Campus Networking Workshop

Layer-2 Network Design
Layer 2 Concepts

Layer 2 protocols basically control access to a shared medium (copper, fiber, electro-magnetic waves)

Ethernet is the de-facto standard today

Reasons:
- Simple
- Cheap
- Manufacturers keep making it faster
Ethernet Functions

Source and Destination identification
MAC addresses

Detect and avoid frame collisions
Listen and wait for channel to be available
If collision occurs, wait a random period before retrying

This is called CASMA-CD: Carrier Sense Multiple Access with Collision Detection
Switched Star Topology Benefits

It’s modular:

- Independent wires for each end node
- Independent traffic in each wire
- A second layer of switches can be added to build a hierarchical network that extends the same two benefits above

ALWAYS DESIGN WITH MODULARITY IN MIND
A frame sent by one node is always sent to every other node. Hubs are also called “repeaters” because they just “repeat” what they hear.
Switch

*Learns* the location of each node by looking at the source address of each incoming frame, and builds a *forwarding table*

*Forwards* each incoming frame only to the port where the destination node is

- Reduces the collision domain
- Makes more efficient use of the wire
- Nodes don’t waste time checking frames not destined to them
Switch

Forwarding Table

<table>
<thead>
<tr>
<th>Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAAAAAAAAAA</td>
<td>1</td>
</tr>
<tr>
<td>BBBBBBBBBBBBBBB</td>
<td>5</td>
</tr>
</tbody>
</table>
Switches and Broadcast

A switch broadcasts some frames:

- When the destination address is not found in the table
- When the frame is destined to the broadcast address (FF:FF:FF:FF:FF:FF)
- When the frame is destined to a multicast ethernet address

So, switches do not reduce the broadcast domain!
Switch vs. Router

Routers more or less do with IP packets what switches do with Ethernet frames

A router looks at the IP packet destination and checks its *routing table* to decide where to forward the packet

Some differences:

- IP packets travel inside ethernet frames
- IP networks can be logically segmented into *subnets*
- Switches do not usually know about IP, they only deal with Ethernet frames
## Switch vs. Router

<table>
<thead>
<tr>
<th>SWITCH</th>
<th>ROUTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer-2 device</td>
<td>Layer-3 device</td>
</tr>
<tr>
<td>Uses MAC addresses</td>
<td>Uses IP Addresses</td>
</tr>
<tr>
<td>Passes Ethernet Frames</td>
<td>Passes IP Packets</td>
</tr>
</tbody>
</table>
Switch vs. Router

Routers do not forward Ethernet broadcasts. So:

- Switches reduce the collision domain
- Routers reduce the broadcast domain

This becomes really important when trying to design hierarchical, scalable networks that can grow sustainably.
Traffic Domains

Broadcast Domain

Collision Domain

Router

Switch

Hub

Hub

Hub

Hub
Traffic Domains

Try to eliminate collision domains
Get rid of hubs!

Try to keep your broadcast domain limited to no more than 250 simultaneously connected hosts
Segment your network using routers
Layer 2 Network Design Guidelines

Always connect hierarchically

If there are multiple switches in a building, use an aggregation switch

Locate the aggregation switch close to the building entry point (e.g. fiber panel)

Locate edge switches close to users (e.g. one per floor)

Max length for Cat5 is 100 meters
Building Network

Level 4
Level 3
Level 2
Level 1
Connect buildings hierarchically
Switching Architectures

Any Questions?
Virtual LANs (VLANs)

Allow us to split switches into separate (virtual) switches

Only members of a VLAN can see that VLAN’s traffic

Inter-vlan traffic must go through a router
Local VLANs

2 VLANs or more within a single switch

*Edge ports*, where end nodes are connected, are configured as members of a VLAN

The switch behaves as several virtual switches, sending traffic only within VLAN members
Local VLANs

- Switch
- VLAN X
- VLAN Y
- Edge ports
- VLAN X nodes
- VLAN Y nodes
VLANs across switches

Two switches can exchange traffic from one or more VLANs.

Inter-switch links are configured as *trunks*, carrying frames from all or a subset of a switch’s VLANs.

Each frame carries a *tag* that identifies which VLAN it belongs to.
802.1Q

The IEEE standard that defines how ethernet frames should be tagged when moving across switch trunks.

This means that switches from different vendors are able to exchange VLAN traffic.
802.1Q tagged frame

Normal Ethernet frame

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>7</td>
</tr>
<tr>
<td>SFD</td>
<td>1</td>
</tr>
<tr>
<td>DA</td>
<td>6</td>
</tr>
<tr>
<td>SA</td>
<td>6</td>
</tr>
<tr>
<td>Type/Length</td>
<td>2</td>
</tr>
<tr>
<td>Data</td>
<td>46 to 1500</td>
</tr>
<tr>
<td>CRC</td>
<td>4</td>
</tr>
</tbody>
</table>

IEEE 802.1Q Tagged Frame

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>7</td>
</tr>
<tr>
<td>SFD</td>
<td>1</td>
</tr>
<tr>
<td>DA</td>
<td>6</td>
</tr>
<tr>
<td>SA</td>
<td>6</td>
</tr>
<tr>
<td>2 TPI</td>
<td></td>
</tr>
<tr>
<td>2 TAG</td>
<td></td>
</tr>
<tr>
<td>Type/Length</td>
<td>2</td>
</tr>
<tr>
<td>Data</td>
<td>46 to 1500</td>
</tr>
<tr>
<td>CRC</td>
<td>4</td>
</tr>
</tbody>
</table>

Inserted fields

- User Priority: 3 bits
- CFI: 1 bit
- 12 bits of VLAN ID to identify 4,096 possible VLANs: 12 bits
VLANs across switches

Tagged Frames

802.1Q Trunk

VLAN X

VLAN Y

Trunk Port

Edge Ports

This is called “VLAN Trunking”
Tagged vs. Untagged

Edge ports are not tagged, they are just “members” of a VLAN.

You only need to tag frames in switch-to-switch links (trunks), when transporting multiple VLANs.

A trunk can transport both tagged and untagged VLANs.

As long as the two switches agree on how to handle those.
VLANs increase complexity

You can no longer “just replace” a switch
Now you have VLAN configuration to maintain
Field technicians need more skills
You have to make sure that all the switch-to-switch trunks are carrying all the necessary VLANs
Need to keep in mind when adding/removing VLANs
Good reasons to use VLANs

You want to segment your network into multiple subnets, but can’t buy enough switches

Hide sensitive infrastructure like IP phones, building controls, etc.

Separate control traffic from user traffic

Restrict who can access your switch management address
Bad reasons to use VLANs

Because you can, and you feel cool 😊
Because they will completely secure your hosts (or so you think)
Because they allow you to extend the same IP network over multiple separate buildings
  This is actually very common, but a bad idea
Do not build “VLAN spaghetti”

Extending a VLAN to multiple buildings across trunk ports
Bad idea because:
  Broadcast traffic is carried across all trunks from one end of the network to another
  Broadcast storm can spread across the extent of the VLAN, and affect all VLANS!
  Maintenance and troubleshooting nightmare
VLANs

Any Questions?
Link Aggregation

Also known as *port bundling, link bundling*

You can use multiple links in parallel as a single, logical link

- For increased capacity
- For redundancy (fault tolerance)

LACP (Link Aggregation Control Protocol) is a standardized method of negotiating these bundled links between switches
LACP Operation

Two switches connected via multiple links will send LACPDU packets, identifying themselves and the port capabilities. They will then automatically build the logical aggregated links, and then pass traffic.

Switch ports can be configured as active or passive.
LACP Operation

- Switches A and B are connected to each other using two sets of Fast Ethernet ports
- LACP is enabled and the ports are turned on
- Switches start sending LACPDUs, then negotiate how to set up the aggregation
LACP Operation

- The result is an aggregated 200 Mbps logical link

- The link is also fault tolerant: If one of the member links fail, LACP will automatically take that link off the bundle, and keep sending traffic over the remaining link
Link Aggregation

Any Questions?
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
- Switches will broadcast each other’s broadcasts
  - All available bandwidth is utilized
  - Switch processors cannot handle the load
Switching Loop

- Node1 sends a broadcast frame (e.g. an ARP request)
Switching Loop

- Switches A, B and C broadcast node 1’s frame out every port
Switching Loop

• But they receive each other’s broadcasts, which they need to forward again and out every port!

• The broadcasts are amplified, creating a broadcast storm
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.
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 flavors:

- Traditional Spanning Tree (802.1d)
- Rapid Spanning Tree or RSTP (802.1w)
- Multiple Spanning Tree or MSTP (802.1s)
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).

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

The election process is based on the Bridge ID, which is composed of:

- The Bridge Priority: A two-byte value that is configurable
- The MAC address: A unique, hardcoded address that cannot be changed.
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 **lower** 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.
- Who 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**. The key is to find the port with the **lowest 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
  e.g. The faster the link, the lower the cost

<table>
<thead>
<tr>
<th>Link Speed</th>
<th>STP Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Mbps</td>
<td>100</td>
</tr>
<tr>
<td>100 Mbps</td>
<td>19</td>
</tr>
<tr>
<td>1 Gbps</td>
<td>4</td>
</tr>
<tr>
<td>10 Gbps</td>
<td>2</td>
</tr>
</tbody>
</table>
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
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 neighbor’s down the line keep adding in the same fashion
Root Port Selection (802.1d)

On each switch, the port where the lowest Root Path Cost was received 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)

Switch A

Switch B

Switch C

Root Port

Root Port

32678.00000000000AA

Cost=19

32678.00000000000BB

Cost=19

32678.00000000000CC

Cost=19
Electing Designated Ports (802.1d)

OK, we now have selected root ports but we haven’t solved the loop problem yet, have we?

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 link.

The one with the lowest cumulative Root Path Cost to the Root Bridge.
Electing Designated Ports (802.1d)

Which port should be the Designated Port on each segment?

Switch B

Switch A

Switch C

Cost: 19

32678.000000000000AA

32678.0000000000BB

Cost: 19

32678.0000000000CC
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:

- Lowest Root Bridge ID
- Lowest Root Path Cost to Root Bridge
- Lowest Sender Bridge ID
- Lowest Sender Port ID
In the B-C link, Switch B has the lowest Bridge ID, so port 2 in Switch B is the Designated Port.
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
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
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

Switch B → Switch C → Switch A

Switch D

Out to router

Root Bridge

32678.0000000000BB

32678.0000000000CC

32678.0000000000DD

32678.0000000000AA
Good Root Bridge Placement

Switch B

Switch C

Switch D

Switch A

Alternative Root Bridge

Out to standby router

Out to active router

Root Bridge

1.0000000000BB

0.0000000000DD

32678.0000000000CC

32678.0000000000AA

Network Startup Resource Center
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
802.1d Convergence Speeds

Moving from the Blocking state to the Forwarding State takes at least $2 \times \text{Forward Delay}$ time units (~ 30 secs.)

This can be annoying when connecting end user stations

Some vendors have added enhancements such as \textit{PortFast}, which will reduce this time to a minimum for edge ports

Never use \textit{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)

Convergence is much faster
Communication between switches is more interactive

Edge ports don’t participate
Edge ports transition to forwarding state immediately

If BPDUs are received on an edge port, it becomes a non-edge port to prevent loops
Questions?

Thank you.