VLANs (Part-2)

Trunk Ports, 802.1Q, and Inter-VLAN Routing

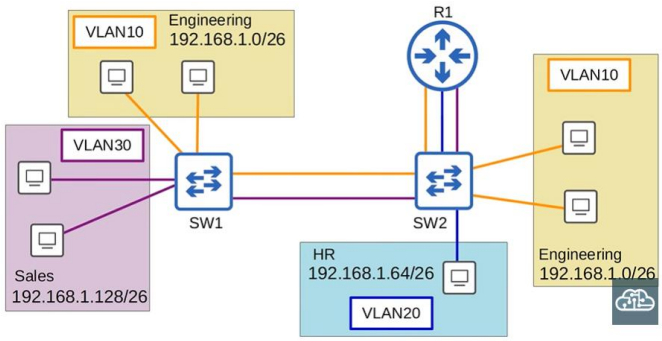
- Topics covered:
  - What is a trunk port
  - Purpose of trunk ports
  - 802.1Q encapsulation
  - Trunk port configuration
  - Router on a stick
- This is important for CCNA
- It is a more efficient way of performing inter-VLAN routing

---

What is a Trunk Port?

- An access port belongs to a single VLAN
- Trunk ports carry traffic from multiple VLANs
- Traffic is carried on a single interface

Network Topology (Initial)



- Two switches are used
- VLAN 10 is split between two switches
- Departments are not always in one location
- Engineers can be on different floors

- Only access ports are used initially

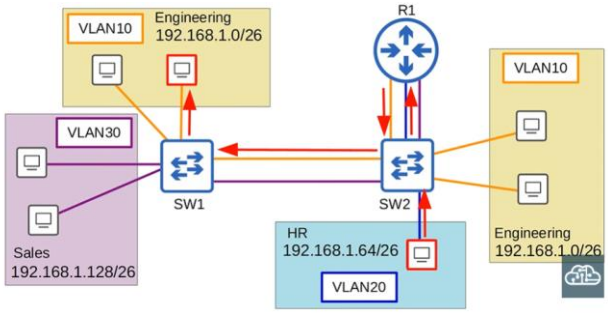
🔹 Why Multiple Links Are Used (Before Trunking)

- One link for VLAN 10 between switches
- One link for VLAN 30 between switches
- VLAN 20 has no link between switches
- No PCs in VLAN 20 are connected to SW1
- PCs in VLAN 20 can still reach other VLANs
- Router performs inter-VLAN routing

🔹 Inter-VLAN Routing Example

- PC in VLAN 20 sends traffic to VLAN 10 PC
- Destination MAC address is R1
- R1 is the default gateway

PC (VLAN20)→SW2→R1→SW2 (traffic now in VLANs10)→SW1→ PC (VLAN10)



- Traffic arrives at switch 2 on VLAN 10 interface
- Traffic is now in VLAN 10
- Switch forwards traffic to switch one
- Switch one forwards traffic to destination PC

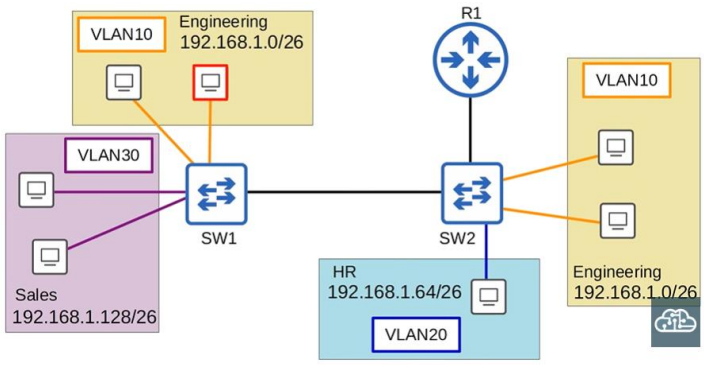
🔹 Problem With Separate Links

- Possible in small networks
- Not viable when VLANs increase

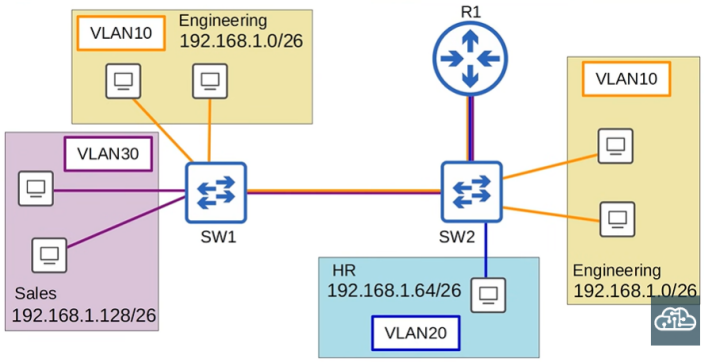
- Wasted interfaces
- Routers may not have enough interfaces

Trunk Port Solution

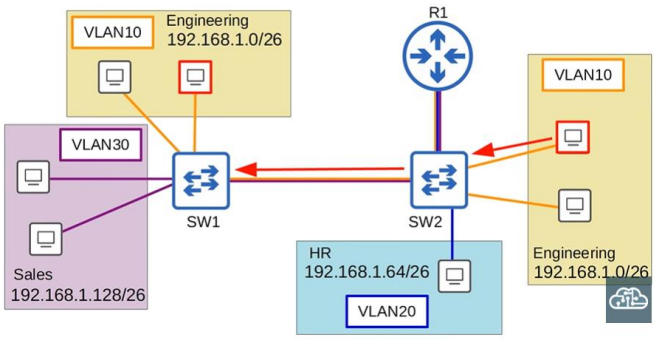
- Trunk ports carry traffic from multiple VLANs
- Single physical connection
- Different from access ports
- Access ports belong to a single VLAN



- For Making It Clear



VLAN Tagging on Trunk Ports



- Switches tag frames sent over trunk links
- Tag identifies which VLAN traffic belongs to
- Another name for trunk port is tagged port
- Another name for access port is untagged port

Access Port vs Trunk Port

- Frames sent over access ports are not tagged
- Interface belongs to a single VLAN
- If a frame arrives on VLAN 10 port
- Switch knows the frame is in VLAN 10

Trunking Protocols (ISL and IEEE 802.1Q)

- **VLANs Tagging** (Trunking Protocols)
  - There are two main trunking protocols
  - ISL interlink and IEEE 802.1Q
  - Usually we call 802.1Q as dot1q
  - ISL is an old Cisco proprietary protocol
  - Created before the industry standard IEEE 802.1Q
  - IEEE 802.1Q is an industry standard protocol
  - Created by the IEEE institute of electrical and electronics engineers
  - Remember IEEE
  - How about IEEE 802.3?
  - That's Ethernet, another industry standard protocol
  - You will probably never use ISL in the real world
  - Even modern Cisco equipment doesn't support it
  - For the CCNA, you only need to learn 1Q
  - You should know what ISL is, but you don't have to study it like 1Q

- **Ethernet Header (From Day 5)**

Does you remember Ethernet Header

  - 802.1Q Tag Placement
  - The 802.1Q tag is inserted between two fields of the Ethernet header

- The 802.1Q tag is inserted between the source MAC address
- And the type or length field

802.1Q Tag Basics

- The tag is four bytes or 32 bits in length
- The tag consists of two main fields
    - Tag Protocol Identifier (TPID)
    - Tag Control Information (TCI)
- The TCI itself consists of three sub fields

802.1Q Tag Format

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

TPID Field

- The field is 16 bits or two bytes in length
- Taking up half of the 802.1Q tag length
- The TPID is always set to a value of 0x8100
- This value indicates that the frame is 802.1Q tagged

TCI Field

- **PCP**
    - PCP stands for Priority Code Point
    - The field is three bits in length
    - Used for class of service (COS)

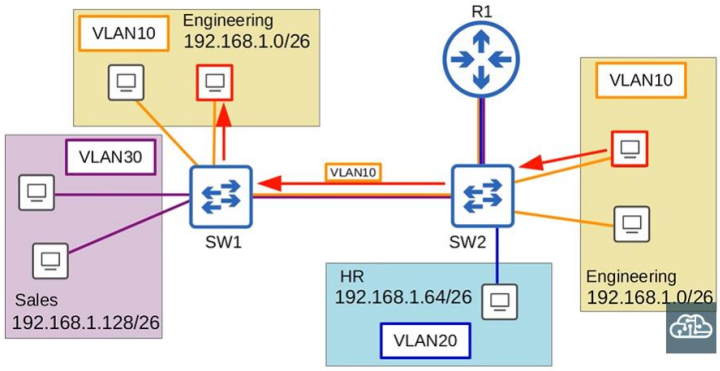
- **DEI Field**
  - DEI stands for Drop Eligible Indicator
  - The field is one bit in length
  - Used to indicate frames that can be dropped during congestion
- **VID Field**
  - VID stands for VLAN ID
  - The field is 12 bits in length
  - It identifies the VLAN the frame belongs to
  - Total VLANs = 4096
  - VLAN 0 and VLAN 4095 are reserved
  - Usable VLAN range = 1 to 4094

🔹 VLAN Ranges

- Normal VLANs: 1 to 1005
- Extended VLANs: 1006 to 4094
- Some Old Switch not support extended VLANs.

🔹 VLAN Traffic on Trunk Port

PC (VLAN 10)→SWITCH 2 ==== (TAG: VLAN 10) ==== SWITCH 1→PC (VLAN 10)

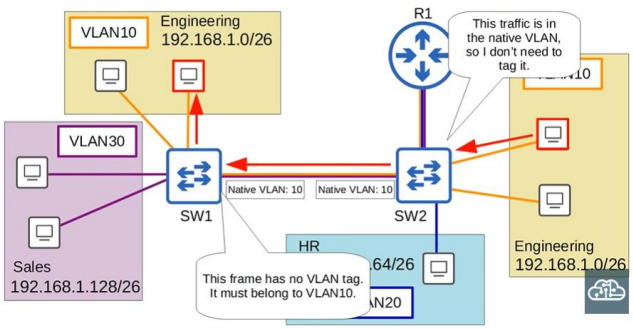


- Switch forwards traffic only within the same VLAN
- Layer 2 switches do not forward traffic between VLANs

Native VLAN

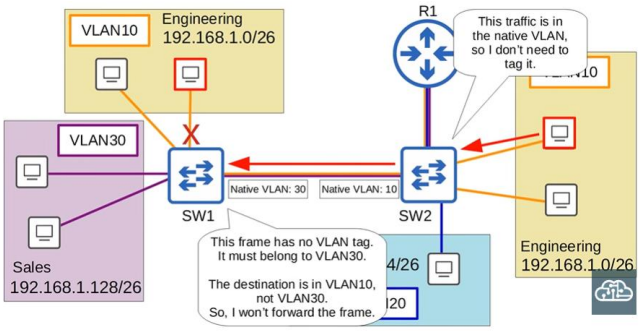
- 802.1Q has a feature called the native VLAN
- ISL does not have this feature
- Native VLAN is VLAN 1 by default
- Can be manually configured per trunk port
- The switch does not add an 802.1Q tag to frames in the native VLAN.
- When a switch receives an untagged frame on a trunk port, it assumes the frame belongs to the native VLAN, It's very important that the native VLAN matches!

Native VLAN (Untagged Frame)



- Frames in the native VLAN are not tagged
- Receiving switch assumes untagged frames belong to the native VLAN

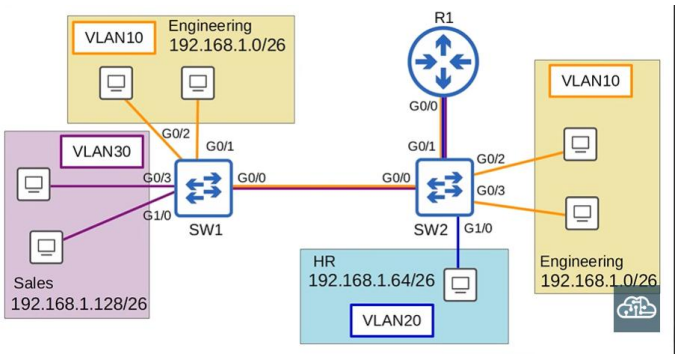
Native VLAN Mismatch Example



- Frame has no VLAN tag
- Switch assumes it belongs to VLAN 30
- Destination is in VLAN 10 & Frame is not forwarded.

Trunk Port Configuration (Deep Understanding)

🔹 Network Diagram (With Interface Numbers)



- G0/0 between SW1 and SW2 will be a **trunk**
- G0/0 and G0/1 on SW2 will also be **trunks**
- Trunks are required because **multiple VLANs must pass**

🔹 What We Are Doing Here

- We are configuring **trunk ports**
- A trunk port:
    - Carries traffic of **multiple VLANs**
    - Uses **802.1Q tagging**
- Access ports carry **only one VLAN**
- Trunks are needed between:
    - Switch ↔ Switch
    - Switch ↔ Router

- **Manual Trunk Configuration (Switch 1)**

interface g0/0

```
SW1(config)#interface g0/0
SW1(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation is "Auto" can not be configured to "trunk" mode.
SW1(config-if)#switchport trunk encapsulation ?
  dot1q      Interface uses only 802.1q trunking encapsulation when trunking
  isl        Interface uses only ISL trunking encapsulation when trunking
  negotiate  Device will negotiate trunking encapsulation with peer on
             interface
```

“switchport mode trunk”

This command tells the switch: this interface should act as a trunk

Command rejected!

An interface whose trunk encapsulation is "Auto", cannot be configured to "trunk" mode.

Reason:

- The switch does not yet know **which trunking protocol** to use
- Encapsulation is still set to **auto**
- Auto means: switch is waiting to negotiate ISL or 802.1Q

- **Important Concept (Very Important)**
  - Some switches support:
    - ISL + 802.1Q
  - Some switches support:
    - Only 802.1Q
  - On switches that support both:
    - You MUST manually set encapsulation first
  - On switches that support only 802.1Q:
    - This step is skipped

- **Setting Trunk Encapsulation**

```
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#
```

interface g0/0

switchport trunk encapsulation dot1q→ But in cisco it automatically

switchport mode trunk

- Now the switch knows:
    - Use **802.1Q** for tagging
- After this:
    - Trunk mode is accepted
- 802.1Q is:
    - Industry standard
    - Used in real networks

🔹 Verify Trunk Status

“show interfaces trunk”

- Mode **on** → manually configured
- Status **trunking** → trunk is active
- Native VLAN **1** → default native VLAN
- VLANS Gi0/1 Allowed ALL BY Default (1-4094).
- This is not secure, not efficient.

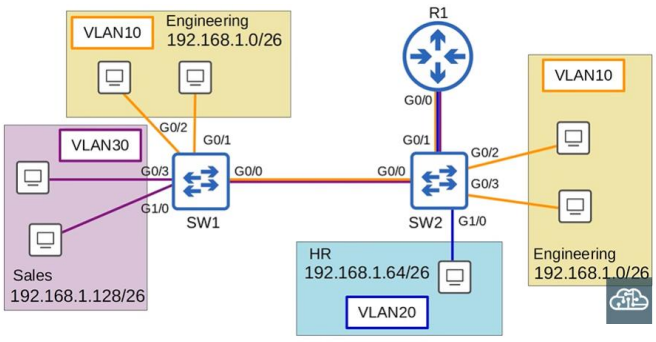
More Configuration

```
SW1(config)#int g0/0
SW1(config-if)#
SW1(config-if)#switchport trunk allowed vlan ?
  WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list

SW1(config-if)#switchport trunk allowed vlan
```

- WORD (VLAN IDs)
  → Specify which VLANs are allowed
  Example: 10, 10,20, 10-30
- add
  → Add VLANs to the existing allowed list
  Example: add 30
- all
  → Allow all VLANs on the trunk (default)
- except
  → Allow all VLANs except the specified ones
  Example: except 1
- none
  → Remove all VLANs from the trunk
  (No traffic will pass)
- remove
  → Remove specific VLANs from allowed list
  Example: remove 20

Correct VLANs for This Network



“switchport trunk allowed vlan 10,30”

```
SW1(config-if)#switchport trunk allowed vlan 10,30
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW1(config-if)#
```

Changing the Native VLAN: “switchport trunk native vlan 1001”

```
SW1(config-if)#switchport trunk native vlan 1001
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001
```

- Native VLAN: Frames sent **without 802.1Q tag**
- Best practice: Change native VLAN to an unused VLAN
- Reason: Security, Avoid VLAN hopping attacks
- Native VLAN must **match on both sides**

🔹 **Why Trunk Ports Don’t Show in VLAN Output**

“show vlan brief”

```
SW1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi1/1, Gi1/2, Gi1/3, Gi2/0
                                                Gi2/1, Gi2/2, Gi2/3, Gi3/0
                                                Gi3/1, Gi3/2, Gi3/3
10   ENGINEERING                      active    Gi0/1, Gi0/2
30   SALES                            active    Gi0/3, Gi1/0
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW1#
```

- This command shows: access ports only
- Trunk ports: carry multiple VLANs, so they are not listed here
- Use: “show interfaces trunk”

- **Switch 2 Configuration**
  - Interface G0/0 (to SW1)

```
SW2(config)#interface g0/0
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW2(config-if)#
```

  - Interface G0/1 (to Router R1)

```
SW2(config)#interface g0/1
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,20,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001
Gi0/1       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
Gi0/1       none
SW2(config-if)#
```

Router on a Stick (ROAS)

🔹 **Why Do We Need Router on a Stick?**

- In the previous lecture, **three separate router interfaces** were used
- Each interface connected to one VLAN
- Each interface had its own IP address
- Each IP acted as the **default gateway** for that VLAN
- Now, there is **only ONE physical link** between:
    - Switch 2
    - Router R1

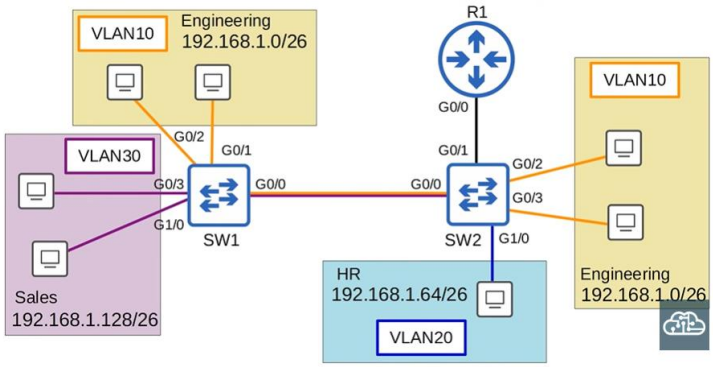
➡️ So, we **cannot** use one physical interface per VLAN
➡️ We must use **subinterfaces**

🔹 **What is Router on a Stick?**

- Router on a Stick is also written as **ROAS**
- It is a method of **inter-VLAN routing**
- It uses:
    - **One physical interface** on the router
    - **Multiple subinterfaces**
- The physical interface looks like a **stick** in the topology

🔹 Physical Connection Used

R1 G0/0 <==================> SW2 G0/1 (TRUNK LINK)



- Only **one physical cable**
- VLAN traffic is separated using **802.1Q tags**

🔹 Dividing One Interface into Subinterfaces

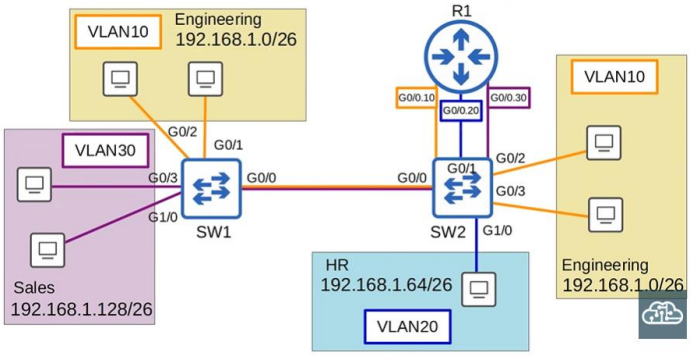
Physical Interface: G0/0

Subinterfaces:

G0/0.10 --> VLAN 10

G0/0.20 --> VLAN 20

G0/0.30 --> VLAN 30



- These are **logical interfaces**
- They behave like **separate interfaces**
- Recommended:
    - Subinterface number = VLAN number
    - Makes configuration easy to understand

🔹 **Important Switch Configuration Note**

- No extra configuration is needed on SW2
- Interface **G0/1 on SW2**:
    - Already configured as a **trunk**
    - VLANs **10, 20, 30 allowed**

➡️ Now we configure **ONLY the router**

🔹 **Router Configuration – Step by Step**

```
R1(config)#interface g0/0
R1(config-if)#no shutdown
R1(config-if)#
*Apr 15 04:29:49.681: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Apr 15 04:29:50.682: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/
0, changed state to up
R1(config-if)#interface g0/0.10
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.1.62 255.255.255.192
R1(config-subif)#interface g0/0.20
R1(config-subif)#encapsulation dot1q 20
R1(config-subif)#ip address 192.168.1.126 255.255.255.192
R1(config-subif)#interface g0/0.30
R1(config-subif)#encapsulation dot1q 30
R1(config-subif)#ip address 192.168.1.190 255.255.255.192
R1(config-subif)#
```

Step 1: Enable Physical Interface

- R1(config)# interface g0/0
- R1(config-if)# no shutdown
- Router interfaces are shutdown by default
- Must be enabled first

Step 2: Subinterface for VLAN 10

- R1(config)# interface g0/0.10
- R1(config-subif)# encapsulation dot1q 10
- R1(config-subif)# ip address 192.168.1.62 255.255.255.192
- encapsulation dot1q 10
    - Tells router: VLAN 10 traffic belongs here
- IP address:
    - Acts as **default gateway** for VLAN 10

Step 3: Subinterface for VLAN 20

- R1(config)# interface g0/0.20
- R1(config-subif)# encapsulation dot1q 20
- R1(config-subif)# ip address 192.168.1.126 255.255.255.192

Step 4: Subinterface for VLAN 30

- R1(config)# interface g0/0.30
- R1(config-subif)# encapsulation dot1q 30
- R1(config-subif)# ip address 192.168.1.190 255.255.255.192

🔹 Verify Subinterfaces

"show ip int brief"

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES NVRAM  up                    up
GigabitEthernet0/0.10      192.168.1.62    YES manual up                    up
GigabitEthernet0/0.20      192.168.1.126   YES manual up                    up
GigabitEthernet0/0.30      192.168.1.190   YES manual up                    up
GigabitEthernet0/1         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/3         unassigned      YES NVRAM  administratively down down
```

- Physical interface has **no IP**
- All routing happens on **subinterfaces**

🔹 Routing Table Verification

"show ip route"

```
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 6 subnets, 2 masks
C        192.168.1.0/26 is directly connected, GigabitEthernet0/0.10
L        192.168.1.62/32 is directly connected, GigabitEthernet0/0.10
C        192.168.1.64/26 is directly connected, GigabitEthernet0/0.20
L        192.168.1.126/32 is directly connected, GigabitEthernet0/0.20
C        192.168.1.128/26 is directly connected, GigabitEthernet0/0.30
L        192.168.1.190/32 is directly connected, GigabitEthernet0/0.30
R1#
```

- Connected and local routes are added automatically
- Same behavior as physical interfaces

- **How Inter-VLAN Routing Works (Example)**

VLAN 10 → VLAN 30 Communication