VLAN Lab Exercise 3: Experimental Plan for IEEE 802.1Q Network Segmentation - Group 1A-4

What is VLAN?

A **Virtual Local Area Network (VLAN)** is a technology that allows you to create multiple logical networks within a single physical network infrastructure. VLANs segment a network at Layer 2 (Data Link Layer), creating separate broadcast domains. This means devices in different VLANs cannot communicate directly, even if connected to the same physical switch, unless routing is configured between them.

Network Topology Identification

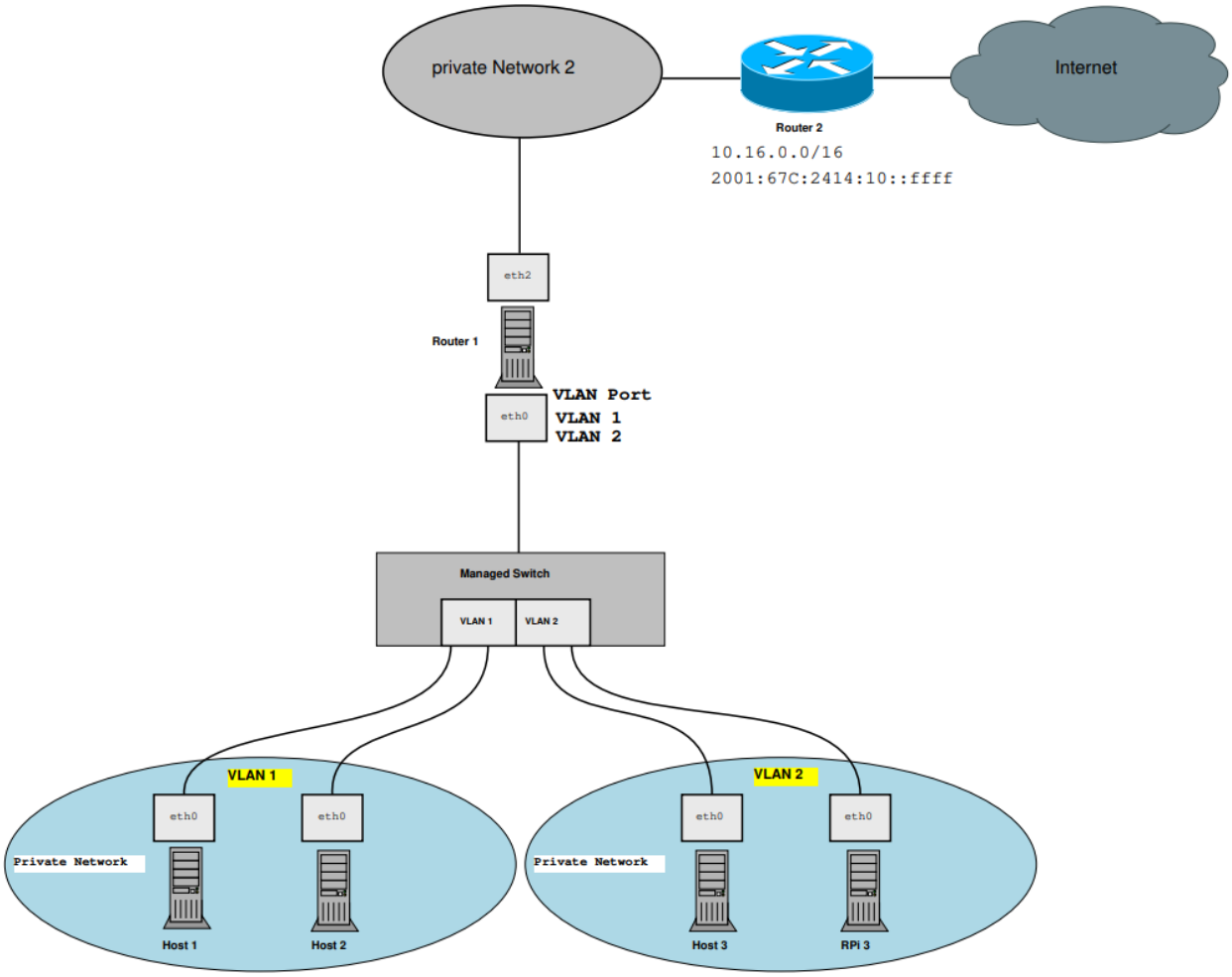

Figure 1: Network Topology of lab exercise 3

***Note**: The interfaces in the Lab shall be: eth0 = enp1s0f0 and eth2 = enp1s0f1*

VLAN Implementation Methods

Feature	Tagged (802.1Q)	Untagged (Access)
Frame Format	Inserts a 4-byte 802.1Q header carrying the VLAN ID and priority bits.	Sends/receives plain Ethernet frames with no VLAN header.

Feature	Tagged (802.1Q)	Untagged (Access)
Typical Use Case	Trunk links (Switch↔Switch, Switch↔Router) carrying many VLANs (e.g. 10, 20, 30) over one cable.	Access links (Switch↔Host) where each port belongs to exactly one VLAN.
Ingress Behavior	Reads the VLAN tag on each incoming frame and forwards into the corresponding VLAN.	Assigns every incoming frame to the port's PVID (no tag expected).
Egress Behavior	Adds the 802.1Q header with the VLAN ID before sending the frame.	Strips any VLAN header and sends a "normal" Ethernet frame.
Advantages	+ Consolidates multiple VLANs on a single link + Essential for larger, multi-switch networks	+ Simple for end-hosts (no VLAN config required) + Zero tagging overhead
Trade-offs	– Slight overhead (4 bytes/frame) – Requires VLAN-aware devices[1] and configuration	– One cable or port per VLAN needed – Not suitable for multi-VLAN links

[1] **VLAN-aware devices**: Network devices (hosts, switches, routers) that understand and can process 802.1Q VLAN tags. These devices must have drivers/firmware supporting VLAN tagging and the ability to create virtual interfaces (e.g., eth0.10 for VLAN 10).

Experimental Setup and Testing

Step 1: Initial Setup

- **Select Router 1 computer**: Choose the Linux machine connected to external Router 2
- **Verify connectivity**: Check enp1s0f1 is connected to Router 2, enp1s0f0 available for switch
- **Enable IPv6**: Configure IPv6 on enp1s0f1 to get Internet access

```
# Enable IPv6
sudo nano /etc/sysctl.conf

# activate the interface
ip link set <interface> up

# process the new IPv6 configuration
sudo sysctl -p
```

Step 2: Choose Implementation Method

Decision: Tagged VLAN with Router 1, since Router 1 is connected to only one port to the switch and should act as a trunk port having access to all the VLANs of the network.

Create VLAN interfaces on Router 1:

```
# Load VLAN module
sudo modprobe 8021q

# Create VLAN 10 and VLAN 20 on enp1s0f0
sudo ip link add link enp1s0f0 name enp1s0f0.10 type vlan id 10
sudo ip link add link enp1s0f0 name enp1s0f0.20 type vlan id 20
```

Start Wireshark capture:

```
sudo wireshark -i enp1s0f0
```

Generate test traffic and observe

First, bring up the VLAN interfaces and assign IPv6 addresses:

```
# Bring up the VLAN interfaces
sudo ip link set enp1s0f0.10 up
sudo ip link set enp1s0f0.20 up

# Assign IPv6 addresses (using ULA for internal network)
sudo ip -6 addr add fd00:1:2:10::1/64 dev enp1s0f0.10
sudo ip -6 addr add fd00:1:2:20::1/64 dev enp1s0f0.20
```

Generate test traffic:

```
# Send IPv6 ping on VLAN 10
ping6 -I enp1s0f0.10 ff02::1

# Send IPv6 ping on VLAN 20
ping6 -I enp1s0f0.20 ff02::1

# Alternative: Use arping for Layer 2 traffic (sends ARP requests at Layer 2)
sudo arping -I enp1s0f0.10 -c 5 fd00:1:2:10::2
```

Note: *arping sends ARP (Address Resolution Protocol) requests at Layer 2, useful for testing VLAN connectivity without relying on Layer 3 (IP) configuration.*

What to observe in Wireshark:

- Filter: `vlan` to see only VLAN-tagged frames
- Look for "802.1Q Virtual LAN" in the frame details
- Check the VLAN ID: should show 10 or 20 depending on which interface you used
- The 802.1Q header adds 4 bytes between the Ethernet header and IP header

Purpose and Practical Benefits

Key Technology Understanding:

- **VLAN Function**: Separates broadcast domains at Layer 2, enabling logical network segmentation
- **802.1Q Tags**: 4-byte header inserted between Ethernet and IP headers (TPID: 0x8100, VLAN ID: 12 bits)
- **Linux Implementation**: Uses 8021q kernel module to create virtual interfaces (e.g., eth0.10)

Testing Tools:

- **Wireshark**: Filter `vlan` to observe 802.1Q headers and VLAN IDs
- **ip command**: Create/manage VLAN interfaces (`ip link add ... type vlan id X`)
- **ping6/arping**: Generate test traffic to verify VLAN isolation

Practical Use Case - Enterprise Network:

Scenario: Company with 100 employees sharing one physical network

- **Without VLAN**: All departments see each other's broadcast traffic, security risk
- **With VLAN**:
  - VLAN 10: Finance (isolated, secure)
  - VLAN 20: Guest WiFi (no access to internal resources)
  - VLAN 30: General office

Benefits demonstrated:

1. Security: Finance traffic invisible to other departments
2. Performance: Reduced broadcast domain size
3. Flexibility: Add/move users by switch port config, not rewiring

Quality Focus - Demonstration Points

(a) How VLAN Works

- **Network Stack**: VLANs operate at Layer 2 (Data Link), adding 802.1Q tag between Ethernet header and payload
- **Broadcast Domain Effect**: Each VLAN creates isolated broadcast domain - broadcasts in VLAN 10 won't reach VLAN 20
- **Demo**: Use Wireshark to show broadcast packets (ff02::1) only visible within same VLAN

(b) Network Division Demo

Objective: Separate Host 1/2 from Host 3/RPi3 using VLANs to demonstrate logical network division on single physical infrastructure.

Configure managed switch via web interface:

Access switch at `http://192.168.0.1` and configure:

- Port to Router 1: Tagged for VLAN 10 & 20 (trunk)
- Ports for Host 1/2: Untagged VLAN 10 (PVID=10)

- Ports for Host 3/RPi3: Untagged VLAN 20 (PVID=20)

Configure hosts with IPv6 addresses:

```
# Host 1 (VLAN 10)
sudo ip -6 addr add 2001:db8:10::11/64 dev eth0
sudo ip -6 route add default via 2001:db8:10::1

# Host 3 (VLAN 20)
sudo ip -6 addr add 2001:db8:20::11/64 dev eth0
sudo ip -6 route add default via 2001:db8:20::1
```

Test and demonstrate separation:

```
# From Host 1 - ping within VLAN 10 (should work)
ping6 2001:db8:10::12  # to Host 2

# From Host 1 - ping across to VLAN 20 (should fail)
ping6 2001:db8:20::11  # to Host 3
```

Observation: Wireshark on trunk shows VLAN tags, while captures on host ports show untagged frames. Ping tests confirm broadcast domain isolation between VLANs.

Here's the improved section (c) with machine locations and explanations:

(c) VLAN Benefits Demonstration

Key Benefits: VLANs provide network segmentation without physical infrastructure changes, unlike physical separation (requires multiple switches/cables) or pure Layer 3 subnetting (no broadcast isolation).

Demonstrate cost efficiency vs physical separation:

On Router 1:

```
# Show that one physical interface serves multiple VLANs
ip link show | grep enp1s0f0
# Output should shows: enp1s0f0, enp1s0f0.10@enp1s0f0, enp1s0f0.20@enp1s0f0
# Meaning: One cable carries both networks (vs needing 2 switches + cables)
```

Demonstrate security vs flat network:

On Host 1 (VLAN 10):

```
# Generate broadcast storm (flood of broadcast packets that can overwhelm a
network)
# This simulates a malfunctioning device or attack
sudo hping3 --flood --rand-source -1 ff02::1 -I enp1s0f0
# Sends rapid broadcast packets to all-nodes IPv6 multicast address
```

Simultaneously on Host 3 (VLAN 20):

```
# Monitor network traffic to prove isolation
sudo tcpdump -i enp1s0f0 -n
# Shows: No packets from the broadcast storm reach VLAN 20
# Proves: VLANs contain network problems/attacks within their boundary
```

Demonstrate flexibility vs static configuration:

On Router 1 (showing the concept):

```
# Check current VLAN membership
bridge vlan show
# Moving a user between VLANs requires only web interface change
# No rewiring, no IP reconfiguration on host needed
```

Wireshark proof: Capture on Router 1's trunk port shows broadcast packets tagged with VLAN 10 only, confirming storm containment and proving VLANs superior isolation vs simple subnetting.

(d) VLAN Use Cases and Alternatives

Practical Use Cases:

1. **Guest Network Isolation**:
    - Benefit: *[Fill: How does VLAN protect internal resources from guests?]*
2. **VoIP Traffic Prioritization**:
    - Benefit: *[Fill: How does VLAN improve call quality?]*
3. **Department Segmentation**:
    - Benefit: *[Fill: Why separate HR/Finance/Engineering?]*
4. **Multi-tenant Environments**:
    - Benefit: *[Fill: How do VLANs help in shared office spaces?]*

Alternative Protocols:

- **VxLAN**: *[Fill: How many VLANs supported? Use case?]*

- **MPLS**: *[Fill: Where is this used instead of VLAN?]*
- **Network Virtualization (SDN)**: *[Fill: How does this differ from VLAN?]*

VLAN Limitations:

- Maximum VLANs: *[Fill: What's the limit and why?]*
- Security vulnerability: *[Fill: What is VLAN hopping?]*
- Geographic limitation: *[Fill: Why can't VLANs span across internet?]*

Purpose of invention: *[Fill: What problem did VLANs solve when invented?]*

(e) Critical Assessment and Recommendations

Where VLANs are well-suited:

- Network size: *[Fill: Ideal number of devices?]*
- Environment type: *[Fill: Campus, enterprise, or data center?]*
- Management complexity: *[Fill: When is VLAN management reasonable?]*

Where VLANs are NOT ideal:

- *[Fill: Why not for home networks?]*
- *[Fill: Why not for very large scale networks?]*
- *[Fill: When is physical separation better?]*

Final Recommendation: *[Fill: Based on your experiment, when should network admins choose VLANs vs alternatives?]*

Key finding from lab: *[Fill: What was your most important observation about VLAN behavior?]*