DAY 11 - Routing Fundamental

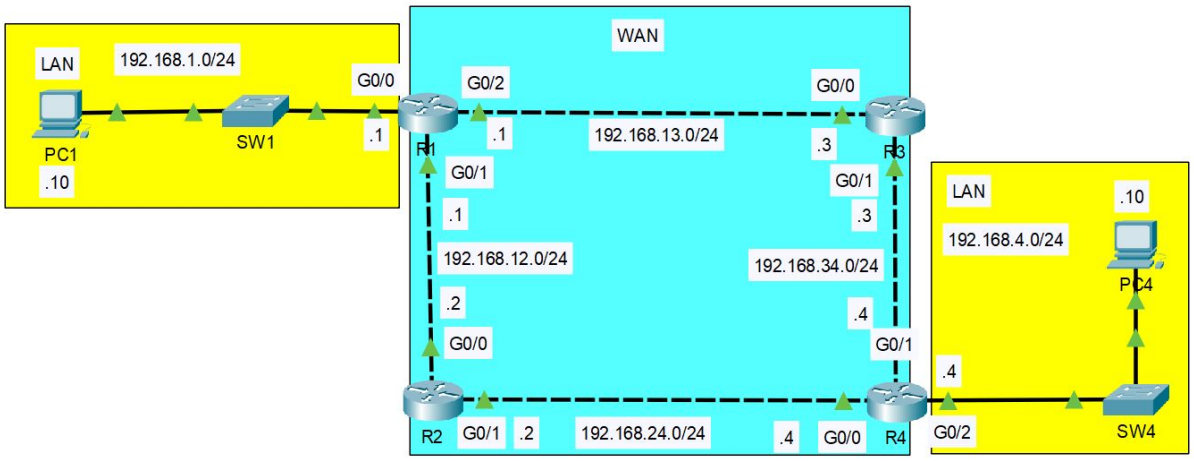
Purinat33

Routing Fundamental

What is Routing

1. Routing is the process that routers use to determine the path that IP packets should take over a network to reach their destination.
   (a) Routers store routes to all of their known destinations in a **Routing Table**.
   (b) When routers receive packets, they look in their **Routing Table** to find the best route to forward that packet.
2. There are *two* main routing methods (methods that routers use to learn routes):
   (a) **Dynamic Routing**: Routers use Dynamic Routing Protocols (eg. `OSPF`) to share routing information with each other automatically and build their routing tables.
   (b) **Static Routing**: A network engineer/admin manually configures routes on the router.
3. A route tells the router: *To send a packet to* `Destination X` *, it should send the packet to* `Next Hop Y` .
   (a) Or, if the destination is directly connected to the router, send the packet directly to the **Destination**.
   (b) Or, if the destination is the router's own IP address, then **Receive the Packet for Yourself** (and don't forward).

Example Topology:



Pre-routing setup:

- Example:
  - `R1> en`
  - `R1# conf t`
  - `R1(config)# interface g0/0`
  - `R1(config-if)# ip address 192.168.1.1 255.255.255.0`
  - `R1(config-if)# no shutdown`
- Repeat for all PC, Routers & Interfaces.
  - **R1**:

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.1.1     YES NVRAM  up                    up
GigabitEthernet0/1     192.168.12.1    YES NVRAM  up                    up
GigabitEthernet0/2     192.168.13.1    YES NVRAM  up                    up
```

  - **R2**:

```
R2#show ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.12.2    YES NVRAM  up                    up
GigabitEthernet0/1     192.168.24.2    YES NVRAM  up                    up
GigabitEthernet0/2     unassigned      YES NVRAM  administratively down down
```

  - **R3**:

```
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.13.3    YES manual up                    up
GigabitEthernet0/1     192.168.34.3    YES manual up                    up
GigabitEthernet0/2     unassigned      YES unset  administratively down down
```

  - **R4**:

```
R4#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.24.4    YES manual up                    up
GigabitEthernet0/1     192.168.34.4    YES manual up                    up
GigabitEthernet0/2     192.168.4.4     YES manual up                    up
```

View Routing Table (R1) using `show ip route` :

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/24 is directly connected, GigabitEthernet0/1
L       192.168.12.1/32 is directly connected, GigabitEthernet0/1
     192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.13.0/24 is directly connected, GigabitEthernet0/2
L       192.168.13.1/32 is directly connected, GigabitEthernet0/2
```

Dissecting routing table (R1):

1. **Codes**:

   ```
   Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter are
          * - candidate default, U - per-user static route, o - ODR
          P - periodic downloaded static route
   ```

   The **Codes** section is the legend for the table which lists the different protocols which routers can use to learn routes.

2. **Routes**:

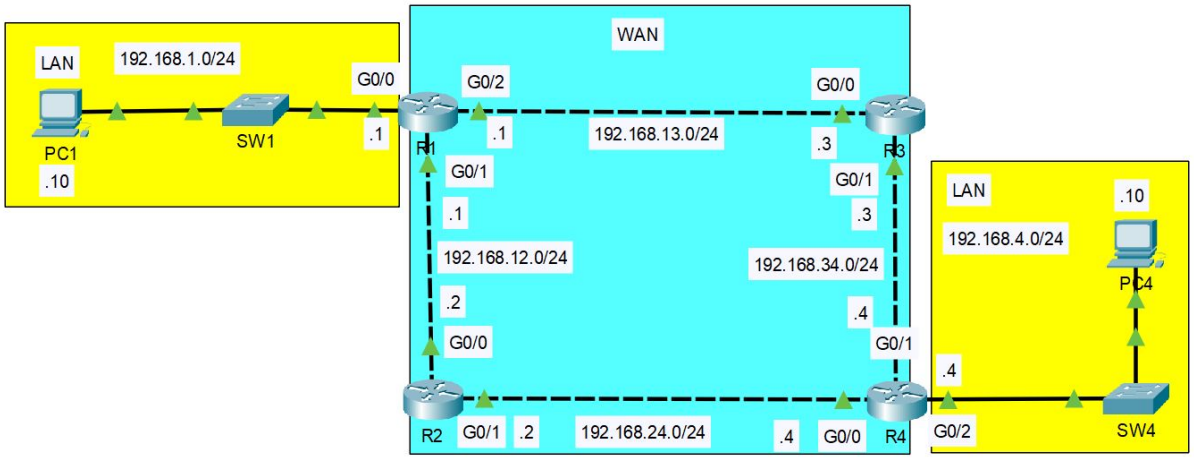
   We automatically get **2 Routes** without even configuring anything.

   - `L` : **Local** route, a route to the **Actual IP Address configured on the interface** (With a `/32` netmask).

     ```
     R1#show ip int brief
     Interface              IP-Address
     GigabitEthernet0/0     192.168.1.1
     GigabitEthernet0/1     192.168.12.1
     GigabitEthernet0/2     192.168.13.1
     ```

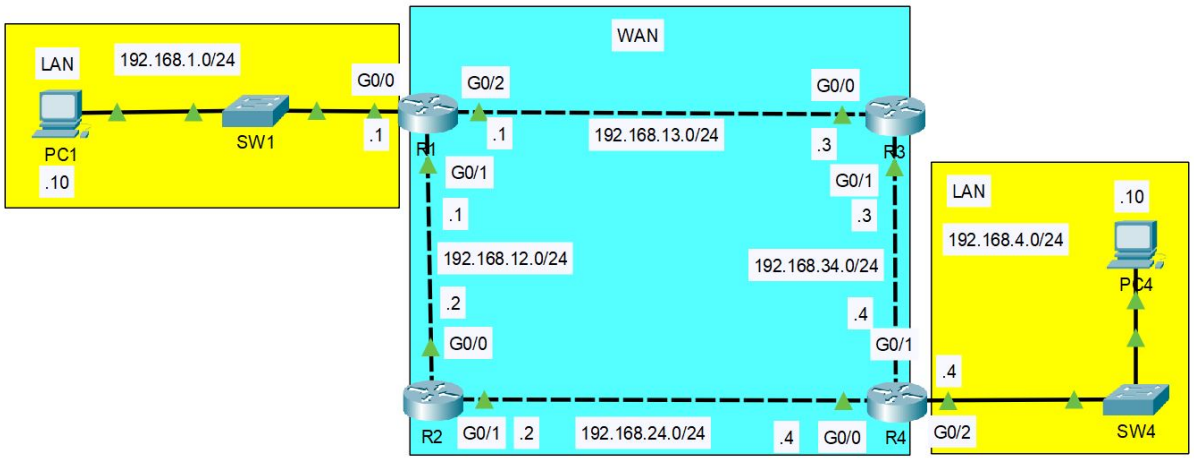
     ```
     L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
          192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
     C       192.168.12.0/24 is directly connected, GigabitEthernet0/1
     L       192.168.12.1/32 is directly connected, GigabitEthernet0/1
          192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
     C       192.168.13.0/24 is directly connected, GigabitEthernet0/2
     L       192.168.13.1/32 is directly connected, GigabitEthernet0/2
     ```

- `C` : **Connected** route, a route to the **Network ID the interface is connected to** (with the actual netmask configured on the interface.)




```
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/24 is directly connected, GigabitEthernet0/1
L       192.168.12.1/32 is directly connected, GigabitEthernet0/1
     192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.13.0/24 is directly connected, GigabitEthernet0/2
L       192.168.13.1/32 is directly connected, GigabitEthernet0/2
```

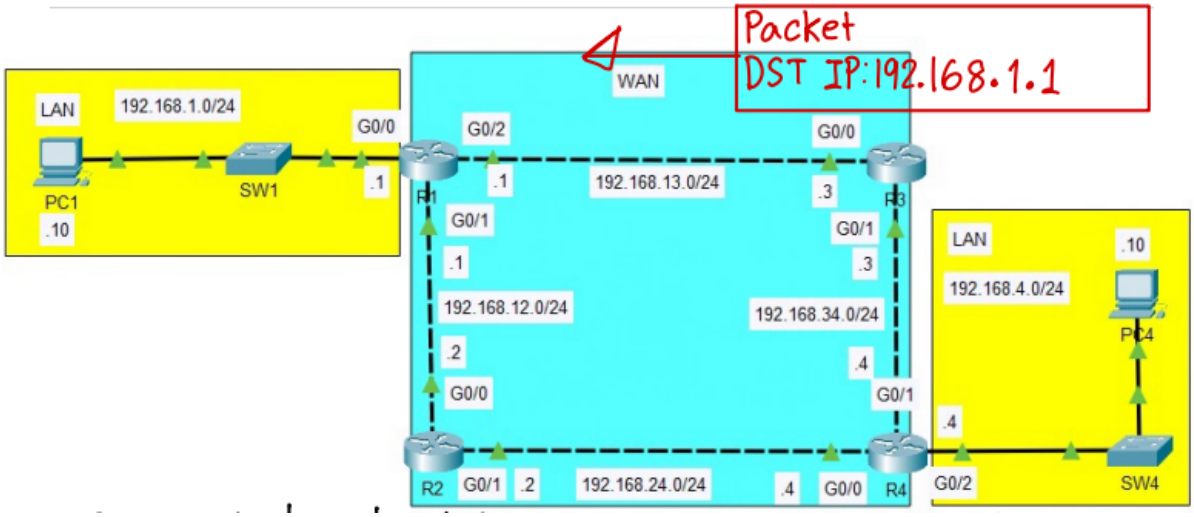
Connected vs. Local Routes:



- A **Connected Route** is a route to the network the *interface* is connected to:
  - **G0/2** of **R1**: 192.168.1. `1` /24
  - **Network Address**: 192.168.1. `0` /24
  - It provides a route to **all hosts in that network**:
    - 192.168.1. `3`
    - 192.168.1. `135`
    - 192.168.1. `254` etc.

  - **R1** knows:
    - *"If I need to send a packet to any host in the* `192.168.1.0/24` *network, I should send it out using the* ***G0/2 Interface****"*
      - 192.168.1. `87` = Within that network, send packet out of **G0/2**
      - 192.168. `2` .1 = **Not Within the same Network**, Send out via a different interface or drop the packet.
- A **Local Route** is a route to the *exact* IP address configured on the interface.
  - A `/32` netmask is used to specify the exact IP address of the interface.
  - **R1** knows:
    - *"If I receive a packet destined to this IP address, the message* ***is for me****"*

Route Selection:




```
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
```

Imagine a packet with **Destination**: `192.168.1.1` coming from **G0/0 R3**.

- The packet is matched by two routes:
  - `C` route: `192.168.0/24` for the network IP.
  - `L` route: `192.168.1.1/32` for the R1's G0/0 IP.
- Which route will R1 use?
  - It will choose the ***MOST SPECIFIC*** **Matching Route**.
    - The route to `192.168.1.0/24` includes **256** different IP addresses (`192.168.1.1` to `192.168.1.255`)
    - The route to `192.168.1.1/32` includes **1** IP address (only `192.168.1.1`)

- We can see that `192.168.1.1/32` route is the most specific.
- Thus, the route **R1** uses is `L` `192.168.1.1/32` .

Misc.

```
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/1
L        192.168.12.1/32 is directly connected, GigabitEthernet0/1
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/2
L        192.168.13.1/32 is directly connected, GigabitEthernet0/2
```

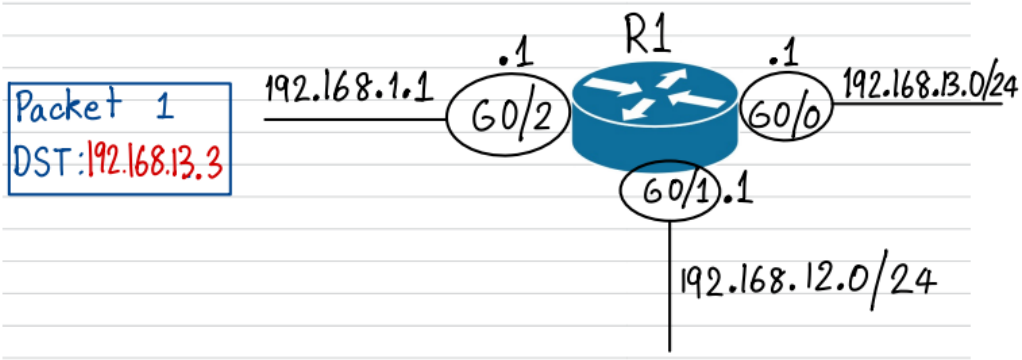
These three lines are not routes, they mean:

1. `192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks`
   (a) In the routing table, there are **Two Routes** to subnets that fit within the `192.168.1.0/24` class C network, with **two different netmasks** (`24` and `32`)
2. `192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks`
   (a) In the routing table, there are **Two Routes** to subnets that fit within the `192.168.12.0/24` class C network, with **two different netmasks** (`24` and `32`)
3. `192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks`
   (a) In the routing table, there are **Two Routes** to subnets that fit within the `192.168.13.0/24` class C network, with **two different netmasks** (`24` and `32`)

We will cover **Subnetting** later, for now just know that these three lines are not routes.

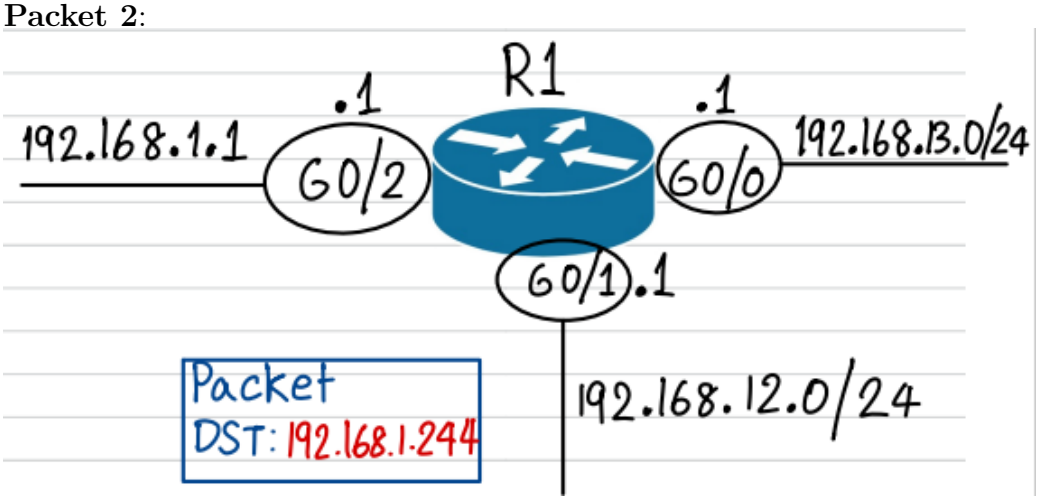
Route Selection (Practice):

1. **Packet 1**:



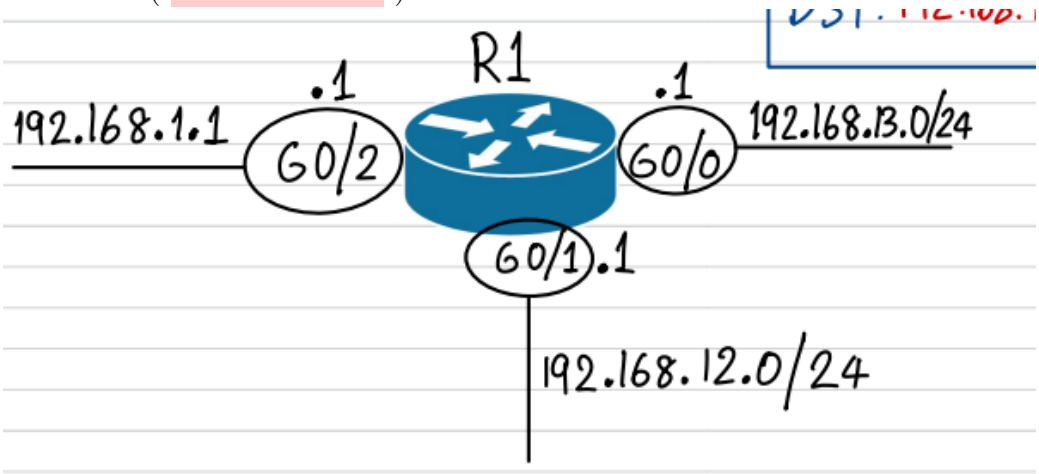
   - Most matching route: `192.168.13.0/24` via **G0/0** interface.

2. **Packet 2**:



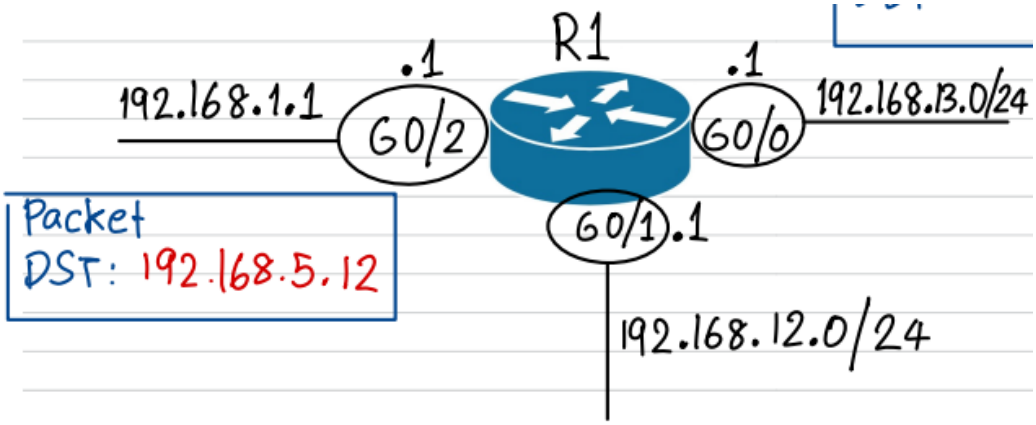
   - Most matching route: `192.168.1.0/24` via **G0/2** interface.

3. **Packet 3** (`192.168.12.1`):



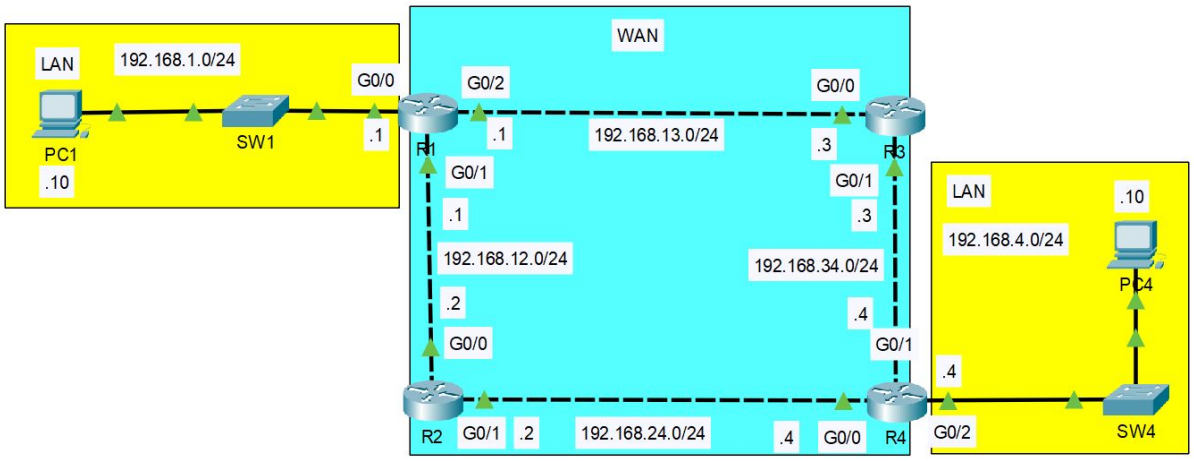
   - Most matching route: `192.168.12.1/32` At the **G0/1** interface.

4. **Packet 4**:



- Most matching route: **None**, R1 will drop the packet.

Summary



- Routers store information about destinations they know in their *Routing Table*.
  - When they receive packets, they look in the routing table to find the best route to forward the packet.
- Each **Route** in the routing table is an instruction:
  - To reach destinations in `Network X`, send the packet to `Next-Hop Y` (The next router in the path to the destination).
    - If the destination is directly connected (**Connected Route**), then send the packet directly to the destination.
    - If the destination is your own IP Address (**Local Route**), then receive the packet for yourself.
- When you configured an IP address on an interface and enable the interface, **Two Routes** are automatically added to the routing table:
  - **Connected Route** (Code `C` in the routing table): A route to the network connected to the interface.

        * If the interface's IP is `192.168.1.1/24`, the `C` route will be to `192.168.1.0/24`
        * Tells the router: "To send a packet to a destination in this network, send it out of the interface specified in the route."
    - **Local Route** (Code `L` in the routing table): A route to the **exact** IP address configured on the interface.
        * If the interface's IP is `192.168.1.1/24`, the `L` route will be to `192.168.1.1/32`
        * Tells the router: "Packets to this destination are for you, you should receive them for yourself only".
- A route **Matches** a destination if the packet's destination IP address is part of the network specified in the route.
    - A packet to `192.168.1.70` is matched by a route to `192.168.1.0/24` **but not** `192.168.0.0/24`
- If a router receives a packet and it doesn't have a route that matches the packet's destination, it will **Drop** the packet.
    - This is different from **Switches**, which will **Flood** the frames if they don't have a **MAC Entry** for the Frame's destination.
- If a router receives a packet and it has multiple routes that match the packet's destination, it will use the **Most Specific** matching route to forward the packet.
    - **Most Specific Matching Route** = The matching route with the **longest** prefix length.
        * `192.168.1.1` : `192.168.1.1/32` > `192.168.1.0/24`
    - This is different from **Switches**, which look for an **Exact Match**.
