NETWORKING ZERO TO HERO

PURE L2 BRIDGE LAB

Flooding, forwarding, and MAC learning on a Linux bridge.

ENVIRONMENT	PRIVILEGE	DURATION	RISK
WSL2 Ubuntu recommended	sudo required	25 minutes	No IP addresses assigned

BEFORE YOU START

Build a switch you can interrogate.

This lab creates a Linux bridge with three ports. Host A and Host B inject frames. Port C only listens. If Port C sees a frame, the bridge flooded it. If Port C is silent, the bridge forwarded directly.

RECOMMENDED

WSL2 Ubuntu on Windows. This is the tested path for the course.

LINUX

Works on a normal Linux machine with sudo, iproute2, Scapy and tcpdump.

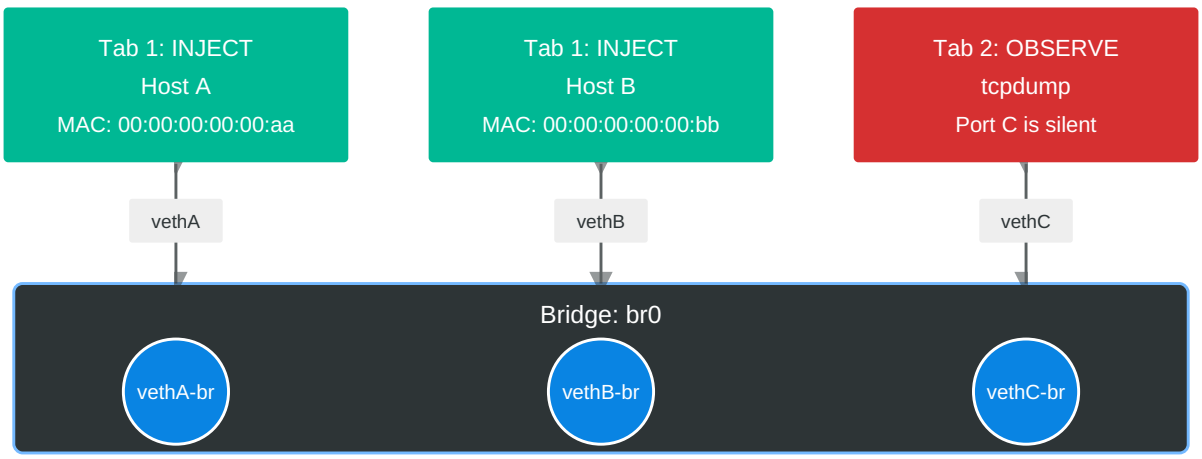
MAC

Use an Ubuntu VM, cloud Linux host, or lab Linux box. Native macOS Terminal will not run these `ip link` bridge and veth commands.

Start here: Open three terminals before you run commands. Keep Tab 2 visible. It is your witness port. Do not assign IP addresses in this lab.

ENVIRONMENT	PRIVILEGE	DURATION	RISK
WSL2 Ubuntu, standard install	Root access with sudo	25 minutes	None. No IP addresses assigned.

Topology



Terminal setup

Open three terminals before beginning. On Windows, use three WSL tabs in the same distribution, for example `wsl -d Ubuntu`. On Linux or an Ubuntu VM, use three normal terminal windows.

TAB	NAME	PURPOSE	PROMPT
1	**INJECT**	Scapy packet sender	>>>
2	**OBSERVE**	tcpdump monitor on Port C	$
3	**INSPECT**	Bridge table management	$

> **Crucial:** Port C never transmits. Tab 2 is your only source of truth for whether frames are flooding or being forwarded.

LAB STEPS

1. Install dependencies

Run this in **Tab 3: INSPECT**.

BASH

```
sudo apt update
sudo apt install python3-scapy tcpdump -y
```

2. Create the bridge

Run this in **Tab 3: INSPECT**.

BASH

```
# Create switch
sudo ip link add br0 type bridge

# Set aging to 1 hour so the table stays populated for the demo
sudo ip link set br0 type bridge ageing_time 360000
sudo ip link set br0 up

# Create and attach ports
for port in A B C; do
  sudo ip link add veth${port} type veth peer name veth${port}-br
  sudo ip link set veth${port}-br master br0

  # Disable IPv6 to keep the MAC table clean
  sudo sysctl -qw net.ipv6.conf.veth${port}.disable_ipv6=1
  sudo sysctl -qw net.ipv6.conf.veth${port}-br.disable_ipv6=1

  sudo ip link set veth${port}-br up
  sudo ip link set veth${port} up
done
```

LAB STEPS

3. Add a MAC table helper

Run this in **Tab 3: INSPECT**.

BASH

```
macs() {
  echo "--- MAC TABLE ---"
  sudo bridge fdb show br br0 | grep -v -e permanent -e self || echo "  (empty)"
}
```

> Run `macs` now to verify the table is empty.

4. Start monitoring

Run this in **Tab 2: OBSERVE**. Leave it running.

BASH

```
sudo tcpdump -i vethC -e -n -l
```

5. Launch Scapy

Run this in **Tab 1: INJECT**.

BASH

```
sudo scapy
```

> **What you are watching:** Port C is the witness. Broadcasts and unknown unicasts should appear there. Known unicasts should not.

DEMONSTRATION SCENARIOS

Demo 1. Broadcast flood

Run this in **Tab 1: INJECT**.

PYTHON

```
frame = Ether(src="00:00:00:00:00:aa", dst="ff:ff:ff:ff:ff:ff") / "Discovery"
sendp(frame, iface="vethA")
```

RESULT
Port C sees the frame.

REASON
Broadcasts always flood to all ports.

Demo 2. Known unicast forwarding

Run this in **Tab 1: INJECT**.

PYTHON

```
reply = Ether(src="00:00:00:00:00:bb", dst="00:00:00:00:00:aa") / "Direct Reply"
sendp(reply, iface="vethB")
```

RESULT
Port C is silent.

REASON
The bridge knows where Host A is, learned in Demo 1, so it forwards directly to Port A.

> **Learning goal:** a switch does not need IP addresses to learn where Ethernet devices live.

DEMONSTRATION SCENARIOS

Demo 3. Unknown unicast flood

Run this in **Tab 1: INJECT**.

PYTHON

```
unknown = Ether(src="00:00:00:00:00:aa", dst="00:00:00:00:00:dd") / "Who is DD?"
sendp(unknown, iface="vethA")
```

RESULT

Port C sees the frame.

REASON

The bridge floods because the destination MAC is unknown.

Demo 4. Memory flush

Run this in **Tab 3: INSPECT**.

BASH

```
sudo ip link set dev br0 type bridge fdb_flush
```

RESULT

Re-run Demo 2. Port C now sees the frame.

REASON

The bridge forgot the location of Host A.

Teardown

Run this in **Tab 3: INSPECT** when you are finished.

BASH

```
for port in A B C; do
  sudo ip link delete veth${port}
done
sudo ip link delete br0
```

Course: missioninstituteoftechnology.com/courses/networking-zero-to-hero.html
Website: missioninstituteoftechnology.com