start to reconfigure their paths
- Traffic starts getting forwarded through the rogue switch which leads to **congestion, increased latency and suboptimal traffic paths**

Solutions to rogue switches

- Enable “**Root Guard**” on edge ports

  - Switch can still be plugged in, and ***can*** participate in STP

  - However if it ever tries to become root, the port is shut down

  - Error condition must be cleared manually, unless you configure automatic recovery (errdisable-timeout)

- An other solution would be to enable “**BPDU Guard**” on edge ports

  - Switch can still be plugged in, but it ***can’t*** participate in STP

  - If any spanning tree BPDU at all is received on this port, the port is immediately shut down

Problem 2: rogue DHCP servers

- A colleague introduces a small Wi-Fi router/repeater to get wireless inside their office
- The rogue device has an embedded DHCP server that starts serving IP addresses and network configuration settings to devices on the same VLAN
- This leads to IP address conflicts and/or incorrect network configuration settings, such as the wrong default gateway or DNS server addresses

Solutions to rogue DHCP servers

- Enable **’’DHCP snooping’’** on switches to filter DHCP messages
- This feature allows only trusted DHCP servers to respond to DHCP requests, preventing rogue servers from providing IP addresses and network configuration
- Enable **’’RA guard’’** on switches to filter Router Advertisements
- This feature allows specifying which ports on a switch are trusted to send Router Advertisement messages, preventing rogue devices from sending incorrect network configuration like default IPv6 gateways or DNS servers

Acknowledgment

This work is inspired from the NSRC campus networks workshop available here: https://nsrc.org/activities/agendas/en/cndo/

Questions ?