

# ChatGPT

## vlan - through routing and switch.pdf

PDF

in this document, the process of creation of VLAN through router and switches would be discussed, in which get me the steps to create the VLAN using switch in two ways (for inter-VLAN communication)

1. VLAN using Multilayered Switch through Access Switch
2. VLAN directly using Access Switch

In your document <sup>vlan - through routing and swit...</sup>, inter-VLAN routing is explained using both Router-on-a-Stick and a Multilayer Switch. Based on that same VLAN plan (10 → 10.0.1.0/24, 20 → 10.0.2.0/24, 30 → 10.0.3.0/24, 40 → 10.0.4.0/24), here are the **step-by-step procedures** for creating VLANs using a switch in the two designs you asked.

## 1 VLAN Using Multilayer Switch Through Access Switch

(Enterprise Design – Access Layer + Distribution Layer)

### Scenario

Imagine a college building.

Each classroom has many PCs connected to a normal Layer-2 access switch.

All those access switches connect to one central multilayer switch in the server room.

The multilayer switch performs inter-VLAN routing.

So traffic flow becomes:

PC → Access Switch → Trunk → Multilayer Switch → Routing → Back

## Step 1 — Configure VLANs on the Multilayer Switch

```
bash

enable
configure terminal
vlan 10
vlan 20
vlan 30
vlan 40
exit
```

This creates the VLAN database.

## Step 2 — Create SVI (Gateway Interfaces) on Multilayer Switch

```
bash

interface vlan 10
ip address 10.0.1.1 255.255.255.0
no shutdown
exit

interface vlan 20
ip address 10.0.2.1 255.255.255.0
no shutdown
exit

interface vlan 30
ip address 10.0.3.1 255.255.255.0
no shutdown
exit

interface vlan 40
ip address 10.0.4.1 255.255.255.0
no shutdown
exit
```

Each SVI becomes the **default gateway** for that VLAN.

---

## Step 3 — Enable Layer 3 Routing (Most Important)

```
bash

ip routing
```

Without this, VLANs will not communicate.

---

## Step 4 — Configure Trunk Port Between Access Switch and Multilayer Switch

On the multilayer switch:

```
bash

interface g0/1 (or whichever port connects to access switch)
switchport mode trunk
```

Now VLAN tags can travel between switches.

---

## Step 5 — Configure Access Switch

On the access switch:

### Create VLANs (must match multilayer switch)

```
bash

vlan 10
vlan 20
vlan 30
vlan 40
```

## Assign ports to VLANs

```
bash

interface range fa0/1-2
switchport mode access
switchport access vlan 10

interface range fa0/3-4
switchport mode access
switchport access vlan 20
```

(Repeat for VLAN 30 and 40)

## Configure trunk toward multilayer switch

```
bash

interface fa0/24
switchport mode trunk
```

## Step 6 — Configure PCs

Example:

VLAN 10 PC

IP: 10.0.1.2

Subnet: 255.255.255.0

Gateway: 10.0.1.1

Now VLAN 10 can talk to VLAN 20 through the multilayer switch.

## 2 VLAN Directly Using Multilayer Switch (No Access Switch)

### Scenario

Small office. Only 8–20 PCs.

You connect all PCs directly into the multilayer switch.

PC → Multilayer Switch → Routing happens internally

## Step 1 — Create VLANs

```
bash

enable
configure terminal
vlan 10
vlan 20
vlan 30
vlan 40
```

---

## Step 2 — Assign Switch Ports to VLANs

```
bash

interface range fa0/1-2
switchport mode access
switchport access vlan 10

interface range fa0/3-4
switchport mode access
switchport access vlan 20

interface range fa0/5-6
switchport mode access
switchport access vlan 30

interface range fa0/7-8
switchport mode access
switchport access vlan 40
```

---

## Step 3 — Create SVI Interfaces

```
bash

interface vlan 10
ip address 10.0.1.1 255.255.255.0
no shutdown

interface vlan 20
ip address 10.0.2.1 255.255.255.0
no shutdown

interface vlan 30
ip address 10.0.3.1 255.255.255.0
no shutdown

interface vlan 40
ip address 10.0.4.1 255.255.255.0
no shutdown
```

---

## Step 4 — Enable Routing

```
bash

ip routing
```

## Step 5 — Configure PC Gateways

VLAN 10 → 10.0.1.1

VLAN 20 → 10.0.2.1

VLAN 30 → 10.0.3.1

VLAN 40 → 10.0.4.1

Now inter-VLAN communication works internally inside the same switch.



### Conceptual Difference

In the first design, the multilayer switch acts like a **central router for multiple access switches**.

In the second design, it acts as both **access switch and router at the same time**.

If you want, next I can explain the **packet flow difference** between these two designs when VLAN 10 pings VLAN 20 — step by step like a troubleshooting engineer.

Sources

is there any possibility for inter-vlan communication only with layer2 switch(access switch)?

Short answer — **no, pure inter-VLAN routing cannot happen using only a Layer-2 (access) switch**.

Now let's understand this properly with a practical scenario.

Imagine you have:

- VLAN 10 → 10.0.1.0/24
- VLAN 20 → 10.0.2.0/24

Both VLANs are created on a normal Layer-2 switch.

Now a PC in VLAN 10 (10.0.1.2) wants to ping a PC in VLAN 20 (10.0.2.2).

Here's what happens internally:

1. The PC checks: "Is 10.0.2.2 in my network 10.0.1.0/24?"
2. It realizes it is not.
3. So it sends the packet to its default gateway.

Now here's the problem:

A Layer-2 switch **does not act as a gateway**.

It does not have:

- A routing table
- The **ip routing** capability

- Layer-3 forwarding logic

It only forwards frames based on MAC addresses **within the same VLAN**.

So what will happen?

The packet reaches the switch, but the switch has no idea how to move traffic from VLAN 10 to VLAN 20 because:

- VLANs are separate broadcast domains
- The switch cannot rewrite Layer-3 headers
- It cannot route between IP networks

The ping will fail.

## Why it is technically impossible

Inter-VLAN communication requires:

- Removing VLAN tag
- Checking destination IP
- Looking up routing table
- Forwarding to another VLAN
- Re-tagging frame

These are Layer-3 operations.

A Layer-2 switch simply doesn't have that capability.

## Are there any tricks or exceptions?

There are only two situations where people get confused:

### 1. If the "Layer-2 switch" is actually a Layer-3 switch

Some switches look like access switches but support **ip routing**. Those are multilayer switches.

### 2. If you connect two VLANs into the same IP subnet

This defeats the purpose of VLANs and is not real inter-VLAN routing.

But with a true access switch (like 2960 in Packet Tracer),

👉 **inter-VLAN routing is not possible without a router or multilayer switch.**

## Final Clear Answer

Using only a Layer-2 access switch:

- You can create VLANs ✓
- You can isolate traffic ✓

- You cannot route between VLANs ❌

You must introduce either:

- A router (Router-on-a-Stick)
- Or a multilayer (Layer-3) switch

If you want, I can explain what exactly changes inside the hardware when a switch becomes Layer-3 capable.

how to reset a switch and a multilayered switch to its original configuration?

Good question — this is something you'll use a lot in labs.

I'll explain it in a practical scenario style so you understand what's happening internally.

Imagine you finished a VLAN lab. Your switch now has:

- Multiple VLANs
- Trunk ports
- SVI interfaces
- Maybe **ip routing** enabled
- Passwords configured

Now you want the switch to behave like it just came out of the box.

There are **two things you must remove**:

1. The startup configuration (saved settings)
2. The VLAN database file

If you don't remove both, the switch will not fully reset.

## ◆ Resetting a Normal Layer-2 Switch (Access Switch)

### Step 1 — Erase Startup Configuration

Enter privileged mode:

```
bash
enable
write erase
```

or

```
bash

erase startup-config
```

This deletes the saved configuration from NVRAM.

## Step 2 — Delete VLAN Database (Very Important)

VLANs are stored separately in a file called `vlan.dat`.

```
bash

delete flash:vlan.dat
```

When prompted, press Enter to confirm.

If you skip this step, your VLANs will still exist after reload.

## Step 3 — Reload the Switch

```
bash

reload
```

When asked to save configuration:

Type **no**

After reboot, the switch will be factory default.

You'll see:

```
pgsql

Would you like to enter the initial configuration dialog?
```

That means reset was successful.

## ◆ Resetting a Multilayer Switch (Layer-3 Switch)

The process is almost the same, but multilayer switches may also have:

- SVI interfaces
- **ip routing**
- Routing protocols

Still, the reset process is identical.

## Step 1 — Erase Startup Config

```
bash  
  
enable  
write erase
```

---

## Step 2 — Delete VLAN File

```
bash  
  
delete flash:vlan.dat
```

---

## Step 3 — Reload

```
bash  
  
reload
```

Answer **no** when asked to save.

---

## What Happens Internally?

- **write erase** → clears NVRAM (startup config)
- **delete vlan.dat** → clears VLAN database
- **reload** → reboots and loads blank config

Without **reload**, changes won't take effect.

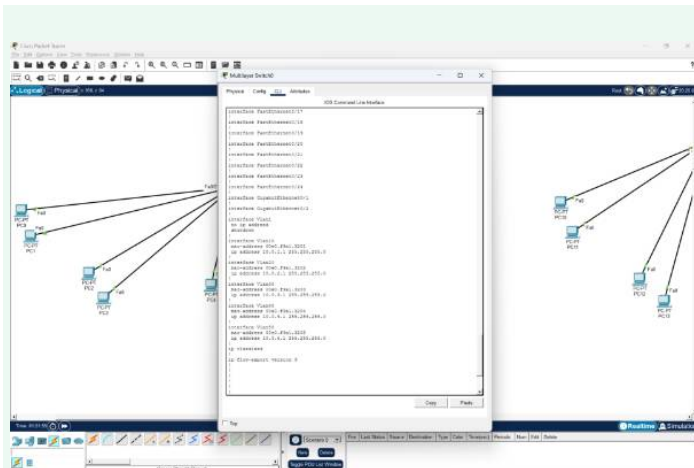
---

## If You're Locked Out (Password Recovery Scenario)

If the switch has an unknown password, the process is different (ROMMON mode, interrupt boot, etc.). Tell me if you want that procedure as well.

---

If you want, I can also give you a quick verification checklist to confirm the reset worked properly.



even after using the command "do delete flash:vlan.dat" - this configuration is not deleted??

Good — your screenshot clearly shows what's happening.

You deleted **vlan.dat**, but the configuration you're seeing is **not coming from vlan.dat**.

Look carefully at your CLI window.

You still have:

```
nginx
interface vlan10
 ip address 10.0.1.1 255.255.255.0
interface vlan20
 ip address 10.0.2.1 255.255.255.0
...
ip routing
```

These are **SVI configurations** stored inside the **startup-config**, not inside **vlan.dat**.

So deleting **vlan.dat** only removes the VLAN database (VLAN numbers).

It does NOT remove:

- Interface VLAN IP addresses
- **ip routing**
- Any Layer 3 configuration

That's why you still see the configuration.

## Why This Happens

There are two different storage areas:

- 1 **startup-config** → stores full configuration
- 2 **vlan.dat** → stores VLAN database

You deleted only **vlan.dat**.

But your startup-config still exists.

# Proper Full Reset (Multilayer Switch – Packet Tracer)

Do this exactly:

## Step 1

```
bash
enable
write erase
```

(or `erase startup-config`)

Wait for confirmation.

## Step 2

```
bash
delete flash:vlan.dat
```

Press Enter to confirm.

## Step 3 (VERY IMPORTANT)

```
bash
reload
```

When asked:

```
nginx
System configuration has been modified. Save? [yes/no]
```

Type:

```
perl
no
```

After reload, the switch will boot clean.

## Important Packet Tracer Detail

If you:

- Delete `vlan.dat`

- BUT do NOT reload

The running-config is still active in RAM.

It will look like nothing changed.

Reload is mandatory.

## Quick Verification After Reload

Run:

```
bash
show running-config
```

You should NOT see:

- interface vlan10
- ip routing
- any custom VLANs

Only default VLAN 1 should exist.

## Why You Still Saw VLAN Interfaces

Because SVI interfaces are Layer 3 configuration.

They are stored inside startup-config.

Deleting vlan.dat does not remove SVI definitions.

If it still doesn't reset after reload, tell me:

- Did you use **write erase**?
- Did you answer **no** when asked to save?
- Did you actually see the switch reboot?

We'll debug it like a real engineer.