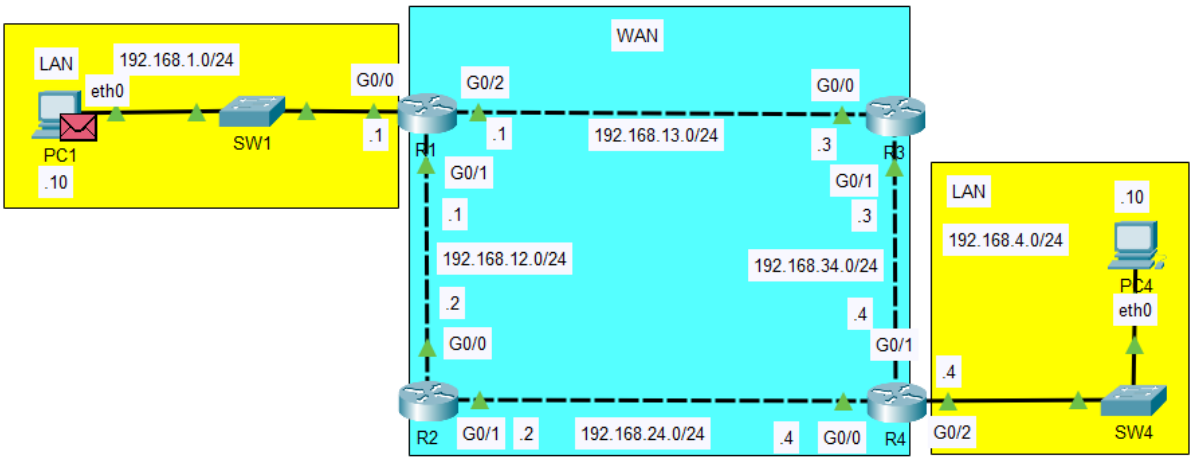
DAY 12 - Packet Life Cycle

Purinat33

The Life of a Packet

This is already covered in previous days, but this is more of a summary into one big picture.

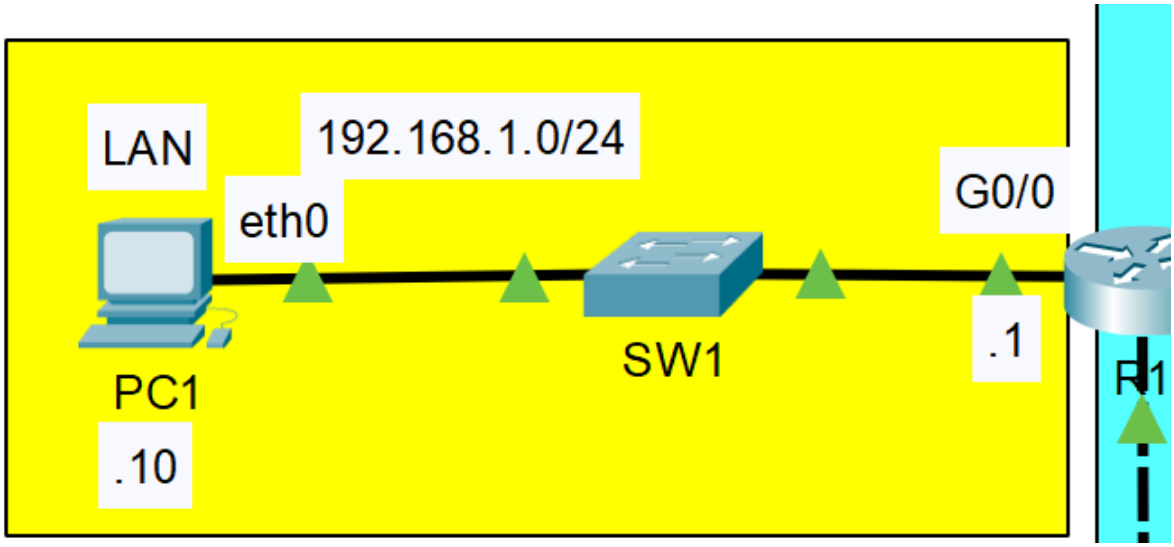
Network Topology



- Looking at the packets going from `192.168.1.10/24` to `192.168.4.10/24`
- All **Static Routes** have been configured to allowed `192.168.1.0/24` network to communication with `192.168.4.0/24`
- We're gonna use some **MAC Addresses** as well in addition to existing **IP**s.

Device	PC1	R1	R1	R2	R2	R4	R4	PC4
Interface	-	G0/0	G0/1	G0/0	G0/1	G0/0	G0/2	-
MAC	1111	aaaa	bbbb	cccc	dddd	eeee	ffff	4444

Life Cycle

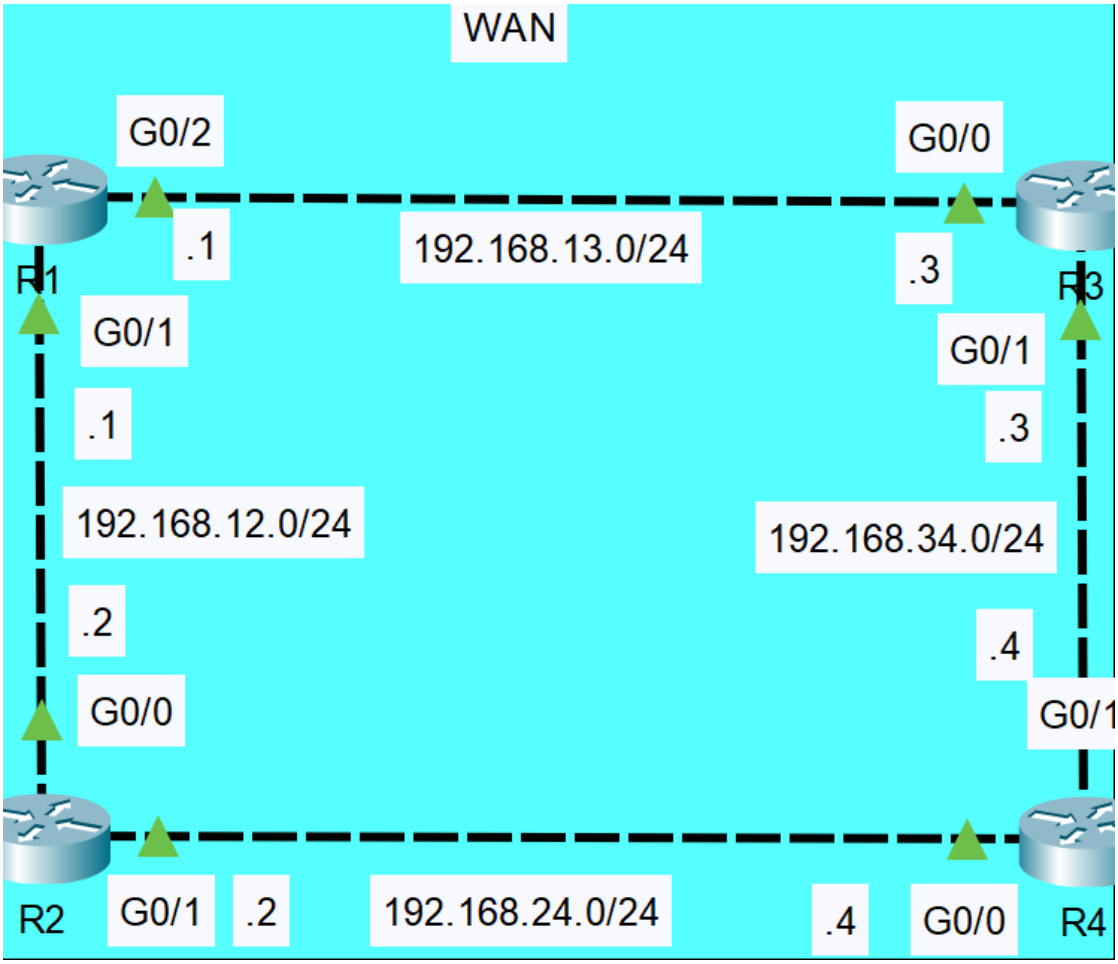


To R1:

1. **PC1** creates a packet with this IP header to **PC4**:
   (a) **SRC**: `192.168.1` .10
   (b) **DST**: `192.168.4` .10
2. **PC1** sees that the **Destination** is in a different network.
   (a) It sends the packet to its **Default Gateway** (**R1**)
3. **PC1** does not know the **MAC Address** of **R1**.
   (a) It will use **ARP** (**Address Resolution Protocol**)
4. **PC1** creates an **ARP Request Packet**:
   (a) **SRC IP**: `192.168.1.10`
   (b) **DST IP**: `192.168.1.1`
   (c) **DST MAC**: *`FFFF.FFFF.FFFF`* (**Broadcast MAC**)
   (d) **SRC MAC**: `xxxx.xxxx.1111` (Kept last 4 digits for simplicity)
5. **SW1** floods the frame on all port except the source port. (To R1's G0/0)
6. **R1's g0/0** matches the request frame with its own IP, and creates an **ARP Reply**:
   (a) **SRC IP**: `192.168.1.1`
   (b) **DST IP**: `192.168.1.10`
   (c) **DST MAC**: *`1111`* (**Unicast MAC**)
   (d) **SRC MAC**: `aaaa`
7. **PC1** now *encapsulate* the original packet with the Ethernet header to **R1's**:
   (a) **IPv4** Header:
      i. **SRC**: `192.168.1` .10 (**PC1**)
      ii. **DST**: `192.168.4` .10 (**PC4**)
   (b) **Ethernet** Header:
      i. **DST**: `aaaa` (**PC1**)

ii. **SRC**: `1111` (**R1's g0/0**)

From R1 to R2:



1. **R1** receives the frame from PC1 and *remove* the **Ethernet Header** (The *IPv4 Header* stays the same).
2. **R1** looks up its routing table for `192.168.4.10` :
   (a) The most specific match is for **Destination:** `192.168.4.0/24` with the **Next Hop:** `192.168.12.2` :

Routing Table for R1

Type	Network	Port	Next Hop IP	Metric
S	192.168.4.0/24	---	192.168.12.2	1/0

3. **R1** will now encapsulate the packet with the ethernet header for `192.168.12.2` 's **MAC Address**:
   (a) **R1** performs **ARP** for **R2's** MAC Address.
      i. **SRC IP**: `192.168.12.1` (R1's g0/1)

ii. **DST IP**: `192.168.12.2` (R2's g0/0)
iii. **DST MAC**: *`FFFF.FFFF.FFFF`*
iv. **SRC MAC**: `bbbb`

4. **R2's g0/0** receives the **ARP Request**, it replies with its own **MAC Address** to **R1's g0/1**:
   (a) **SRC IP**: `192.168.12.1` (R1's g0/1)
   (b) **DST IP**: `192.168.12.2` (R2's g0/0)
   (c) **DST MAC**: `bbbb`
   (d) **SRC MAC**: `cccc`
5. **R1** encapsulates the still-the-same IPv4 header with the new Ethernet Header, with **R2's g0/0** MAC Address being the destination MAC Address.

The same pattern is used for **R2 to R4** and **R4 to PC4** with the **ARP Request/Reply** for the MAC address of the next hop, **De-Encapsulation** of the Ethernet Header at each hop while the IPv4 Packet stays the same the entire time.

LAB: