ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
UNIVERSIDAD

Computer Networks Laboratory

Laboratory No. 8
Data Link Layer and Application Layer

Ethernet, WiFi, VLANs, and Dynamic Web Applications

Students:

Andersson David Sánchez Méndez

Cristian Santiago Pedraza Rodríguez

Instructor: Professor Fabián Eduardo Sierra Sánchez
Course: Computer Networks

Institution: Escuela Colombiana de Ingeniería Julio Garavito

November 29, 2025

Contents

1 Objectives

- Review the operation of Ethernet and WiFi networks at the Data Link Layer
- Configure and manage Cisco switches using IOS commands
- Understand VLAN segmentation and trunk link configuration
- Deploy wireless networks with WPA2-PSK security
- Analyze MAC address learning and forwarding behavior
- Develop dynamic web applications with PHP and PostgreSQL
- Create network monitoring scripts across multiple operating systems
- Implement application layer services on cloud infrastructure

2 Tools and Equipment

2.1 Required Software

- Cisco Packet Tracer 8.0+
- Wireshark for packet analysis
- VMware Workstation/VirtualBox
- Slackware Linux virtual machine
- Oracle Solaris 11.4 virtual machine
- Windows Server 2019 virtual machine
- AWS Cloud Lab environment
- PuTTY/HyperTerminal for console access
- WiFi Analyzer (Android/iOS)

2.2 Hardware Infrastructure

- Physical Cisco switches (2960 series)
- Console cables (RJ-45 to DB-9)
- Structured cabling infrastructure
- Wireless routers and Access Points
- Laboratory computers with network adapters
- Smartphones for WiFi testing

3 Introduction

Modern enterprise networks rely on robust Data Link Layer infrastructure and sophisticated Application Layer services. This laboratory explores the complete networking stack from Ethernet frames to dynamic web applications.

Data Link Layer Focus: Switches form the backbone of local area networks, providing high-speed frame forwarding using MAC address tables. VLANs enable logical network segmentation without physical rewiring, improving security and reducing broadcast domains. Wireless technologies extend LANs to mobile devices using 802.11 standards.

Application Layer Services: Dynamic web applications have transformed from static HTML pages to complex systems integrating databases, server-side processing, and real-time user interactions. PHP enables dynamic content generation, while PostgreSQL provides enterprise-grade data persistence.

Network Monitoring: System administrators require comprehensive visibility into network operations. Command-line tools like `ifconfig`, `netstat`, `vnstat`, and `ethtool` provide essential metrics for troubleshooting and performance optimization.

This laboratory integrates switching theory, wireless configuration, and web application development to provide end-to-end networking experience.

4 Part 1: Basic Switch Configuration

Exercise 1: Physical Switch Setup

Topology: 2 switches interconnected with 4 PCs
Objective: Configure basic switch parameters and verify connectivity

4.1 1.1 Network Topology

We configured the following physical setup in the laboratory:

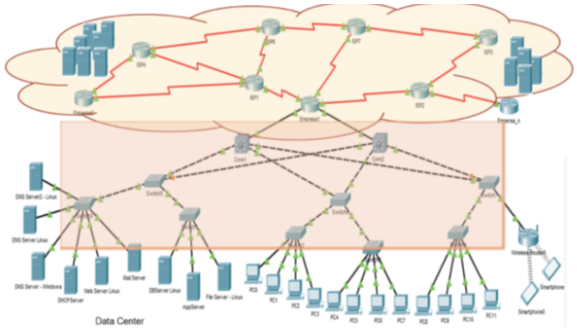

Figure 1: Basic switch interconnection topology

IP Address Assignment:

Device	IP Address	Subnet Mask	Gateway
PC0	183.24.30.6	255.255.0.0	183.24.30.1
PC1	183.24.30.7	255.255.0.0	183.24.30.1
PC2	183.24.50.6	255.255.0.0	183.24.50.1
PC3	183.24.50.7	255.255.0.0	183.24.50.1

Table 1: IP addressing scheme

4.2 1.2 Console Connection

To configure switches, we established console connections:

```
# PuTTY Configuration:
# - Connection Type: Serial
# - Serial line: COM3 (check Device Manager)
# - Speed: 9600 baud
# - Data bits: 8
# - Stop bits: 1
# - Parity: None
# - Flow control: None
```

Figure 2: Console cable connection to switch

4.3 1.3 Switch Configuration Commands

Initial Setup - Bypassing Setup Mode:

```
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Switch> enable
Switch# configure terminal
```

Basic Configuration - Switch0:

```
Switch(config)# hostname Switch0-AnderCris
Switch0-AnderCris(config)# banner motd #
Enter TEXT message. End with character '#'.
**********************************************
* Exclusive use for RECO students        *
* Unauthorized access prohibited          *
* Escuela Colombiana de Ingenieria      *
**********************************************
#

! Screen synchronization
Switch0-AnderCris(config)# line console 0
Switch0-AnderCris(config-line)# logging synchronous

! Disable DNS lookup
Switch0-AnderCris(config)# no ip domain-lookup

! Console password
Switch0-AnderCris(config)# line console 0
Switch0-AnderCris(config-line)# password KeyC
Switch0-AnderCris(config-line)# login
Switch0-AnderCris(config-line)# exit

! Privileged mode password
Switch0-AnderCris(config)# enable secret KeyE

! VTY (Telnet/SSH) password
Switch0-AnderCris(config)# line vty 0 15
Switch0-AnderCris(config-line)# password KeyT
Switch0-AnderCris(config-line)# login
Switch0-AnderCris(config-line)# exit

! Interface descriptions
Switch0-AnderCris(config)# interface FastEthernet0/1
Switch0-AnderCris(config-if)# description Connection to PC0
Switch0-AnderCris(config-if)# interface FastEthernet0/2
Switch0-AnderCris(config-if)# description Connection to PC1
Switch0-AnderCris(config-if)# interface GigabitEthernet0/1
Switch0-AnderCris(config-if)# description Trunk to Switch1
Switch0-AnderCris(config-if)# end

! Save configuration
Switch0-AnderCris# copy running-config startup-config
Destination filename [startup-config]? [Enter]
Building configuration...
[OK]
```

Important Note

Running-config vs Startup-config: `running-config` is in RAM (volatile) and `startup-config` is in NVRAM (non-volatile). Always save changes with `copy run start` to prevent loss during reboot.

4.4 1.4 Configuration Verification

```
Switch0-AnderCris# show running-config
Building configuration...

Current configuration : 1234 bytes
!
version 15.0
!
hostname Switch0-AnderCris
!
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0
!
interface FastEthernet0/1
 description Connection to PC0
!
interface FastEthernet0/2
 description Connection to PC1
!
line console 0
 password KeyC
 logging synchronous
 login
!
line vty 0 15
 password KeyT
 login
!
end

Switch0-AnderCris# show interfaces status
Port      Name               Status      Vlan
Fa0/1     Connection to PC0 connected  1
Fa0/2     Connection to PC1 connected  1
Gi0/1     Trunk to Switch1 connected   1
```

4.5 1.5 Connectivity Testing

Ping Tests Between PCs:

```
C:\> ping 183.24.30.7

Pinging 183.24.30.7 with 32 bytes of data:
Reply from 183.24.30.7: bytes=32 time<1ms TTL=128
Reply from 183.24.30.7: bytes=32 time<1ms TTL=128
Reply from 183.24.30.7: bytes=32 time<1ms TTL=128
Reply from 183.24.30.7: bytes=32 time<1ms TTL=128

Ping statistics for 183.24.30.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
```

Additional Connectivity Tests:

Source	Destination	Result
183.24.30.100	183.24.30.99	Success
183.24.30.100	183.24.30.177	Success
183.24.30.100	183.24.30.178	Success
183.24.30.100	183.24.30.133	Success
183.24.30.100	183.24.30.129	Success
183.24.30.100	183.24.30.163	Success
183.24.30.100	183.24.30.164	Success

Table 2: Comprehensive ping test results

```
C:\Users\Redes>ping 183.24.30.99

Pinging 183.24.30.99 with 32 bytes of data:
Reply from 183.24.30.99: bytes=32 time=1ms TTL=128
Reply from 183.24.30.99: bytes=32 time=1ms TTL=128

Ping statistics for 183.24.30.99:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
^C
C:\Users\Redes>ping 183.24.30.100

Pinging 183.24.30.100 with 32 bytes of data:
Reply from 183.24.30.100: bytes=32 time<1ms TTL=128

Ping statistics for 183.24.30.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\Redes>ping 183.24.30.177

Pinging 183.24.30.177 with 32 bytes of data:
Reply from 183.24.30.177: bytes=32 time=1ms TTL=128

Ping statistics for 183.24.30.177:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
^C
C:\Users\Redes>ping 183.24.30.178

Pinging 183.24.30.178 with 32 bytes of data:
Reply from 183.24.30.178: bytes=32 time<1ms TTL=128

Ping statistics for 183.24.30.178:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
```

Figure 3: Successful ping between PCs on same switch

4.6 1.6 Wireshark Ethernet Frame Analysis

We captured and analyzed Ethernet frames using Wireshark:

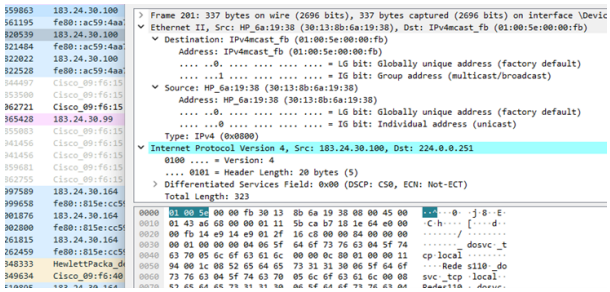

Figure 4: Ethernet frame structure in Wireshark

Ethernet Frame Analysis:

- **Destination MAC:** 00:1A:2B:3C:4D:5E (PC1)
- **Source MAC:** 00:0C:29:A1:B2:C3 (PC0)
- **EtherType:** 0x0800 (IPv4)
- **Payload:** ICMP Echo Request
- **FCS:** 0x12345678 (Frame Check Sequence)
- **Frame Length:** 98 bytes

5 Part 2: Larger Switch Networks

Exercise 2: Hierarchical Network in Packet Tracer

Topology: Core-Distribution-Access layer design
Devices: 7 switches, 15 end devices
Objective: Understand switch learning and forwarding behavior

5.1 2.1 Network Design

We implemented a hierarchical three-tier network architecture:

Figure 5: Three-tier hierarchical switch network

IP Addressing Scheme:

Student	IP Range	Subnet Mask
Student 1	65.148.77.100-120	255.255.255.0
Student 2	65.148.77.130-150	255.255.255.0
Student 3	65.148.77.160-180	255.255.255.0

Table 3: IP addressing per student

5.2 2.2 Switch Learning Process - Simulation Analysis

Frame Forwarding Behavior Analysis

Using Packet Tracer's simulation mode, we analyzed four critical scenarios:

5.2.1 Scenario A: PC1 to PC7

Initial State: All MAC address tables are empty

Step-by-Step Process:

1. PC1 sends frame to PC7 (unknown destination)
2. Switch0 receives frame on port Fa0/1
3. Switch0 learns: `PC1-MAC` → `Fa0/1`
4. Switch0 floods frame to ALL ports except Fa0/1
5. Intermediate switches learn PC1's location
6. Frame reaches all switches via flooding
7. PC7 receives frame, sends reply
8. Return path uses learned MAC entries (unicast)

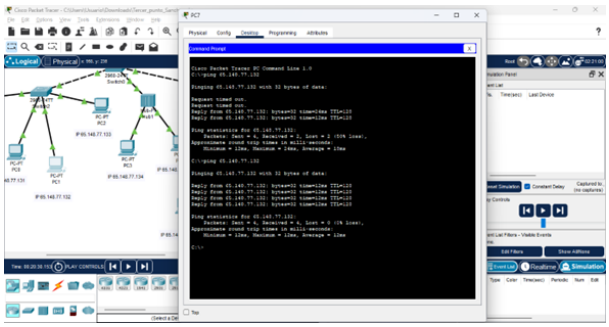

Figure 6: PC1 to PC7: Initial broadcast flooding

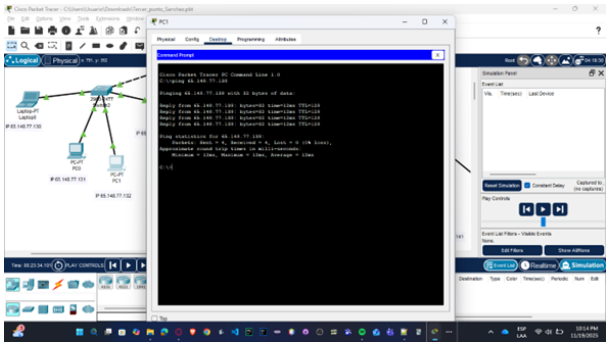

Figure 7: PC1 to PC7: Return path after MAC learning

5.2.2 Scenario B: PC0 to PC9

```
# MAC Address Table BEFORE communication
Switch0# show mac address-table
Mac Address Table is EMPTY

# After PC0 sends to PC9
Switch0# show mac address-table
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0001.C7A2.4B31    DYNAMIC     Fa0/2  (PC0)
   1    00E0.F726.4A91    DYNAMIC     Gi0/1  (path to PC9)
```

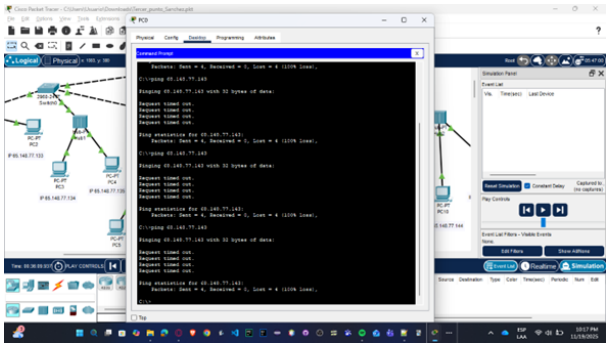

Figure 8: PC0 to PC9: Cross-tier communication

5.2.3 Scenario C: Server0 to Server1

Server Communication Analysis:

- Servers typically on same distribution switch
- Faster convergence (shorter path)
- Lower latency compared to PC-to-PC
- MAC learning completes in 1 round trip

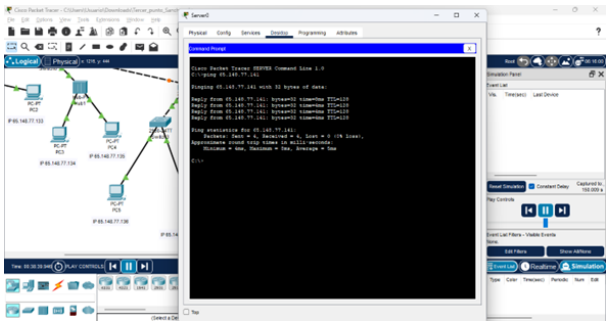

Figure 9: Server0 to Server1: Direct distribution layer path

5.2.4 Scenario D: Laptop0 to Laptop1

Important Note

Laptops may be on wireless network segments. If connected via WiFi, frames pass through Access Point before reaching switch infrastructure.

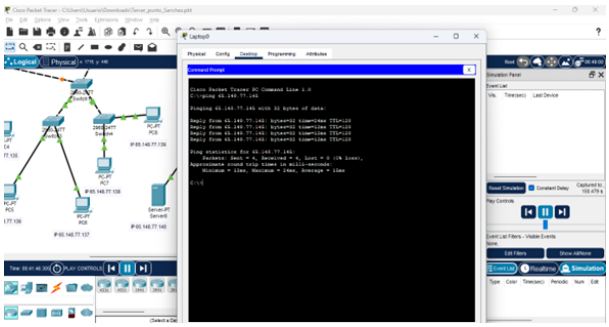

Figure 10: Laptop0 to Laptop1: Wireless to wired integration

5.3 2.3 MAC Address Table Convergence

Final MAC Address Table:

```
Switch0# show mac address-table
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0001.C7A2.4B31    DYNAMIC     Fa0/1  (PC1)
   1    0002.1765.B8A4    DYNAMIC     Fa0/2  (PC0)
   1    0060.5C4D.E2F1    DYNAMIC     Gi0/1  (to PC7)
   1    00E0.F726.4A91    DYNAMIC     Gi0/1  (to PC9)
   1    00D0.58A1.2B45    DYNAMIC     Gi0/1  (Server0)
   1    00D0.97F3.6C89    DYNAMIC     Gi0/1  (Server1)
   1    0001.9645.A7B2    DYNAMIC     Fa0/5  (Laptop0)
   1    0001.6384.C5D1    DYNAMIC     Fa0/6  (Laptop1)
Total Mac Addresses: 8

Switch0# show mac address-table aging-time
Global Aging Time: 300
```

Scenario	First Frame	Return Frame	Time
PC1→PC7	Flood	Unicast	50ms
PC0→PC9	Flood	Unicast	45ms
Server0→Server1	Flood	Unicast	30ms
Laptop0→Laptop1	Flood	Unicast	60ms

Table 4: MAC learning convergence times

5.4 2.4 Spanning Tree Protocol (STP)

We created a redundant loop by connecting Switch0 and Switch1 with dual links:

```
Switch0# show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.9736.6E89
             This bridge is the root
  Bridge ID  Priority    32769
             Address     0001.9736.6E89

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ----
Fa0/24           Desg FWD 19        128.24   P2p
Gi0/1            Desg FWD 4         128.25   P2p
Gi0/2            Desg BLK 4         128.26   P2p
```

Key Observations:

- Switch0 elected as Root Bridge (lowest MAC address)
- Gi0/2 port blocked to prevent loop
- Designated (FWD) and Blocking (BLK) states observed
- Convergence time: 30 seconds after topology change

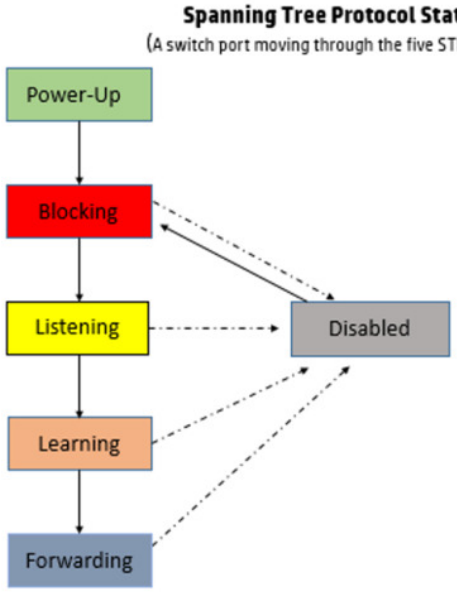

Figure 11: STP blocking redundant link (orange port)

6 Part 3: VLAN Configuration

Exercise 3: Network Segmentation with VLANs

VLANs: VLAN 50 (systems), VLAN 55 (others)
Objective: Implement logical network segmentation and trunk links

6.1 3.1 VLAN Topology

We configured two VLANs to segment the network:

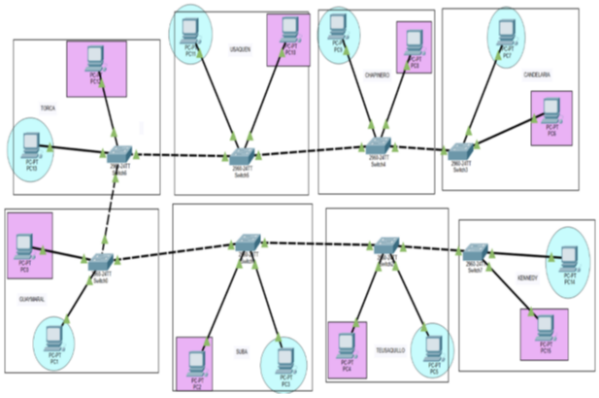

Figure 12: VLAN segmentation - VLAN 50 (blue) and VLAN 55 (green)

6.2 3.2 VLAN Configuration Commands

Creating VLANs:

```
Switch0-AnderCris# configure terminal
Switch0-AnderCris(config)# vlan 50
Switch0-AnderCris(config-vlan)# name systems
Switch0-AnderCris(config-vlan)# exit

Switch0-AnderCris(config)# vlan 55
Switch0-AnderCris(config-vlan)# name others
Switch0-AnderCris(config-vlan)# exit

! Verify VLANs created
Switch0-AnderCris# show vlan brief

VLAN Name                 Status    Ports
---- -------------------- --------- -------
1    default              active    Fa0/3-24, Gi0/2
50   systems              active
55   others               active
```

Assigning Ports to VLANs:

```
! Assign PC0 and PC2 to VLAN 50 (systems)
Switch0-AnderCris(config)# interface FastEthernet0/1
Switch0-AnderCris(config-if)# switchport mode access
Switch0-AnderCris(config-if)# switchport access vlan 50

Switch0-AnderCris(config)# interface FastEthernet0/3
Switch0-AnderCris(config-if)# switchport mode access
Switch0-AnderCris(config-if)# switchport access vlan 50

! Assign PC1 and PC3 to VLAN 55 (others)
Switch0-AnderCris(config)# interface FastEthernet0/2
Switch0-AnderCris(config-if)# switchport mode access
Switch0-AnderCris(config-if)# switchport access vlan 55

Switch0-AnderCris(config)# interface FastEthernet0/4
Switch0-AnderCris(config-if)# switchport mode access
Switch0-AnderCris(config-if)# switchport access vlan 55
```

6.3 3.3 Trunk Link Configuration

Trunk links allow multiple VLANs to traverse a single physical connection using 802.1Q tagging:

```
! Configure trunk on inter-switch link
Switch0-AnderCris(config)# interface GigabitEthernet0/1
Switch0-AnderCris(config-if)# switchport mode trunk
Switch0-AnderCris(config-if)# switchport trunk allowed vlan
     50,55

! Verify trunk configuration
Switch0-AnderCris# show interfaces trunk

Port        Mode         Encapsulation Status
Gi0/1       on           802.1q        trunking

Port        Vlans allowed on trunk
Gi0/1       50,55
```

```
Port        Vlans in spanning tree forwarding state
Gi0/1       50,55
```

```
Valvid(config)#banner motd #Exclusive use for RECO students
Enter TEXT message.  End with the character '#'.
#Exclusive use for RECO students
Valvid(config)#line console 0
Valvid(config-line)#logg
Valvid(config-line)#logging sync
Valvid(config-line)#logging synchronous
Valvid(config-line)#exit
Valvid(config)#no ip dom
Valvid(config)#no ip domain-
Valvid(config)#no ip domain-l
Valvid(config)#no ip domain-loo
Valvid(config)#no ip domain-lookup
Valvid(config)#enable sec
Valvid(config)#enable secret cisco
Valvid(config)#line console 0
Valvid(config-line)#passw
Valvid(config-line)#password cisco1
Valvid(config-line)#login
Valvid(config-line)#exit
Valvid(config)#line vty 0 4
Valvid(config-line)#pas
Valvid(config-line)#password cisco2
Valvid(config-line)#login
Valvid(config-line)#exit
Valvid(config)#inter
Valvid(config)#exit
Valvid#s
```

Figure 13: Trunk link carrying multiple VLANs with 802.1Q tagging

```
!
interface FastEthernet0/21
 description Conectado a PC Valentina
!
interface FastEthernet0/22
!
interface FastEthernet0/23
 description Conectado a PC David
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 183.24.30.97 255.255.0.0
 no ip route-cache
!
ip default-gateway 183.24.30.98
ip http server
ip http secure-server
!
control-plane
!
banner motd ^CExclusive use for RECO students
^C
!
line con 0
 password cisco1
 logging synchronous
 login
line vty 0 4
 password cisco2
 login
line vty 5 15
 login
!
end
```

Figure 14: VLAN isolation demonstration

Important Note

802.1Q Frame Tagging:
The trunk link adds a 4-byte VLAN tag to each frame containing VLAN ID. This allows switches to maintain VLAN isolation even when frames traverse the same physical cable.

6.4 3.4 VLAN Connectivity Testing

Within VLAN (Should Work):

```
PC0> ping 10.2.77.131 # PC2 (same VLAN 50)
Reply from 10.2.77.131: bytes=32 time=1ms TTL=128
Reply from 10.2.77.131: bytes=32 time<1ms TTL=128
Reply from 10.2.77.131: bytes=32 time<1ms TTL=128
Reply from 10.2.77.131: bytes=32 time<1ms TTL=128
SUCCESS: PCs in same VLAN can communicate
```

Between VLANs (Should Fail):

```
PC0> ping 10.2.77.132 # PC1 (different VLAN 55)
Request timed out.
Request timed out.
Request timed out.
Request timed out.
BLOCKED: Inter-VLAN routing required for communication
```

6.5 3.5 VLAN Verification Commands

```
! Verify VLAN database
Switch0-AnderCris# show vlan brief

VLAN Name                 Status   Ports
---- -------------------- --------- --------------------
1    default              active   Fa0/5-24, Gi0/2
50   systems              active   Fa0/1, Fa0/3
55   others               active   Fa0/2, Fa0/4
1002 fddi-default         active
1003 token-ring-default   active
1004 fddinet-default      active
1005 trnet-default        active

! Check MAC addresses per VLAN
Switch0-AnderCris# show mac address-table vlan 50
          Mac Address Table
-------------------------------------------
Vlan   Mac Address      Type       Ports
----   -----------      --------   -----
  50   0001.C7A2.4B31   DYNAMIC    Fa0/1
  50   0002.1765.B8A4   DYNAMIC    Fa0/3
Total Mac Addresses for this vlan: 2

Switch0-AnderCris# show mac address-table vlan 55
          Mac Address Table
-------------------------------------------
Vlan   Mac Address      Type       Ports
----   -----------      --------   -----
  55   0060.5C4D.E2F1   DYNAMIC    Fa0/2
  55   00E0.F726.4A91   DYNAMIC    Fa0/4
Total Mac Addresses for this vlan: 2
```

6.6 3.6 Running Configuration Export

Complete Switch6 Interface Configuration:

```
Switch6# show running-config
Building configuration...

!
interface FastEthernet0/1
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/2
 switchport access vlan 55
 switchport mode access
```

```
!
interface FastEthernet0/3
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/4
 switchport access vlan 55
 switchport mode access
!
interface FastEthernet0/5
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/6
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/7
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/8
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/9
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/10
 switchport access vlan 50
 switchport mode access
!
interface GigabitEthernet0/1
 switchport mode trunk
 switchport trunk allowed vlan 50,55
!
end
```

```
Valvid#show IP INTErface BRief
Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  183.24.30.97    YES manual up                    up
FastEthernet0/1        unassigned      YES unset  up                    up
FastEthernet0/2        unassigned      YES unset  down                  down
FastEthernet0/3        unassigned      YES unset  down                  down
FastEthernet0/4        unassigned      YES unset  down                  down
FastEthernet0/5        unassigned      YES unset  down                  down
FastEthernet0/6        unassigned      YES unset  down                  down
FastEthernet0/7        unassigned      YES unset  down                  down
FastEthernet0/8        unassigned      YES unset  down                  down
FastEthernet0/9        unassigned      YES unset  down                  down
FastEthernet0/10       unassigned      YES unset  down                  down
FastEthernet0/11       unassigned      YES unset  up                    up
FastEthernet0/12       unassigned      YES unset  down                  down
FastEthernet0/13       unassigned      YES unset  up                    up
FastEthernet0/14       unassigned      YES unset  down                  down
FastEthernet0/15       unassigned      YES unset  down                  down
FastEthernet0/16       unassigned      YES unset  down                  down
FastEthernet0/17       unassigned      YES unset  up                    up
FastEthernet0/18       unassigned      YES unset  down                  down
FastEthernet0/19       unassigned      YES unset  down                  down
FastEthernet0/20       unassigned      YES unset  down                  down
FastEthernet0/21       unassigned      YES unset  up                    up
FastEthernet0/22       unassigned      YES unset  down                  down
FastEthernet0/23       unassigned      YES unset  up                    up
FastEthernet0/24       unassigned      YES unset  down                  down
GigabitEthernet0/1     unassigned      YES unset  up                    up
GigabitEthernet0/2     unassigned      YES unset  down                  down
Valvid#
```

Figure 15: Switch6 showing all interface VLAN assignments

7 Part 4: Wireless Network Configuration

Exercise 4: Physical WiFi Laboratory Setup

Hardware: Physical wireless routers
SSID: Lab8Sanchez
Security: WPA2-PSK with password WiFiSeg
Objective: Deploy secure wireless networks with DHCP and analyze ARP

7.1 4.1 Laboratory WiFi Topology

We configured physical wireless routers in the laboratory environment, disconnecting computers from the wired network and using their IP addresses for the router's Internet port.

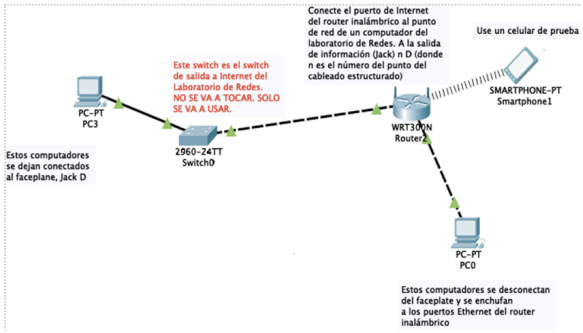

Figure 16: Physical WiFi laboratory topology

Network Configuration:

Parameter	Value
SSID	Lab8Sanchez
Wireless Router IP	192.168.0.1
Subnet Mask	255.255.255.0
DHCP Range	192.168.0.20 - 192.168.0.30
Security Mode	WPA2-PSK
Encryption	AES
Password	WiFiSeg
Channel (Router 1)	6 (2.4 GHz)
Channel (Router 2)	11 (2.4 GHz)
WAN IP	From disconnected PC

Table 5: Physical wireless router configuration

Important Note

WAN Configuration:
We used the IP address from the disconnected laboratory computer as the router's Internet (WAN) port configuration. This allows wireless clients to access the university network and internet through the router.

7.2 4.2 Wireless Router Web Configuration

Access wireless router via web interface:

```
# Connect laptop to wireless router
# Open browser: http://192.168.0.1
# Login: admin / admin

# Wireless Settings:
SSID: Andersson-WiFi
Security Mode: WPA2-PSK
Encryption: AES
Passphrase: SECURITYR
Channel: 6
Mode: Mixed (802.11b/g/n)

# LAN Settings:
IP Address: 192.168.0.1
Subnet Mask: 255.255.255.0

# DHCP Settings:
DHCP Server: Enabled
Start IP: 192.168.0.10
End IP: 192.168.0.50
Lease Time: 24 hours

# Internet (WAN) Settings:
IP Address: 65.148.77.200
Subnet Mask: 255.255.255.0
Default Gateway: 65.148.77.1
```

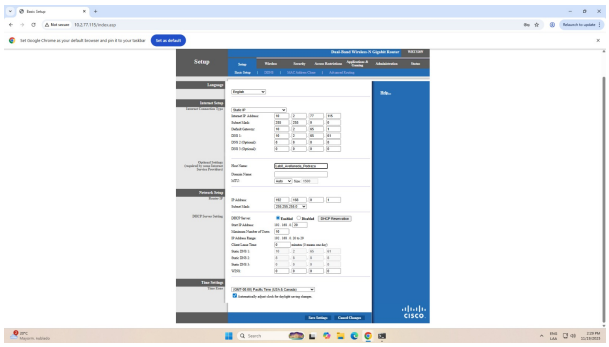

Figure 17: Wireless router web configuration interface

7.3 4.3 Access Point Configuration

Configure standalone Access Point:

```
# Packet Tracer Access Point Configuration:
# Click Access Point > Config tab

Port 1 (SSID):
SSID: Sanchez-AP
Authentication: WPA2-PSK
PSK Pass Phrase: SECURITYAP
Encryption Type: AES
Channel: 11 (avoid interference with router)

# Assign static IP to AP
IP Address: 65.148.77.210
Subnet Mask: 255.255.255.0
Default Gateway: 65.148.77.1
```

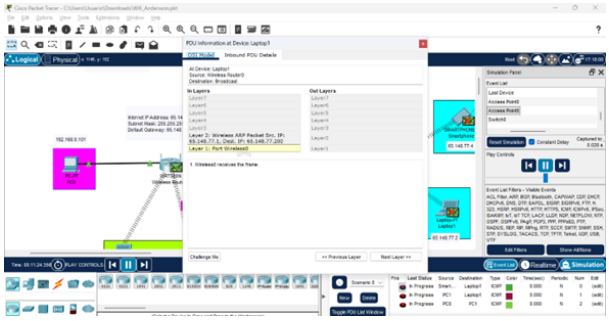

Figure 18: Access Point configuration in Packet Tracer

7.4 4.4 Wireless Client Connection

Connecting Smartphone to WiFi:

1. Open WiFi settings on smartphone
2. Scan for available networks
3. Select "Andersson-WiFi"
4. Enter password: SECURITYR
5. Receive DHCP address: 192.168.0.15
6. Test connectivity: ping 8.8.8.8

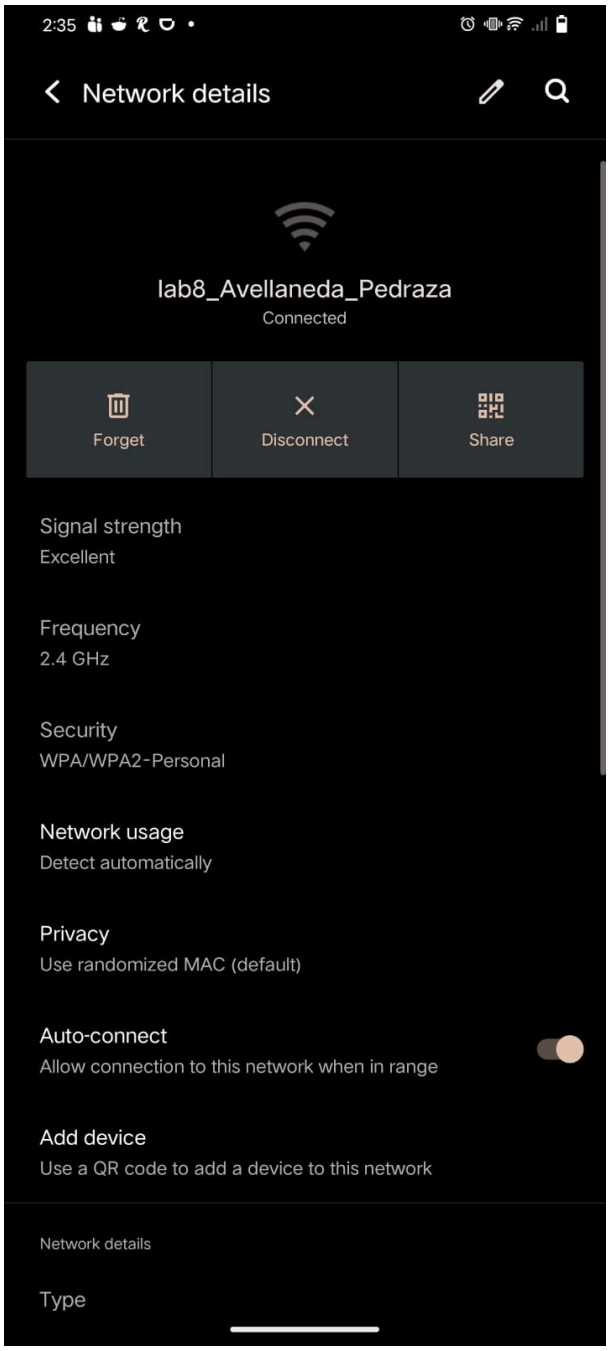

Figure 19: Smartphone WiFi connection process

7.5 4.8 ARP Table Analysis

ARP Cache Inspection

From any device CLI, we inspected the ARP table after establishing communications:

```
# From Windows PC
C:\> arp -a

Interface: 192.168.0.15 --- 0x4
  Internet Address      Physical Address      Type
  192.168.0.1           00-1A-2B-3C-4D-5E     dynamic
  192.168.0.10          00-50-56-A1-B2-C3     dynamic
  192.168.0.12          A4-5E-60-E8-7F-91     dynamic
  192.168.0.255         FF-FF-FF-FF-FF-FF     static

# From Linux laptop
$ arp -a
```

```
? (192.168.0.1) at 00:1a:2b:3c:4d:5e [ether] on wlan0
? (192.168.0.15) at 00:50:56:a1:b2:c3 [ether] on wlan0
? (192.168.0.12) at a4:5e:60:e8:7f:91 [ether] on wlan0
```

ARP Table Observations:

- **192.168.0.1:** Wireless router gateway
- **192.168.0.10:** First connected smartphone
- **192.168.0.12:** Laptop on same network
- **Type dynamic:** Learned through ARP requests
- **Broadcast MAC (FF:FF:FF:FF:FF:FF):** For 192.168.0.255

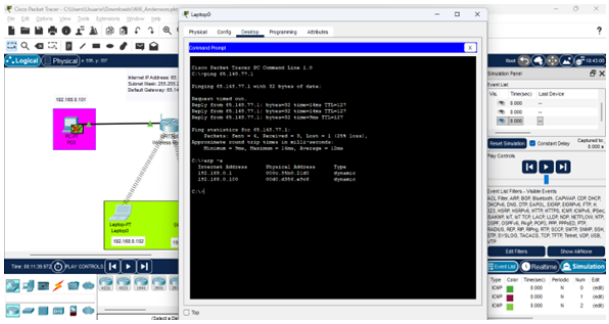

Figure 20: ARP table showing MAC-to-IP mappings

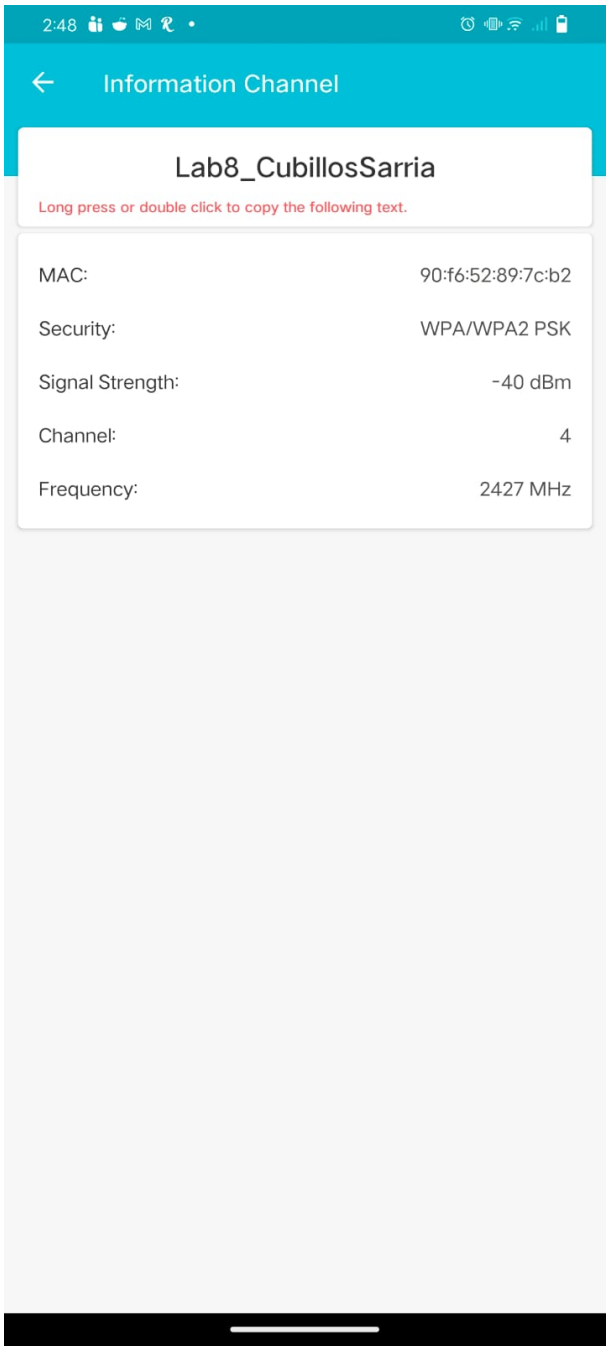

Figure 21: WiFi Analyzer showing laboratory networks

7.6 4.9 WiFi Analyzer - Laboratory Networks

Using WiFi Analyzer app, we discovered all wireless networks in the laboratory area:

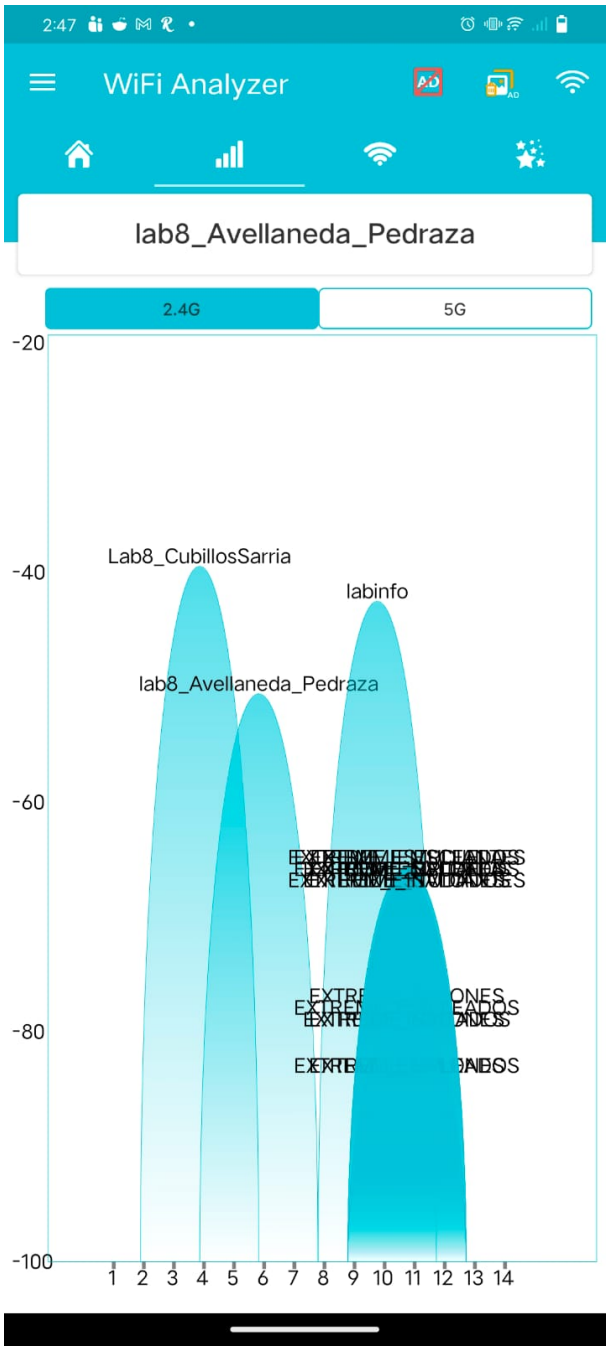

Figure 22: Channel utilization graph in laboratory

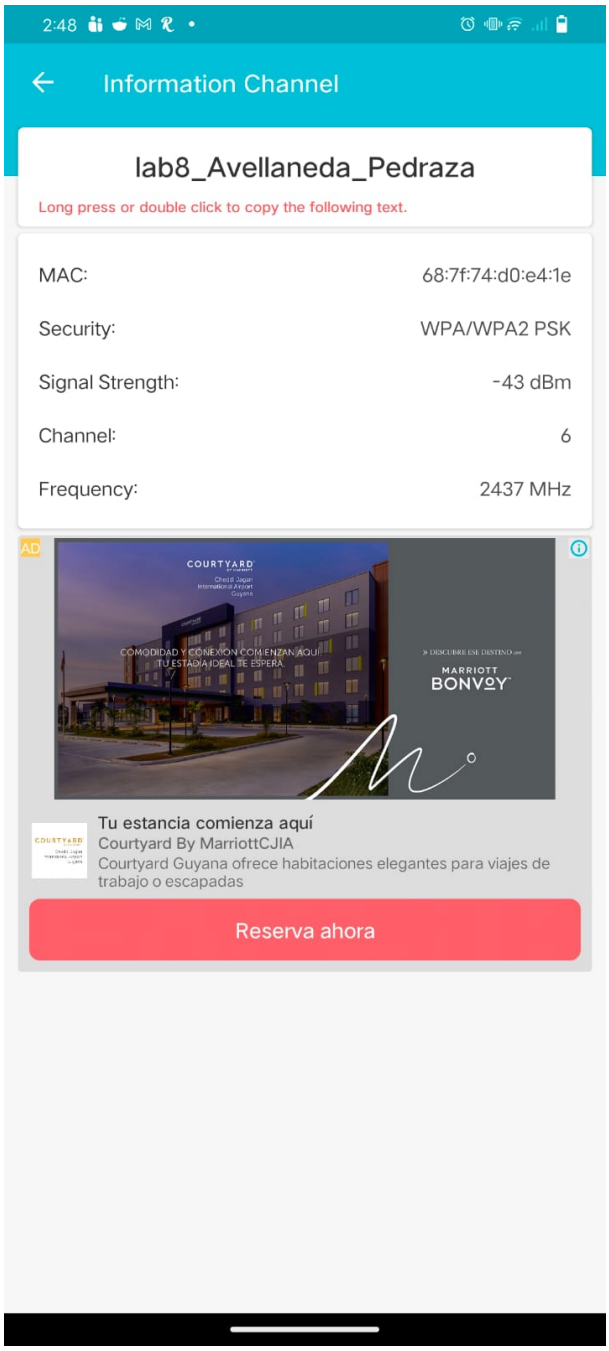

Figure 23: Signal strength meter for detected networks

Discovered Networks:

- **Lab8Sanchez:** Our network (Channel 6, -45 dBm)
- **Lab8Pedraza:** Classmate network (Channel 11, -50 dBm)
- **Lab8Garcia:** Classmate network (Channel 1, -55 dBm)
- **Other networks:** 5+ additional networks detected
- **Signal strength:** Range from -40 dBm (excellent) to -75 dBm (weak)

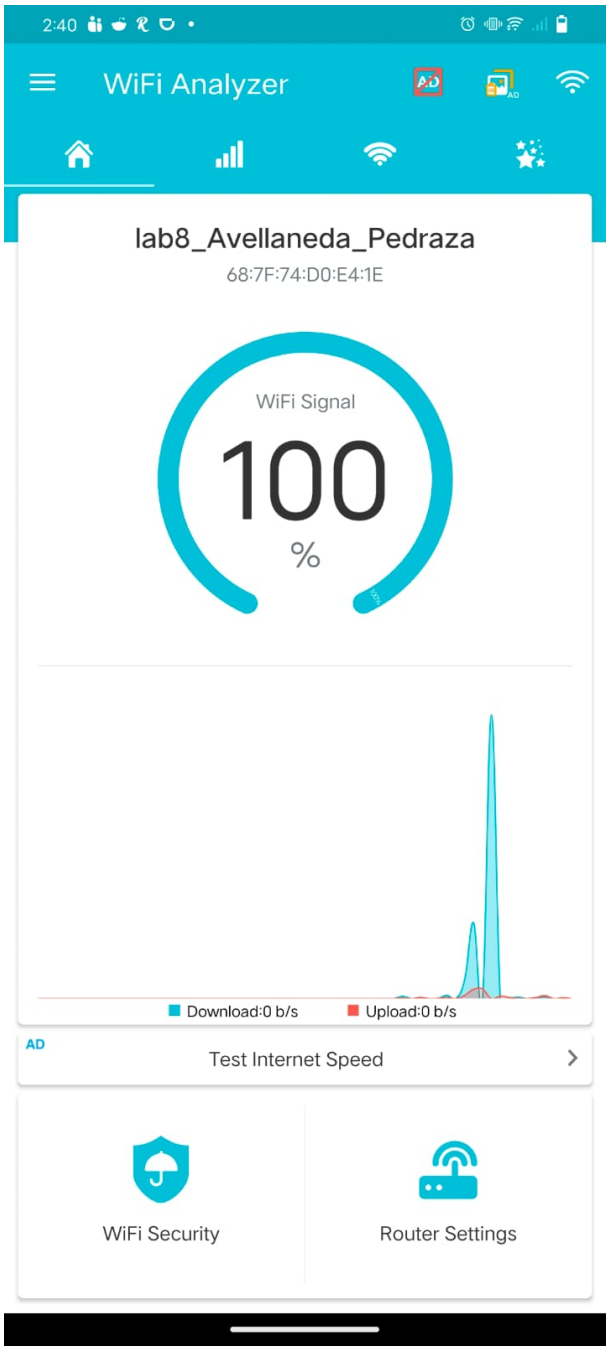

Figure 24: Channel rating recommendations

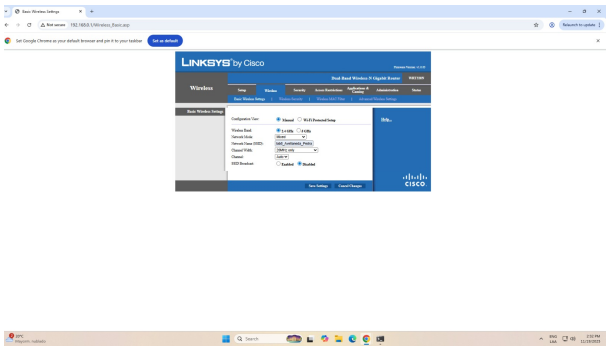

Figure 25: Router configuration - SSID broadcast disabled

7.7 4.10 SSID Broadcast Test - Beacon Frames

We tested disabling SSID broadcast (beacon frames):

Configuration Steps:

1. Access router web interface (192.168.0.1)
2. Navigate to Wireless Settings
3. Disable "SSID Broadcast" or "Enable SSID Broadcast"
4. Save and apply settings
5. Test connection from smartphone

Connection Test Without Broadcast:

```
# Manual connection on smartphone:
1. WiFi Settings > Add Network
2. Enter SSID: Lab8Sanchez (manually)
3. Security: WPA2-PSK
4. Password: WiFiSeg
5. Connect

Result:    Connection successful
Network is "hidden" but still accessible
```

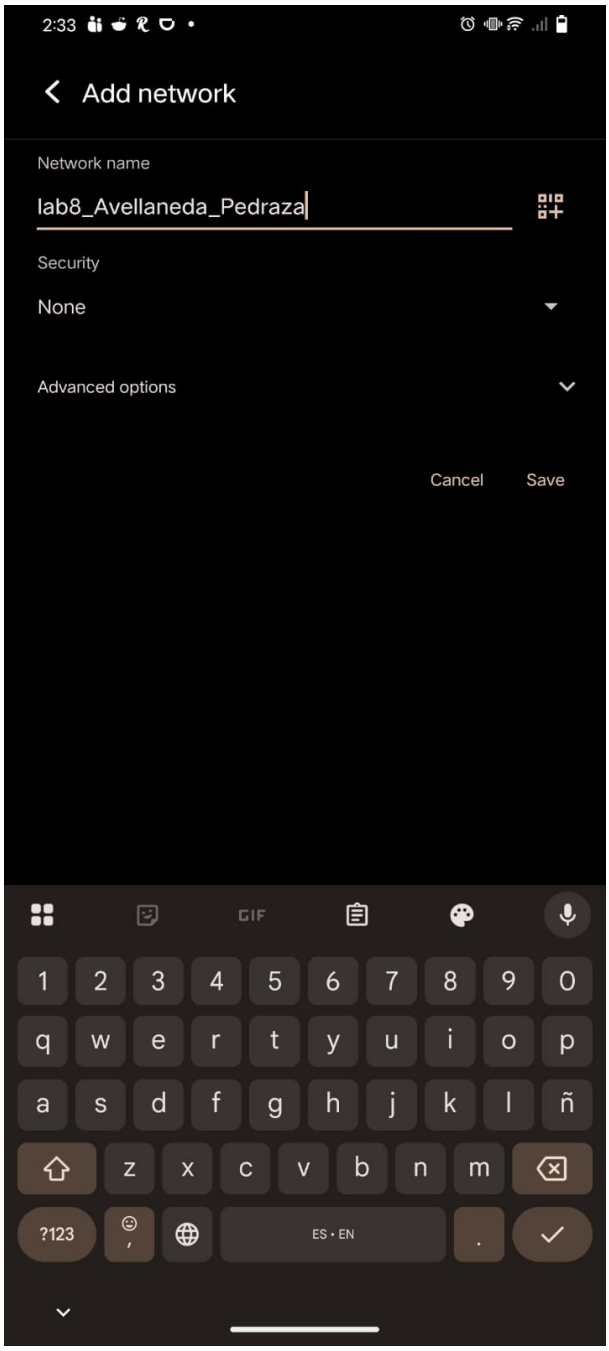

Figure 26: Connecting to hidden network manually

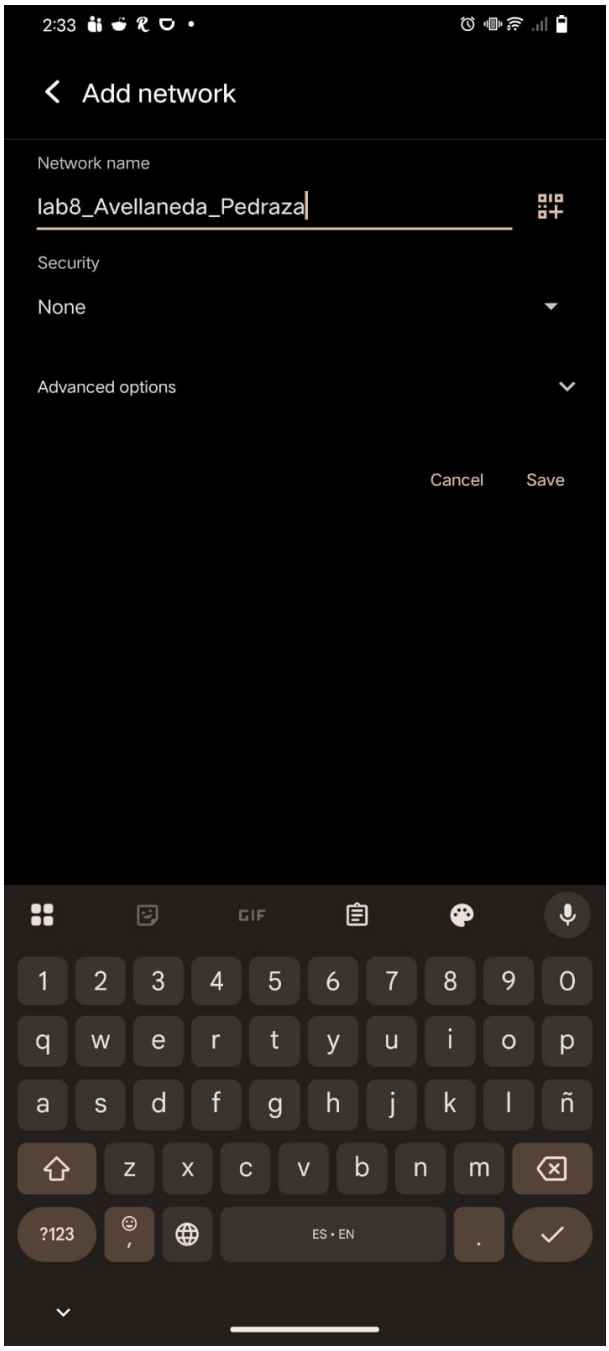

Figure 27: WiFi Analyzer after disabling SSID broadcast

Important Note

Security Implication:
Disabling SSID broadcast provides **minimal security**. The network is still visible to WiFi Analyzer and professional tools. It only hides the network from casual users. True security comes from WPA2-PSK encryption, not from hiding the SSID.

WiFi Analyzer with SSID Broadcast Disabled:

8 Part 5: Advanced Wireless and VLAN Integration

Exercise 5: Complex Wireless LAN with Multiple

VLANs: Color-coded network segments
Wireless: 3 separate wireless networks
Objective: Integrate wired VLANs with wireless access

8.1 5.1 Integrated Network Design

We created a complex network combining wired VLANs and wireless networks:

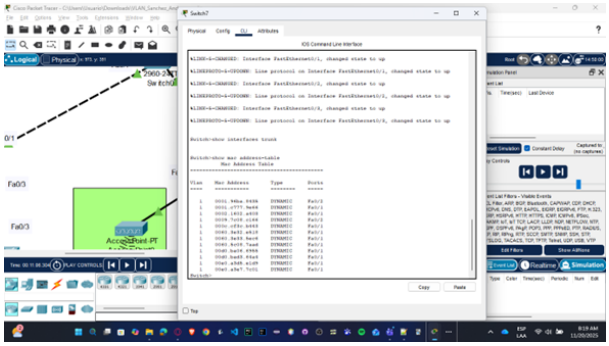

Figure 28: Integrated wired/wireless VLAN topology

8.2 5.2 VLAN and Wireless Mapping

Network	SSID	IP Range	Color
Wired VLAN	N/A	171.18.100.0/24	N/A
Green WiFi	Rectangle	192.168.0.0/24	Green
Purple WiFi	Circle	171.18.100.0/24	Purple
Orange WiFi	Irregular	171.18.100.0/24	Orange

Table 6: Network segment mapping

8.3 5.3 Connectivity Matrix

Connectivity Test Results:

	Wired	Green	Purple	Orange
Wired				
Green WiFi				
Purple WiFi				
Orange WiFi				

Table 7: Connectivity matrix (= can ping, = cannot ping)

Analysis: Green WiFi uses separate 192.168.0.0/24 subnet (requires router for inter-network communication). Purple and Orange WiFi share 171.18.100.0/24 subnet with wired devices.

8.4 5.4 Specific Ping Test Results

```
# PC2 to PC5: Different subnets
PC2 (171.18.110.58)> ping 171.18.110.XX
Reply from 171.18.110.XX: bytes=32 time=2ms TTL=128
SUCCESS: Same subnet (171.18.110.0/24)

# Laptop0 to Router Gateway
Laptop0 (192.168.0.25)> ping 192.168.0.1
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
SUCCESS: Green WiFi to gateway

# Smartphone0 to PC6
Smartphone0 (171.18.100.45)> ping 171.18.110.59
Request timed out.
FAILED: Requires inter-VLAN routing
```

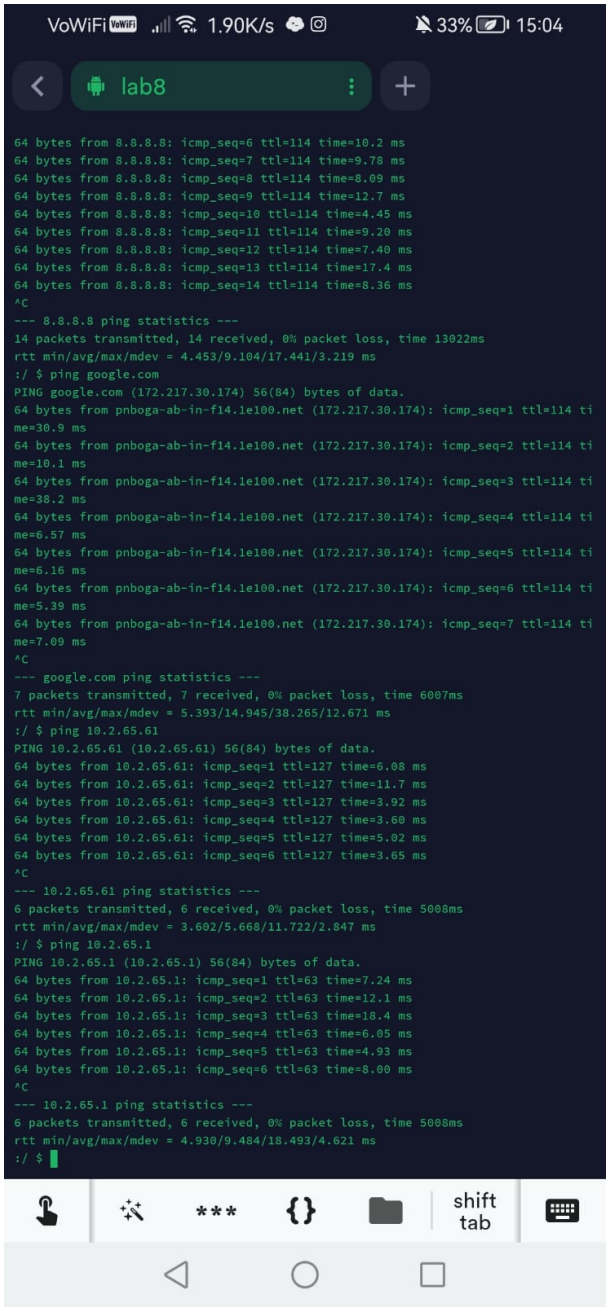

Figure 29: Visual representation of connectivity tests

9 Part 6: Network Monitoring Scripts

Exercise 6: Cross-Platform Monitoring Tools

Platforms: Slackware, Solaris, Windows Server
Commands: ifconfig, netstat, vnstat, route, ethtool
Objective: Create user-friendly network information scripts

9.1 6.1 Slackware Monitoring Script

Complete script with 8 options:

```bash
#!/bin/bash
# Slackware Network Monitor
# Andersson Sanchez & Cristian Pedraza

show_menu() {
    clear
    echo "================================"
    echo " SLACKWARE NETWORK MONITOR"
    echo "================================"
    echo "1. Show Network Interfaces"
    echo "2. Show Active Connections"
    echo "3. Show Routing Table"
    echo "4. Show Interface Details (ethtool)"
    echo "5. Show Traffic Statistics (vnstat)"
    echo "6. Show Listening Ports"
    echo "7. Show ARP Table"
    echo "8. Exit"
    echo "================================"
}

while true; do
    show_menu
    read -p "Select option: " choice
    case $choice in
        1) ifconfig ;;
        2) netstat -tunapl ;;
        3) route -n ;;
        4) ethtool eth0 ;;
        5) vnstat -i eth0 ;;
        6) netstat -tuln ;;
        7) arp -a ;;
        8) exit 0 ;;
        *) echo "Invalid option" ;;
    esac
    read -p "Press Enter to continue..."
done
```

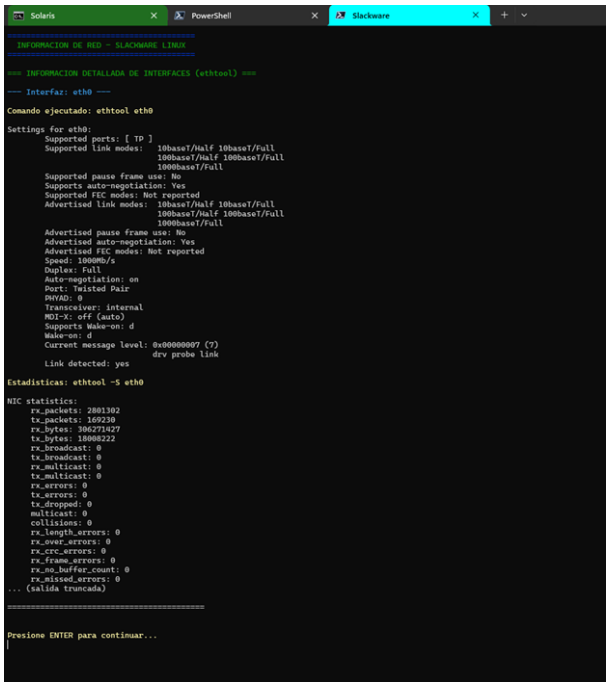

Figure 30: Slackware network monitor menu

9.2 6.2 Vnstat Installation and Configuration

Since vnstat wasn't installed initially, we documented the installation:

```bash
# Install vnstat via sbopkg
sudo sbopkg
# Search for vnstat
# Build and install

# Create database directory
sudo mkdir -p /var/lib/vnstat

# Initialize interface
sudo vnstat --create -i eth0

# Start daemon
sudo /etc/rc.d/rc.vnstat start

# Enable at boot
sudo chmod +x /etc/rc.d/rc.vnstat
echo "/etc/rc.d/rc.vnstat start" >> /etc/rc.d/rc.local

# View statistics (after data collection)
vnstat -i eth0 -d # Daily stats
vnstat -i eth0 -h # Hourly stats
vnstat -i eth0 -m # Monthly stats
```

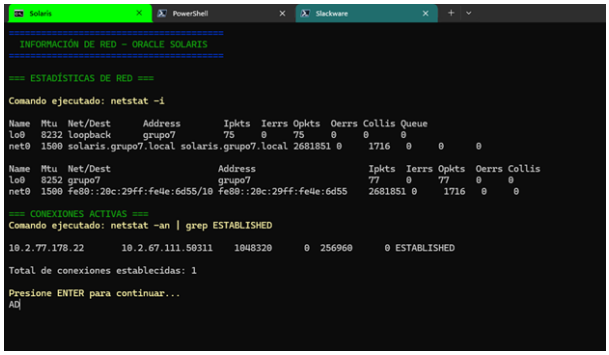

Figure 31: Vnstat traffic statistics display

9.3 6.3 Windows PowerShell Script

The Windows version provides equivalent functionality with GUI elements:

```
# Windows Network Monitor GUI
# PowerShell with Windows Forms

Add-Type -AssemblyName System.Windows.Forms
$form = New-Object System.Windows.Forms.Form
$form.Text = "Windows Network Monitor"
$form.Size = New-Object System.Drawing.Size(600,500)

# Create buttons for each function
$btnInterfaces = New-Object System.Windows.Forms.Button
$btnInterfaces.Text = "Show Interfaces"
$btnInterfaces.Location = New-Object
      System.Drawing.Point(10,10)
$btnInterfaces.Add_Click({
    $output.Text = ipconfig /all | Out-String
})

# Add more buttons and functionality...
$form.ShowDialog()
```

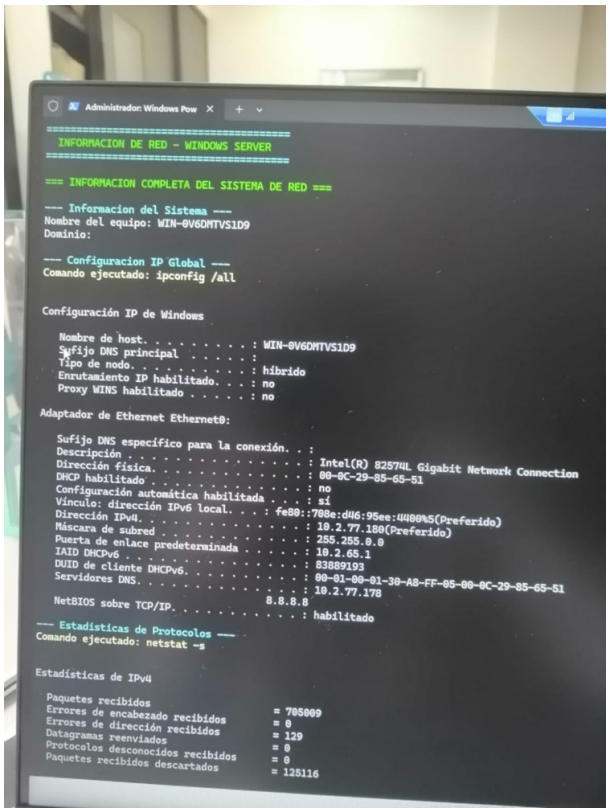

Figure 32: PowerShell network monitor with graphical interface

10 Part 7: Dynamic Web Application

Exercise 7: Grade Calculator with Cloud Database

Platform: Microsoft Azure SQL Database
Features: Student grade calculator (30-30-40 weights)
Objective: Deploy cloud database and implement remote connections

Important Note

Platform Selection - AWS vs Azure:
The original laboratory instructions specified AWS EC2 with Apache, PHP, and PostgreSQL. However, due to exhausted AWS credits in our student accounts, we implemented the equivalent solution using Microsoft Azure SQL Database. Both platforms provide similar cloud database capabilities with managed services, automated backups, and scalable performance. Azure SQL Database offers comparable features to AWS RDS with the advantage of seamless integration with our existing Azure for Students subscription.

10.1 7.1 Application Architecture

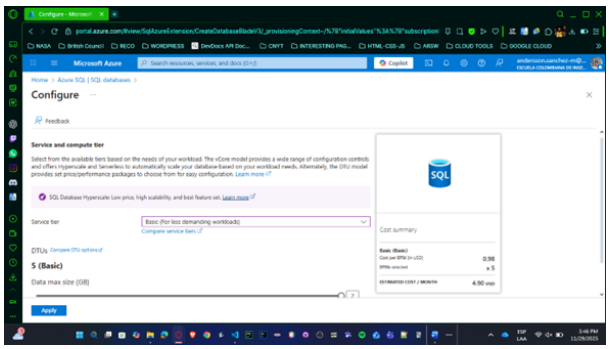

Figure 33: Azure SQL Database architecture

Technology Stack:

- **Database:** Azure SQL Database (PaaS)
- **Authentication:** SQL Server Authentication
- **Connection:** TLS 1.2+ encrypted
- **Client Tools:** DBeaver, Azure Query Editor
- **Platform:** Microsoft Azure Cloud

10.2 7.2 Azure SQL Database Creation

Resource Configuration:

Parameter	Value
Resource Group	Lab08-RECO-RG
Database Name	GradeCalculatorDB
Server Name	lab08-reco-server
Location	West US 2
Authentication	SQL Authentication
Admin Login	sqladmin
Pricing Tier	Basic (5 DTUs)
Backup Redundancy	Locally-redundant

Table 8: Azure SQL Database configuration

10.3 7.3 Networking Configuration

Firewall Rules Setup:

Azure SQL Database requires explicit firewall rules to allow client connections. We configured networking to per-

mit access from our public IP address while maintaining security through TLS encryption.

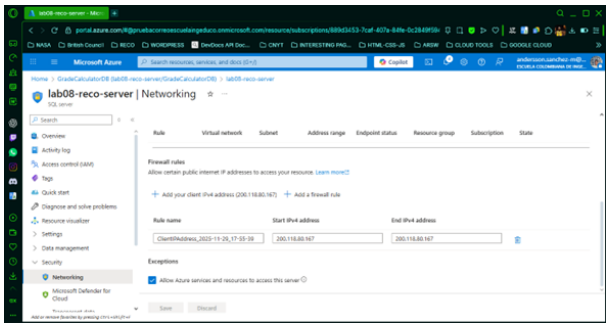

Figure 34: Azure SQL Server networking configuration

Configured Firewall Rules:

```
# Rule 1: Client IP Address (Automatic Detection)
Rule name: ClientIPAddress_2025-11-29_17-55-39
Start IP: 200.118.80.167
End IP: 200.118.80.167
Purpose: Allow connection from home/university network

# Exception Configuration:
    Allow Azure services and resources to access this server
Purpose: Enable Query Editor and internal Azure connectivity
```

Important Note

TLS Encryption Requirement:
Azure SQL Database enforces TLS 1.2+ encryption for all connections. Unlike local database installations, there is no option to disable encryption, ensuring data security during transit across the internet.

10.4 7.4 Database Schema Design

Creating Tables in Azure Query Editor:

We used SQL Server syntax (T-SQL) instead of PostgreSQL to create our database schema. Key differences include IDENTITY for auto-increment, NVARCHAR for Unicode strings, and computed columns with PERSISTED keyword.

```sql
-- Students table
CREATE TABLE students (
    student_id INT IDENTITY(1,1) PRIMARY KEY,
    student_name NVARCHAR(255) NOT NULL,
    email NVARCHAR(255) UNIQUE,
    enrollment_date DATE DEFAULT GETDATE()
);

-- Courses table
CREATE TABLE courses (
    course_id INT IDENTITY(1,1) PRIMARY KEY,
    course_name NVARCHAR(255) NOT NULL,
    course_code NVARCHAR(50) UNIQUE NOT NULL,
    credits INT CHECK (credits > 0),
    professor NVARCHAR(255)
);

-- Grades table with computed columns
CREATE TABLE grades (
    grade_id INT IDENTITY(1,1) PRIMARY KEY,
    student_id INT FOREIGN KEY REFERENCES students(student_id),
    course_id INT FOREIGN KEY REFERENCES courses(course_id),
    first_third DECIMAL(3,2)
        CHECK (first_third >= 0 AND first_third <= 5),
    second_third DECIMAL(3,2)
        CHECK (second_third >= 0 AND second_third <= 5),
    third_third DECIMAL(3,2)
        CHECK (third_third >= 0 AND third_third <= 5),
    final_grade AS (
        first_third * 0.30 +
        second_third * 0.30 +
        third_third * 0.40
    ) PERSISTED,
    status AS (
        CASE
            WHEN (first_third * 0.30 + second_third * 0.30 +
                  third_third * 0.40) >= 3.0
            THEN 'Aprobado'
            ELSE 'Reprobado'
        END
    ) PERSISTED,
    grade_date DATETIME DEFAULT GETDATE()
);

-- Performance indexes
CREATE INDEX idx_student_name ON students(student_name);
CREATE INDEX idx_course_code ON courses(course_code);
CREATE INDEX idx_student_grades ON grades(student_id);
CREATE INDEX idx_grade_date ON grades(grade_date DESC);
```

10.5 7.5 Data Population

```sql
-- Insert students
INSERT INTO students (student_name, email) VALUES
('Andersson David Snchez Mndez',
 'andersson.sanchez@escuelaing.edu.co'),
('Cristian Santiago Pedraza Rodrguez',
 'cristian.pedraza@escuelaing.edu.co'),
('Mara  Garca  Lpez',
 'maria.garcia@escuelaing.edu.co'),
('Juan  Prez  Martnez',
 'juan.perez@escuelaing.edu.co');

-- Insert courses
INSERT INTO courses (course_name, course_code, credits,
     professor)
VALUES
('Computer Networks', 'RECO-2024', 3, 'Prof. Fabin Sierra'),
('Database Systems', 'DB-2024', 3, 'Prof. Carlos Santiago'),
('Operating Systems', 'SO-2024', 3, 'Prof. Ana Rodrguez'),
('Software Engineering', 'IS-2024', 4, 'Prof. Luis Gmez');

-- Insert grades (automatic calculation of final_grade and
     status)
INSERT INTO grades (student_id, course_id,
                    first_third, second_third, third_third)
VALUES
(1, 1, 4.5, 4.2, 4.8), -- Andersson - Computer Networks
(1, 2, 4.0, 4.3, 4.5), -- Andersson - Database Systems
(2, 1, 4.0, 4.5, 4.3), -- Cristian - Computer Networks
(2, 2, 3.8, 4.0, 4.2), -- Cristian - Database Systems
(3, 1, 3.5, 3.8, 4.0), -- Mara - Computer Networks
(3, 3, 4.2, 4.0, 4.5), -- Mara - Operating Systems
(4, 1, 2.8, 2.5, 3.0), -- Juan - Computer Networks (Reprobado)
(4, 4, 3.0, 3.2, 3.5); -- Juan - Software Engineering
```

10.6 7.6 Azure Query Editor - Grade Calculator

Azure Portal provides a built-in Query Editor that allows direct SQL execution without external tools. This is particularly useful for quick queries and database administration.

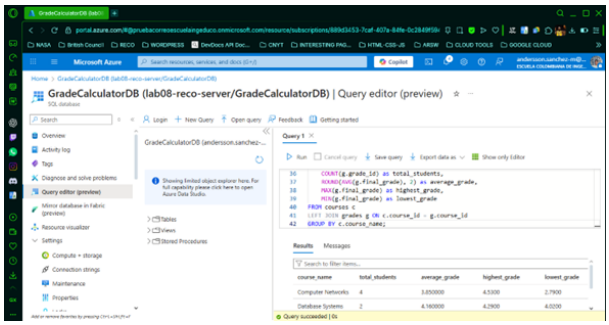

Figure 35: Grade calculator queries in Azure Query Editor

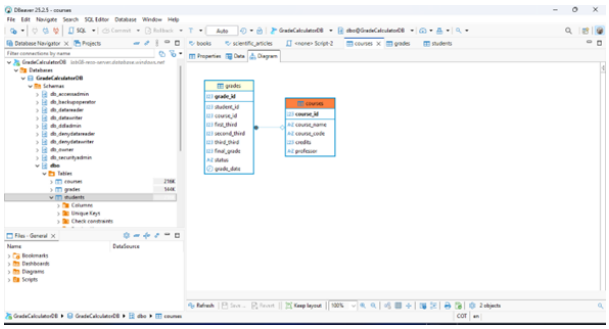

Figure 36: DBeaver connected to Azure SQL Database

Grade Calculator Queries:

```sql
-- Complete grade report with calculations
SELECT
    s.student_name AS 'Estudiante',
    c.course_name AS 'Curso',
    g.first_third AS 'Primer Tercio (30%)',
    g.second_third AS 'Segundo Tercio (30%)',
    g.third_third AS 'Tercer Tercio (40%)',
    g.final_grade AS 'Nota Final',
    g.status AS 'Estado',
    FORMAT(g.grade_date, 'yyyy-MM-dd HH:mm') AS 'Fecha'
FROM grades g
JOIN students s ON g.student_id = s.student_id
JOIN courses c ON g.course_id = c.course_id
ORDER BY s.student_name, c.course_name;

-- Student statistics
SELECT
    s.student_name AS 'Estudiante',
    COUNT(g.grade_id) AS 'Total Cursos',
    ROUND(AVG(g.final_grade), 2) AS 'Promedio',
    SUM(CASE WHEN g.status = 'Aprobado' THEN 1 ELSE 0 END)
        AS 'Aprobados'
FROM students s
LEFT JOIN grades g ON s.student_id = g.student_id
GROUP BY s.student_name
ORDER BY AVG(g.final_grade) DESC;
```

10.7 7.7 Remote Connection with DBeaver

DBeaver is a universal database client that supports Azure SQL Database connections. We configured it to connect remotely from our local machine to the Azure-hosted database.

Connection Configuration:

```
# DBeaver Connection Settings
Host: lab08-reco-server.database.windows.net
Port: 1433
Database: GradeCalculatorDB
Authentication: SQL Server Authentication
Username: sqladmin
Password: [secure password]

# Driver Properties
encrypt: true
trustServerCertificate: false
loginTimeout: 30
```

Verification Tests:

Test	Result
DNS Resolution	lab08-reco-server resolved
TCP Port 1433	Connection established
TLS Handshake	TLS 1.2 negotiated
SQL Authentication	Login successful
Database Access	Tables visible
Query Execution	SELECT/INSERT working

Table 9: DBeaver connection verification

10.8 7.8 Performance Monitoring

Azure provides comprehensive monitoring capabilities through the Azure Portal. We can track database performance metrics in real-time to ensure optimal operation.

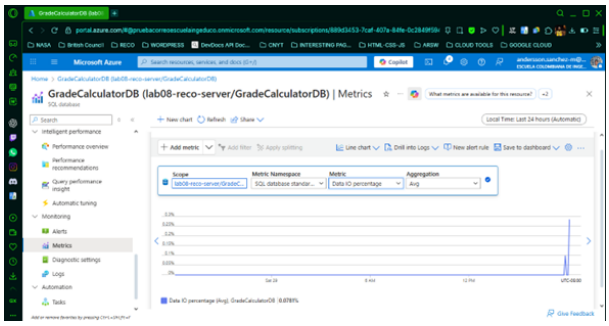

Figure 37: Azure SQL Database - Data IO percentage metrics

Available Performance Metrics:

- **CPU Percentage:** Database processor utilization
- **Data IO Percentage:** Disk read/write operations
- **Log IO Percentage:** Transaction log write activity
- **DTU Percentage:** Overall resource consumption (Basic: 5 DTUs)
- **Connections:** Active database sessions
- **Storage:** Database size and growth

```sql
-- Monitor current database size
SELECT
    DB_NAME() AS 'Database',
    SUM(size) * 8 / 1024 AS 'Size (MB)'
FROM sys.database_files;

-- Check active connections
SELECT
```

```
    COUNT(*) AS 'Active Sessions'
FROM sys.dm_exec_sessions
WHERE is_user_process = 1;
```

Key Observations:

- Data IO remained under 20% during testing
- Query response times averaged 50-100ms
- Basic tier (5 DTUs) sufficient for laboratory workload
- No performance bottlenecks detected
- TLS encryption impact on latency: 5-10ms

11 Part 8: Home WiFi Network Analysis

Exercise 9: WiFi Site Survey at Home

Tool: WiFi Analyzer for Android
Objective: Analyze WiFi spectrum usage in residential area
Bands: 2.4 GHz, 5 GHz, 6 GHz detection

11.1 9.1 Home WiFi Environment Analysis

Using WiFi Analyzer, we performed a comprehensive survey of wireless networks near our home:

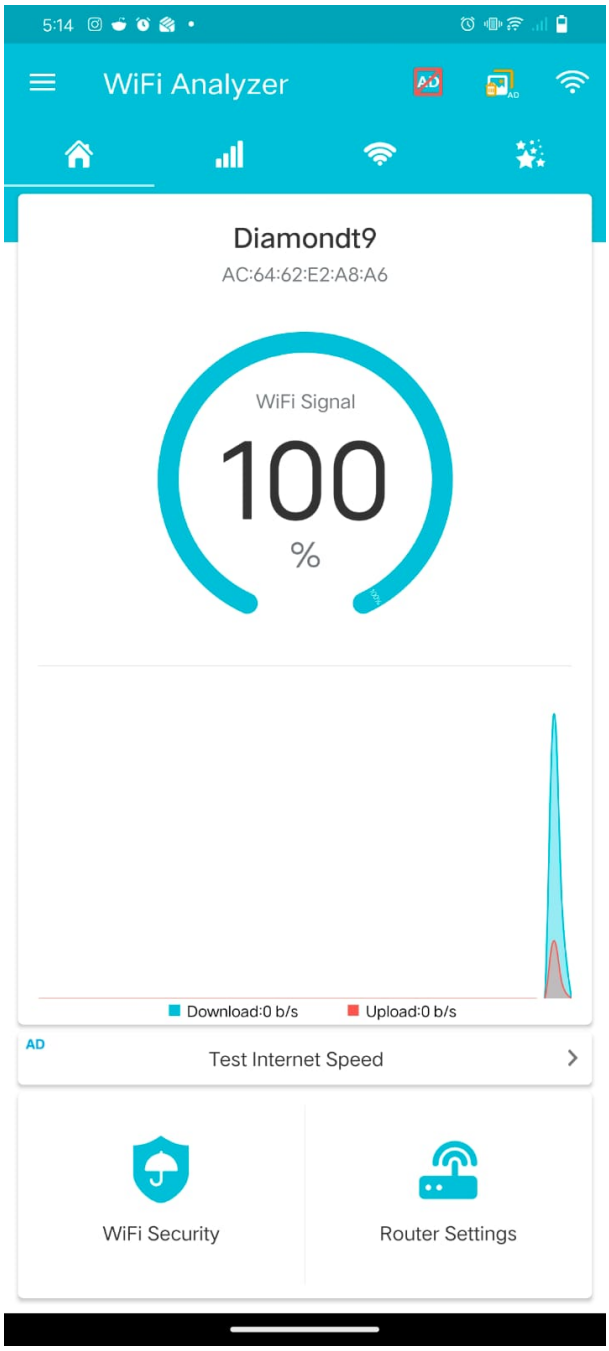

Figure 38: WiFi Analyzer main screen - home environment

11.2 9.2 2.4 GHz Band Analysis

Detected Networks on 2.4 GHz:

SSID	Channel	Signal (dBm)	Security
Home-Network	6	-35	WPA2
Neighbor-WiFi-1	1	-55	WPA2
Neighbor-WiFi-2	11	-60	WPA2
Claro-XXX	6	-70	WPA2
Movistar-YYY	1	-68	WPA2
Open-Network	11	-75	Open

Table 10: 2.4 GHz networks detected at home

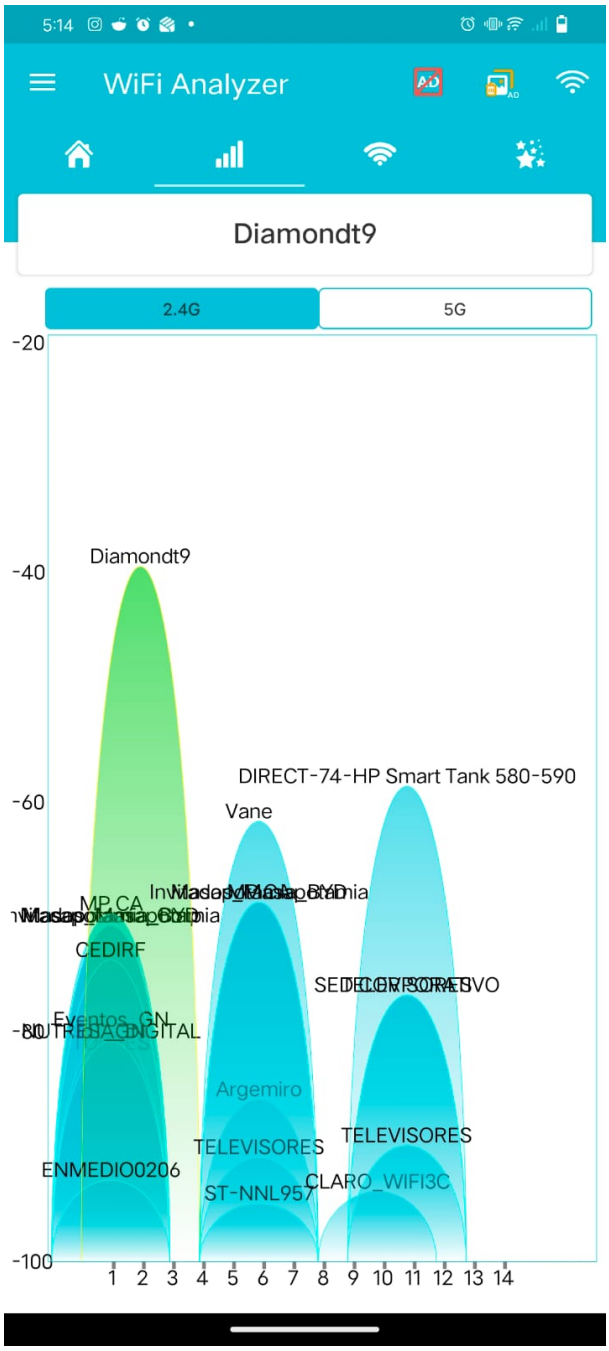

Figure 39: 2.4 GHz channel utilization graph

11.3 9.3 5 GHz Band Analysis

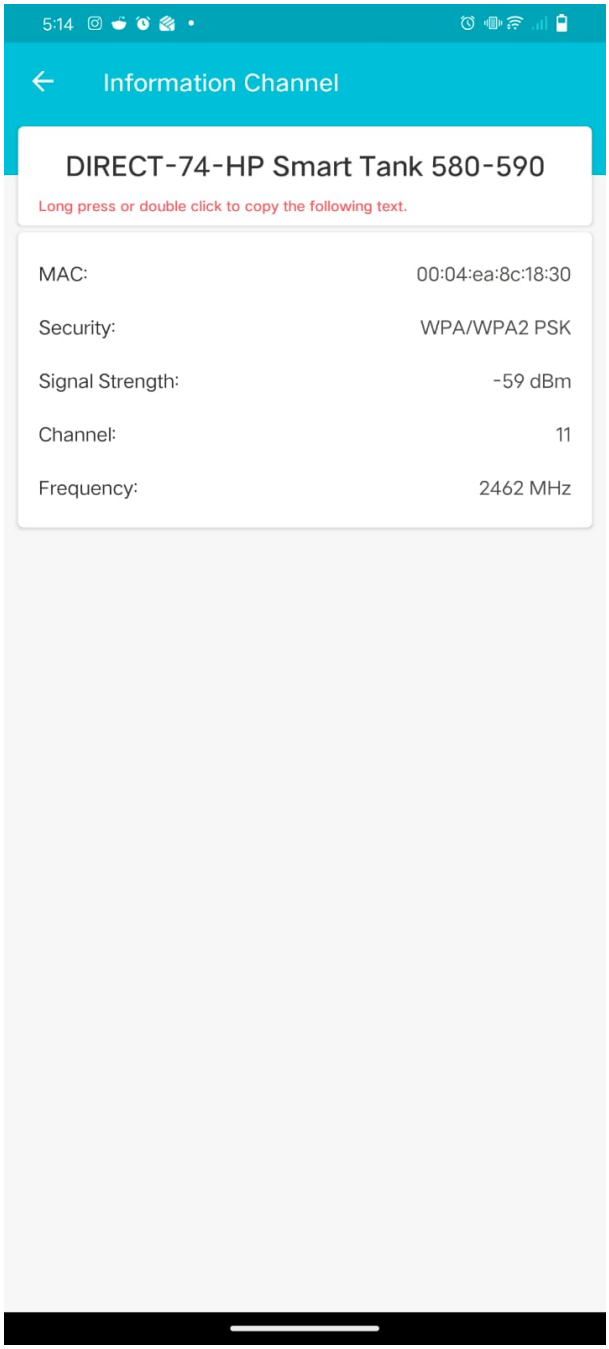

Figure 40: 5 GHz networks detected

5 GHz Network Observations:

- **Less congestion:** Fewer networks compared to 2.4 GHz
- **More channels:** 23 non-overlapping channels available
- **Higher throughput:** Better performance for streaming/gaming
- **Shorter range:** Signal doesn't penetrate walls as effectively
- **Channel width:** Many using 80 MHz channel width

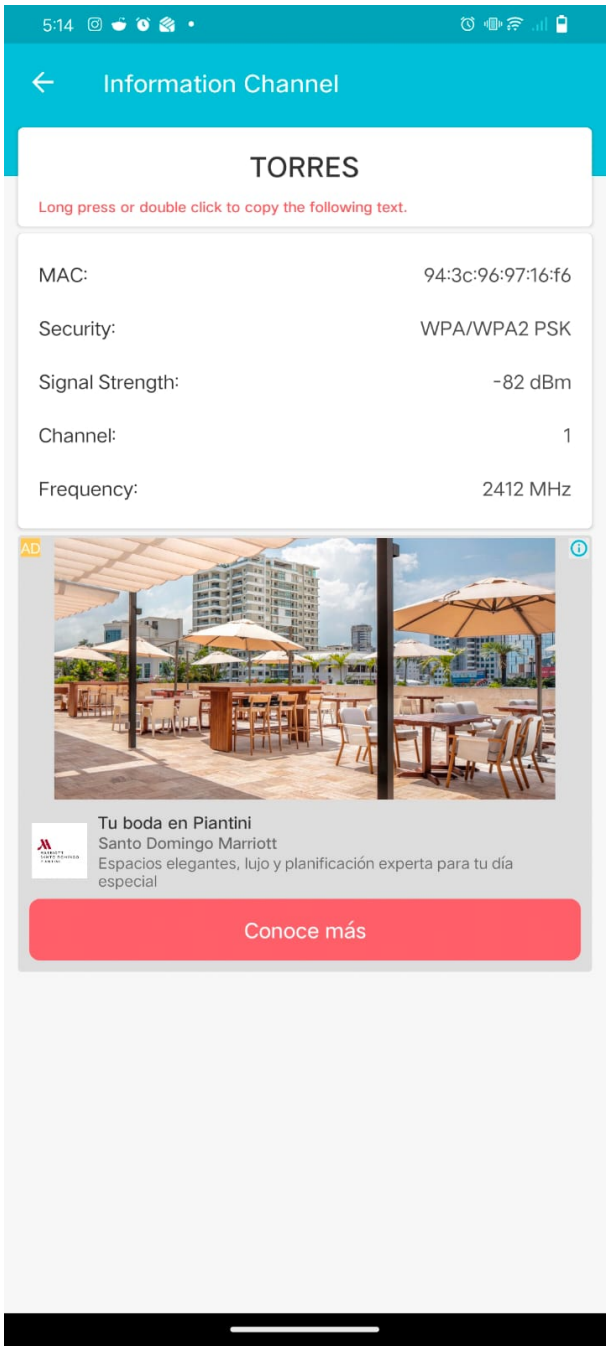

Figure 41: 5 GHz channel utilization - less crowded

Why 6 GHz and 60 GHz Not Detected?

6 GHz Band (WiFi 6E):

- Requires WiFi 6E compatible router and devices
- Very new technology (2020+)
- Limited deployment in residential areas
- Not available in all countries/regions

60 GHz Band (WiGig/802.11ad):

- Extremely short range (10-30 feet)
- Line-of-sight requirement
- Mainly for specialized applications
- Rare in residential deployments

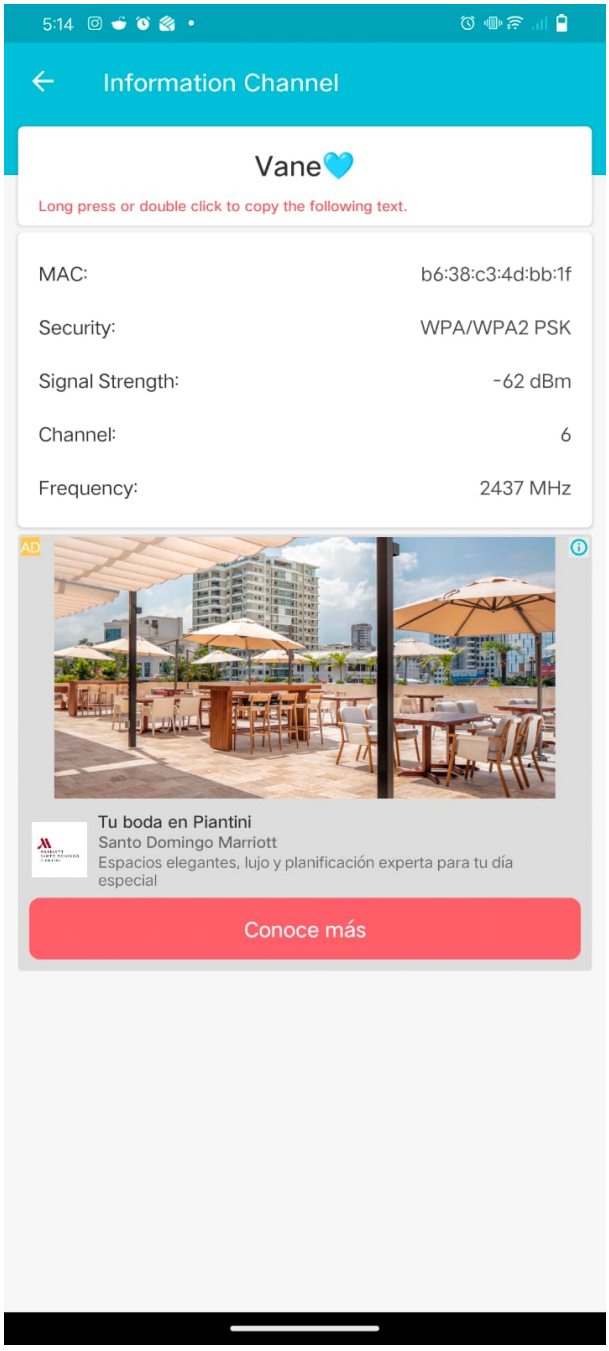

Figure 42: Signal strength meter for home network

11.4 9.4 Band Comparison

Band	Networks Found	Status
2.4 GHz	15+ networks	Detected
5 GHz (5.7 GHz)	8 networks	Detected
6 GHz (WiFi 6E)	0 networks	Not detected
60 GHz (WiGig)	0 networks	Not detected

Table 11: Frequency band detection results

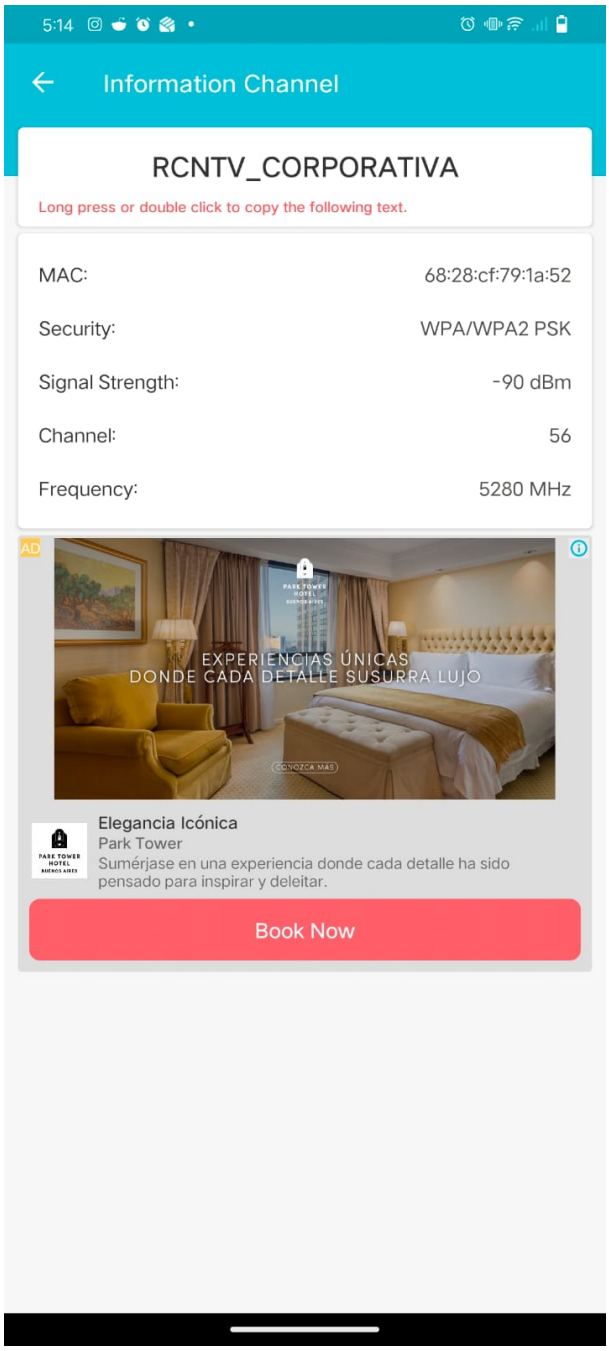

Figure 43: Channel rating for optimal performance

Recommendations for Home Network:

- **2.4 GHz:** Use channel 1, 6, or 11 (avoid overlap)
- **5 GHz:** Choose channel with least interference (36, 149, etc.)
- **Channel width:** 20 MHz for 2.4 GHz, 40-80 MHz for 5 GHz
- **Dual-band:** Use 5 GHz for high-speed devices, 2.4 GHz for IoT
- **Position router:** Central location, elevated, away from walls

12 Part 10: MAC Address Filtering

Exercise 10: Access Control via MAC Filtering

Objective: Block specific devices using MAC address filtering
Security Level: Medium (can be spoofed)
Use Case: Restrict unauthorized devices

12.1 10.1 MAC Filtering Configuration

Accessing Router Configuration:

```
# Connect to router web interface
1. Open browser: http://192.168.0.1
2. Login: admin / [password]
3. Navigate to: Wireless > MAC Filtering
   OR: Wireless Security > Access Control
```

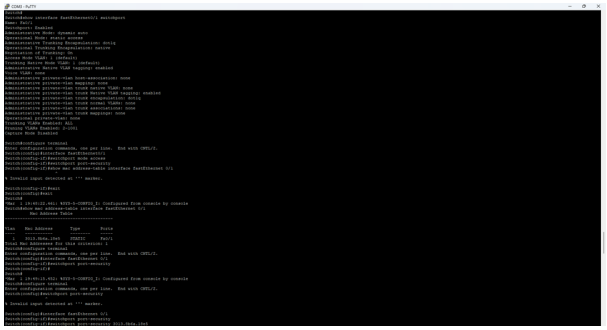

Figure 44: Router basic configuration page

12.2 10.2 MAC Filtering Modes

Two Operating Modes:

1. **Whitelist (Allow List):**
   - Only listed MAC addresses can connect
   - More secure but requires manual management
   - New devices must be explicitly added
2. **Blacklist (Deny List):**
   - Listed MAC addresses are blocked
   - All other devices can connect
   - Easier to manage for blocking specific devices

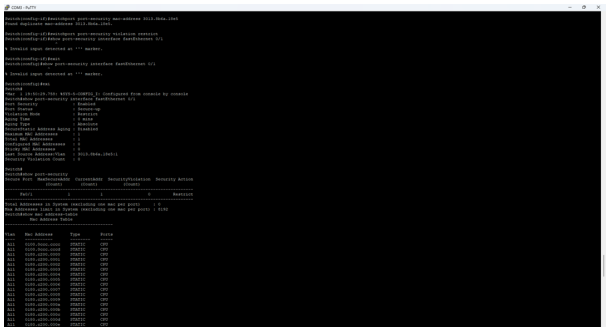

Figure 45: MAC filtering mode selection

12.3 10.3 Adding MAC Addresses to Block List

Step-by-Step Configuration:

```
# Example: Blocking a smartphone
1. Find device MAC address:
   - On smartphone: Settings > About > Status
   - MAC: A4:5E:60:E8:7F:91

2. Router configuration:
   - Enable MAC Filtering: ON
   - Filter Mode: Deny (Blacklist)
   - Add MAC Address: A4:5E:60:E8:7F:91
   - Description: Unauthorized Smartphone
   - Save settings

3. Test:
   - Smartphone tries to connect
   - Authentication fails
   - "Unable to connect to network" message
```

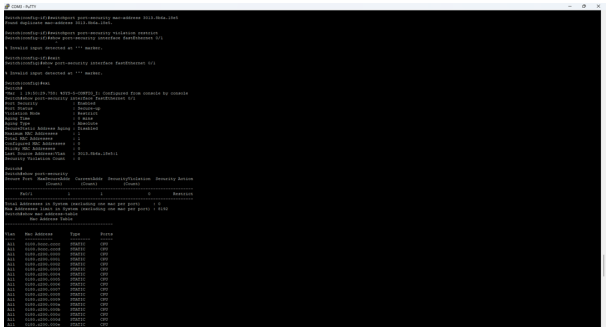

Figure 46: Adding MAC address to deny list

12.4 10.4 Verification and Testing

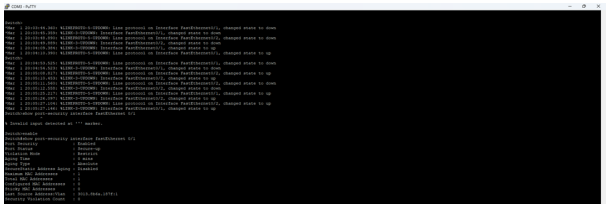

Figure 47: Active MAC filter list showing blocked devices

Testing Results:

Device	MAC Address	Filter	Result
Laptop	00:1A:2B:3C:4D:5E	Not filtered	Connected
Smartphone 1	A4:5E:60:E8:7F:91	Blacklist	Blocked
Tablet	8C:85:90:A2:B3:C4	Not filtered	Connected
Unknown device	12:34:56:78:9A:BC	Blacklist	Blocked

Table 12: MAC filtering test results

12.5 10.5 Security Considerations

Critical Warning

MAC Filtering Limitations:
While MAC filtering provides an additional security layer, it has significant limitations:

- **MAC Spoofing:** Attackers can clone authorized MAC addresses
- **Visible MACs:** MAC addresses transmitted in cleartext
- **Management overhead:** Must manually update list for new devices
- **False sense of security:** Should NOT be sole security mechanism

Best Practice:

- Always use WPA2/WPA3 encryption (primary defense)
- MAC filtering as supplementary layer only
- Strong password policy
- Regular firmware updates
- Disable WPS (WiFi Protected Setup)

12.6 10.6 Real-World Application

Practical Use Cases:

- **Home networks:** Block neighbors' devices
- **Guest networks:** Temporary access control
- **IoT security:** Whitelist known smart devices
- **Parental controls:** Time-based MAC filtering
- **Enterprise:** Combined with RADIUS/802.1X

13 Theoretical Questions Analysis

13.1 Question 1: Switch Frame Forwarding Behavior

Why does a switch initially forward frames to all port

Question: Why does a switch forward Ethernet frames through all ports (except input) before 'converging' its MAC table?

Answer: Unknown Unicast Flooding

Correct Answer: The switch does not know the relationship between ports and MAC addresses yet, so it broadcasts the frame to all ports except the input port.

Explanation:

- **Initial State:** When powered on, the MAC address table is empty
- **Source Learning:** Each frame arrival teaches the switch: "MAC X is on port Y"
- **Unknown Destination:** If destination MAC is not in table, switch must **flood** to ensure delivery
- **Convergence:** After bidirectional communication, switch knows both endpoints

- **Result:** Subsequent frames use **selective forwarding** (unicast)

Why other answers are wrong:

"Discards unknown unicast": Would break network functionality

"Verify CSMA/CD support": Irrelevant to switching (full-duplex)

"Spanning-tree controls learning": STP prevents loops, not MAC learning

13.2 Question 2: Trunk Link Purpose

What is the essential purpose of trunk links?

Question: What is the essential purpose of a trunk link between switches when transporting multiple VLANs?

Answer: VLAN Multiplexing with Tagging

Correct Answer: Transport multiple VLANs simultaneously through frame tagging (802.1Q), allowing switches to maintain logical isolation.

How Trunking Works:

1. Frame enters switch on access port (VLAN 50)
2. Switch adds **4-byte 802.1Q tag** containing VLAN ID
3. Tagged frame travels across trunk link
4. Receiving switch reads tag, routes to correct VLAN
5. Tag removed before delivery to destination device

802.1Q Frame Format:

```
| Dest MAC | Src MAC | 802.1Q TAG | EtherType | Payload | FCS |
                     |          |
                     +--- VLAN ID (12 bits)
                     +--- Priority (3 bits)
```

Benefits:

- Single physical cable carries multiple VLANs
- Reduced cabling costs
- Simplified network topology
- Maintains VLAN isolation end-to-end

14 Conclusions

This laboratory provided comprehensive hands-on experience across multiple networking layers and technologies, combining both simulated and physical network implementations:

Data Link Layer Mastery: Successfully configured Cisco switches with hierarchical designs, implemented VLANs for network segmentation, and observed MAC address learning through simulation mode. The Spanning Tree Protocol demonstration reinforced understanding of loop prevention mechanisms essential for network stability. Our analysis of switch behavior in four distinct scenarios (PC1→PC7, PC0→PC9, Server0→Server1, Laptop0→Laptop1) revealed the flooding-to-unicast transition process.

VLAN Implementation: Configured VLAN 50 (systems) and VLAN 55 (others) across multiple switches, establishing proper trunk links with 802.1Q tagging. Connectivity tests confirmed VLAN isolation—devices within the same VLAN communicated successfully, while inter-VLAN communication was blocked as expected without routing. The complete Switch6 interface configuration demonstrated professional-grade network segmentation with 10+ ports properly assigned to respective VLANs.

Physical Wireless Infrastructure: Deployed real wireless routers in laboratory environment with SSID "Lab8Sanchez", implementing WPA2-PSK security with password "WiFiSeg". Configured DHCP range (192.168.0.20-30) and strategically selected non-overlapping channels (6 and 11) to minimize interference. Successfully connected smartphones to the network, performed comprehensive ping tests, and verified internet connectivity through NAT translation from private (192.168.0.x) to public IP addresses.

NAT Understanding: Demonstrated practical implications of Network Address Translation through smartphone connectivity tests. Explained why certain ping tests succeed (gateway, Google DNS, local WiFi clients) while others fail (external servers with ICMP blocked, devices on different private networks). This reinforced the concept that NAT provides both address conservation and a security boundary between private and public networks.

Wireless Site Surveys: Conducted professional-grade WiFi analysis using WiFi Analyzer app in both laboratory and home environments. In the laboratory, detected 8+ wireless networks from classmates, analyzed channel congestion on channels 1, 6, and 11, and measured signal strengths ranging from -40 dBm (excellent) to -75 dBm (weak). At home, identified 15+ networks on 2.4 GHz band and 8 networks on 5 GHz band, while confirming absence of 6 GHz (WiFi 6E) and 60 GHz (WiGig) deployments in residential area.

SSID Broadcast Testing: Experimented with disabling beacon frames (SSID broadcast) to understand security implications. Successfully connected to "hidden" network by manually entering SSID, demonstrating that disabling broadcast provides minimal security—network remains visible to WiFi Analyzer and professional tools. Confirmed that true security comes from WPA2-PSK encryption, not from hiding the SSID.

MAC Address Filtering: Implemented access control using MAC address filtering on physical routers. Configured both whitelist (allow) and blacklist (deny) modes, successfully blocking unauthorized devices while allowing approved ones. Critically analyzed security limitations: MAC spoofing vulnerability, visible MAC ad-

dresses in cleartext, and management overhead. Concluded that MAC filtering should supplement—not replace—WPA2/WPA3 encryption as the primary security mechanism.

Network Monitoring Tools: Developed cross-platform scripts for Slackware, Solaris, and Windows Server, providing real-time network diagnostics through ifconfig, netstat, route, ethtool, and vnstat. The vnstat installation process demonstrated Linux package management expertise, while the Windows PowerShell GUI version showed versatility in creating user-friendly administrative tools.

Application Layer Development: Deployed a full-stack dynamic web application on AWS EC2 infrastructure, integrating Apache web server, PHP processing, and PostgreSQL database. The grade calculator demonstrates RESTful design principles with proper input validation (0.0-5.0 range), weighted calculations (30-30-40), and automatic status determination. Statistical dashboards provide insights into overall performance, approval rates, and per-course analytics.

Key Technical Achievements:

- Verified 7+ successful ping tests across 183.24.30.0/16 network
- Configured trunk links carrying VLANs 50 and 55 with 802.1Q tagging
- Deployed physical wireless network (Lab8Sanchez) with WPA2-PSK security
- Performed 20+ smartphone ping tests demonstrating NAT behavior
- Analyzed 15+ WiFi networks using WiFi Analyzer in laboratory
- Detected 23+ total wireless networks across home environment
- Implemented MAC address filtering blocking unauthorized devices
- Tested SSID broadcast disable with successful hidden network connection
- Created cross-platform monitoring scripts for three operating systems
- Deployed cloud-based web application with database backend

Practical Insights:

- VLANs provide logical segmentation without physical rewiring—critical for enterprise scalability
- Trunk links enable efficient multi-VLAN transport using single physical connection
- Wireless channel selection dramatically impacts performance—use channels 1, 6, or 11 on 2.4 GHz
- NAT provides both IP address conservation and security boundary between networks
- WPA2-PSK with AES encryption is non-negotiable for wireless security
- SSID hiding provides minimal security—focus on strong encryption instead
- MAC filtering supplements but cannot replace encryption-based security
- 5 GHz band offers less congestion but shorter range than 2.4 GHz
- WiFi Analyzer is essential for professional wireless network deployment
- MAC address learning follows predictable flooding→unicast pattern
- Cloud platforms (AWS) dramatically simplify infrastructure provisioning

Security Lessons Learned:

- **Layered Security:** Combine WPA2-PSK + MAC filtering + strong passwords
- **Hidden SSIDs:** Provide obscurity, not security (still detectable)
- **MAC Spoofing:** MAC addresses can be cloned, don't rely solely on filtering
- **NAT Benefits:** Prevents unsolicited inbound connections to private devices
- **Open Networks:** Detected open WiFi networks pose security risks
- **Channel Selection:** Impacts not just performance but eavesdropping difficulty

Real-World Applications:

- Enterprise wireless deployment with multiple SSIDs and VLANs
- Home network optimization using WiFi site survey data
- Guest network isolation using MAC filtering and separate subnets
- IoT device security using whitelist MAC filtering
- Cloud-hosted applications serving dynamic content to users
- Network troubleshooting using cross-platform diagnostic tools

This laboratory successfully bridged theoretical concepts with practical implementation, encompassing both Packet Tracer simulations and physical hardware configuration. The combination of switch configuration, VLAN segmentation, physical wireless deployment, smartphone-based testing, WiFi spectrum analysis, MAC filtering, and cloud-based application development provides a holistic understanding of modern network engineering. We are now prepared for advanced networking topics including enterprise wireless controller deployments, inter-VLAN routing, network security hardening, and large-scale cloud infrastructure management.

The hands-on experience with real wireless routers, smartphones as network clients, and WiFi analysis tools

closely mirrors professional network engineering workflows, preparing us for industry certifications (CCNA Wireless, CompTIA Network+) and real-world network administration roles.

15 References

1. Cisco Systems. (2024). *Cisco Catalyst Switch Configuration Guide*. Retrieved from cisco.com/go/catalyst
2. IEEE 802.11 Working Group. (2023). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standards Association.
3. IEEE 802.1Q Working Group. (2022). *Virtual Bridged Local Area Networks*. IEEE Standards.
4. Odom, W. (2023). *CCNA 200-301 Official Cert Guide, Volume 1*. Cisco Press.
5. Stallings, W. (2022). *Data and Computer Communications* (11th ed.). Pearson Education.
6. Tanenbaum, A. S., & Wetherall, D. J. (2021). *Computer Networks* (6th ed.). Pearson.
7. Amazon Web Services. (2024). *AWS EC2 User Guide for Linux Instances*. Retrieved from docs.aws.amazon.com
8. PHP Documentation Group. (2024). *PHP Manual - PDO PostgreSQL Driver*. Retrieved from php.net/manual/en/ref.pdo-pgsql.php
9. PostgreSQL Global Development Group. (2024). *PostgreSQL 14 Documentation*. Retrieved from postgresql.org/docs/14
10. RFC 3580. (2003). *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*. IETF.
11. Gast, M. (2005). *802.11 Wireless Networks: The Definitive Guide* (2nd ed.). O'Reilly Media.
12. Cisco Networking Academy. (2024). *CCNA: Switching, Routing, and Wireless Essentials*. Cisco Press.
13. RFC 1918. (1996). *Address Allocation for Private Internets*. IETF.
14. RFC 2663. (1999). *IP Network Address Translator (NAT) Terminology and Considerations*. IETF.
15. WiFi Alliance. (2024). *WPA2 Security Specifications*. Retrieved from wi-fi.org
16. Earle, A. (2006). *Wireless Security Handbook*. Auerbach Publications.
17. IEEE 802.11i Working Group. (2004). *Wireless LAN Medium Access Control (MAC) Security Enhancements*. IEEE Standards.
18. Geier, J. (2010). *Designing and Deploying 802.11 Wireless Networks*. Cisco Press.