

Computer Network I

Reti di Calcolatori I

Università di Napoli Federico II – Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Informatica

Riccardo Caccavale
(riccardo.caccavale@unina.it)



Network Layer

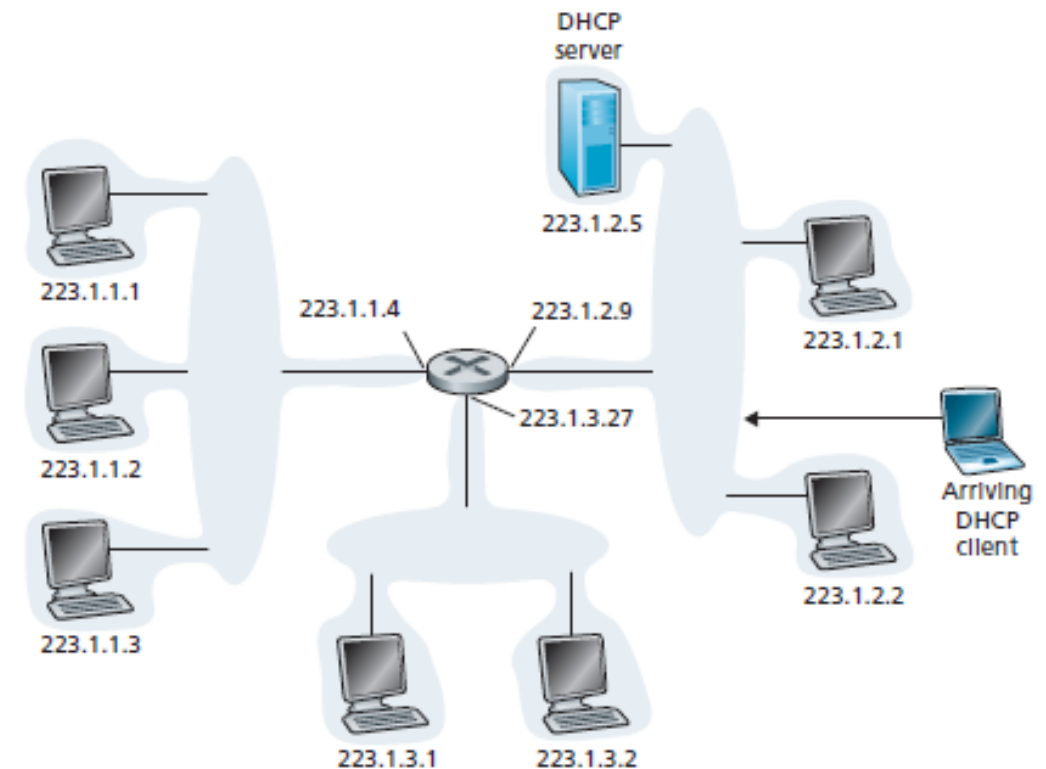
Internet Protocol: IP Assignment

- Within a block of addresses, **IP addresses must be assigned** to individual interfaces.
- A **system administrator** can configure the IP addresses in two ways:
 - **Manually**: by assigning one-to-one IP addresses to hosts.
 - **Automatically**: the network autonomously assigns free IP to incoming hosts.
- The second approach (most common) is done by using the **Dynamic Host Configuration Protocol (DHCP)**.
- In addition to host IP address, **DHCP provides a host with the information needed to join the network**, such as subnet mask, the address of the default gateway (to go outside of the network), and the address of the local DNS server.
- The **DHCP is plug-and-play**, and it is typically used in our homes!

Network Layer

Internet Protocol: DHCP

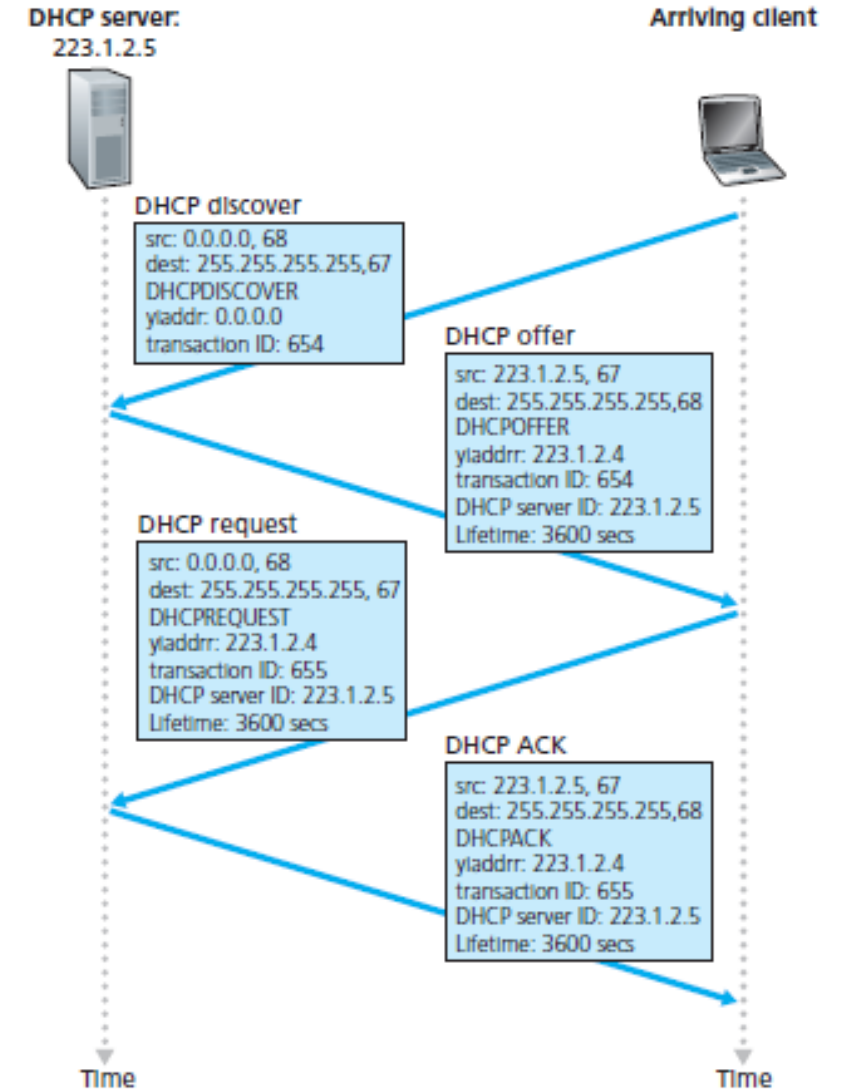
- The **DHCP is a client-server protocol**: a newly arriving host (client) connects to the DHCP server to receive network information. **DHCP service may be provided by a computer or by the router itself.**
- Let's look to the previous example of **3 subnets connected by a single router.**
- Here we assume **the network on the right (223.1.2.0/24) to be endowed with a DHCP server.**
- When a new host joins the network, **it trades the IP address with the DHCP.**



Network Layer

Internet Protocol: DHCP

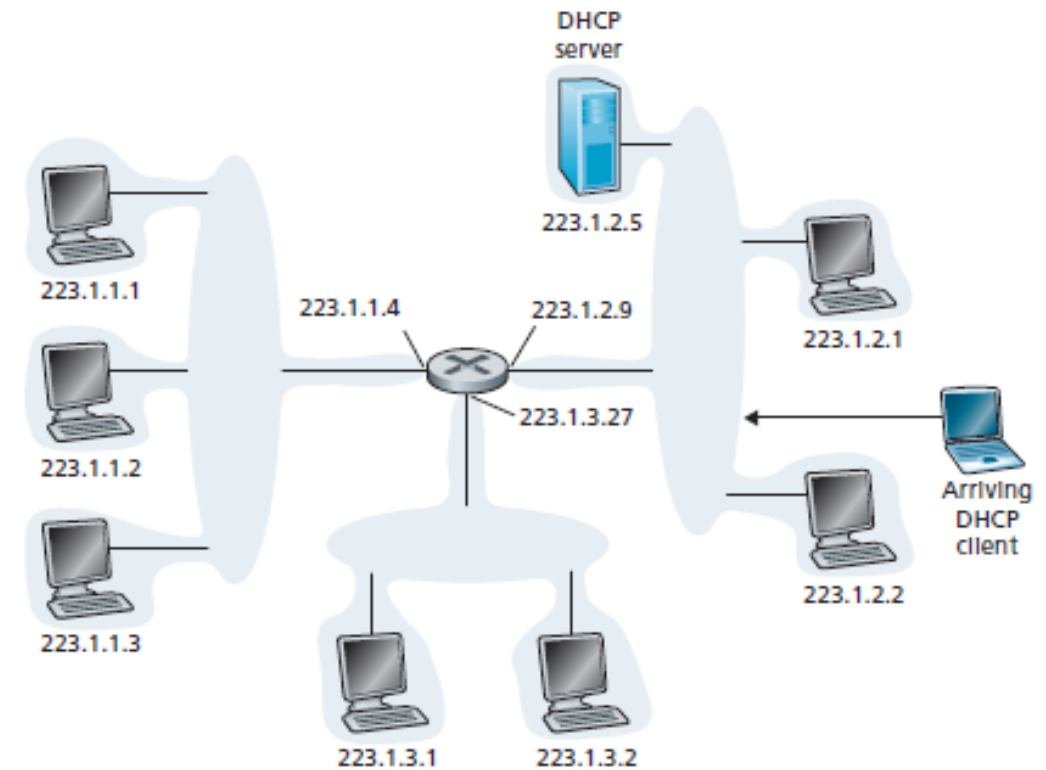
- Adding a new host is a 4-steps process:
 1. **DHCP server discovery:** the new host sends in **broadcast a DHCP discovery message (UDP on port 67)** to find the DHCP server.
 - Destination IP: 255.255.255.255 (broadcast).
 - Source IP: 0.0.0.0 (this host).
 2. **DHCP server offer:** since multiple DHCP servers may be present, when a discovery message is received, a **server responds with a broadcast message (on port 68) offering a possible network configuration (yiaddr filed – Your Internet ADDRESS).**
 - Destination IP: 255.255.255.255 (broadcast).
 - Source IP: 223.1.2.5 (DHCP server).
 3. **DHCP request:** the new client selects the accepted offer by **echoing back the offer message.**
 - Destination IP: 255.255.255.255 (broadcast).
 - Source IP: 223.1.2.5 (DHCP server).
 4. **DHCP ACK:** final **ACK message** confirming the transaction.



Network Layer

Internet Protocol: DHCP

- In this case everything works smoothly because the **DHCP** and the arriving host are in the same subnet.
- If a server exists but in a different subnet, we need a **DHCP relay agent** to forward DHCP messages (a router can do that).
- In this example we can configure our router to be a relay agent so that **also** hosts from subnets **223.1.1.0/24** and **223.1.3.0/24** are served by our DHCP.



Network Layer

Route

- In Linux we can use the ifconfig in combination with the route command to see the network configuration provided by the DHCP.
- Usage:
 - See our current configuration (IP address, mask):
 - `$ ifconfig`
 - See our default gateway:
 - `$ route -n`

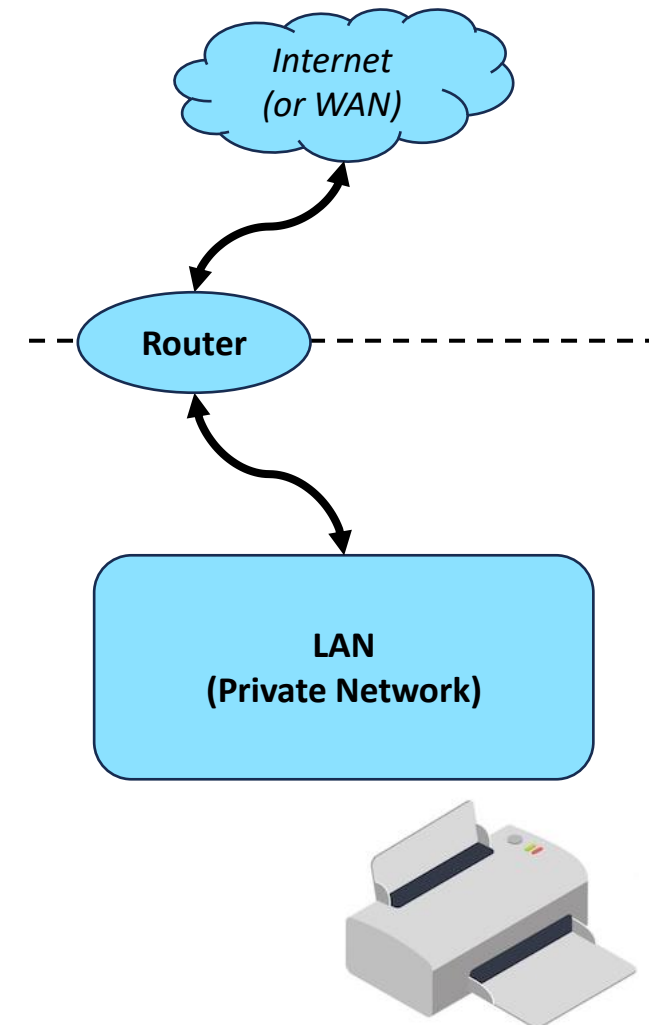
```
Lenovo-B50-80:~$ ifconfig wlp9s0
wlp9s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 100.103.0.64 netmask 255.255.0.0 broadcast 100.103.255.255
    inet6 fe80::47c4:ec38:2dc3:5d70 prefixlen 64 scopeid 0x20<link>
    ether 34:e6:ad:f0:d5:11 txqueuelen 1000 (Ethernet)
    RX packets 14398 bytes 20600768 (20.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4100 bytes 451906 (451.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Lenovo-B50-80:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 100.103.0.1 0.0.0.0 UG 600 0 0 wlp9s0
```

Network Layer

Internet Protocol: Visibility

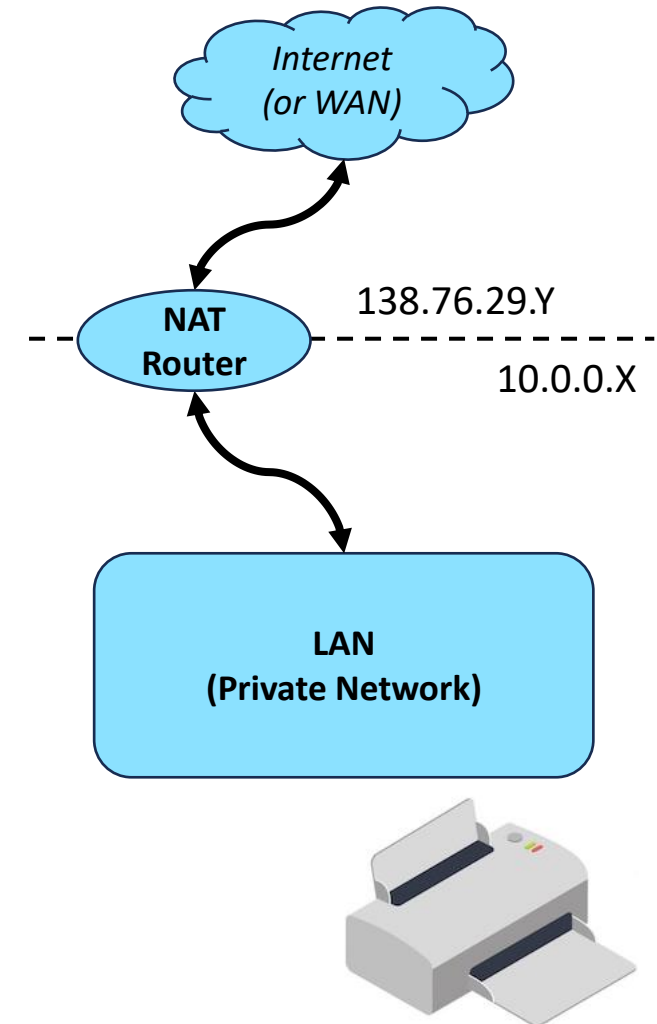
- In Internet there are billions of connected devices that must be **associated with unique IP addresses to be reached by anyone**.
- Is it really necessary **to have all devices visible to anyone**?
 - For example: is it good to have **my printer reachable (usable) by anyone** on Internet?
- There are **clear issues in having all devices reachable** from outside local networks:
 - IP addresses are finite.
 - If devices are locally connected **it is often unreasonable (or undesirable) to expose all of them on Internet**.
 - Local **administrators should be fully aware of the overlying IP block** structure in order to assign unused addresses (which is often not up to them, e.g., in home networks).



Network Layer

Internet Protocol: NAT

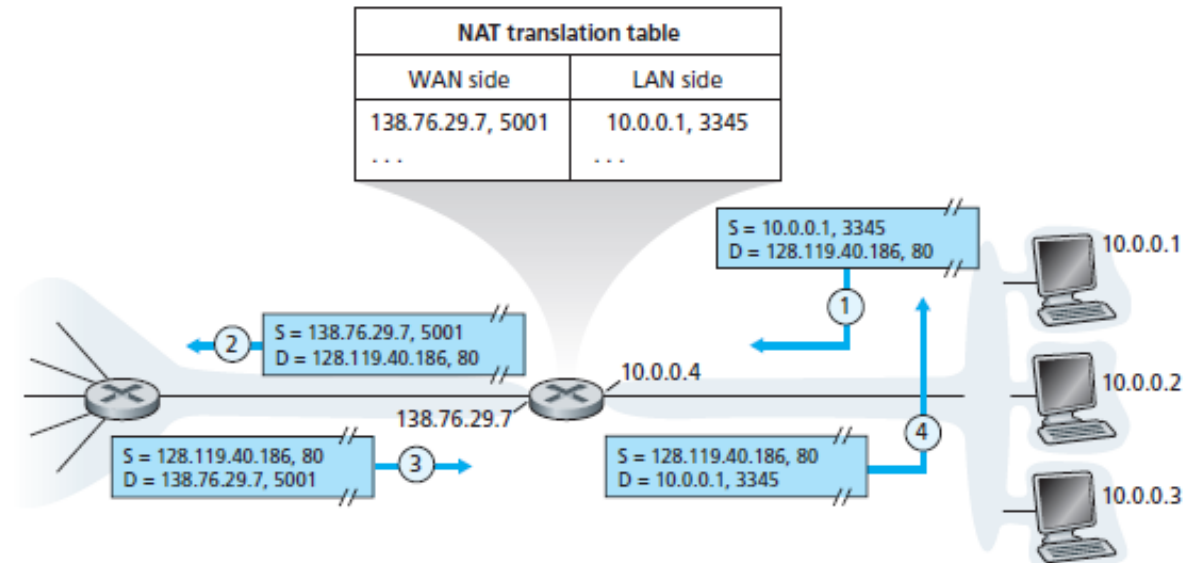
- The **Network Address Translation (NAT)** service allows to **remap the IP addresses** of packets within a network into different ones (**network masquerading**).
- This is performed by using a **translation table** associating **local IPs/ports (of the LAN)** into **global IPs/ports (of the WAN)**. This service can be offered by **routers**.
- The masqueraded local network is also called ***private network or realm with private addresses***. Private IP addresses are valid only inside the private network.



Network Layer

Internet Protocol: NAT

- This is an **example of a NAT-enabled router** connecting a network to internet.
- The four interfaces in the local network have the same subnet address of 10.0.0.0/24. The **router behaves as a message passer** that overrides the IPs with the one specified in the table:
 - **Outgoing packets** have their source IPs/ports overwritten **with a single WAN-side IP** (138.76.29.7) and a **fresh port** (5001).
 - **Incoming packets** have their destination IPs/ports overwritten **with the specific LAN-side IP** of the host (10.0.0.1) and the **initial port** (3345)
- It is important to notice that, since **hosts are unaware of the other hosts' traffic**, the **NAT cannot use the initial port** because multiple hosts may select the same port simultaneously.
- **Since the port is 16bits, NATs can manage over 60000 simultaneous connections.**



Network Layer

Ping

- To check if a host is reachable on a network, we can use the ping command (ping is the same on Linux/Windows machines).
- **Ping** (Packet Internet Groper) is a utility based on the **ICMP (Internet Control Message Protocol)** transport protocol and sends a “echo request” packet to a host that automatically answers with a “echo reply” message.
 - Ping also provide **the time elapsed between request/reply**, measuring the RTT.
- Usage:
 - To check the reachability:
 - \$ ping [target address]

```
Lenovo-B50-80:~$ ping google.com
PING google.com (142.251.209.46) 56(84) bytes of data.
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=1 ttl=117 time=19.8 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=2 ttl=117 time=20.3 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=3 ttl=117 time=19.5 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=4 ttl=117 time=18.9 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=5 ttl=117 time=18.8 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=6 ttl=117 time=29.1 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=7 ttl=117 time=68.6 ms
64 bytes from mil04s51-in-f14.1e100.net (142.251.209.46): icmp_seq=8 ttl=117 time=19.6 ms
□
```

Network Layer

Nmap (pt.2)

- Besides port scanning, the nmap command can also be used to **map/scan the devices on the network**.
- It **relies on ping (ICMP)** to discover hosts on the network.
- Usage:
 - Ping-based hosts discovery:
 - `$ sudo nmap -sn [gate address]/[subnet bits]`

```
Lenovo-B50-80:~$ sudo nmap -sn 100.103.0.1/16

Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-03 09:41 CET
Nmap scan report for _gateway (100.103.0.1)
Host is up (0.0024s latency).
MAC Address: 10:BD:18:E5:74:80 (Cisco Systems)
Nmap scan report for 100.103.0.3
Host is up (0.43s latency).
MAC Address: 6C:20:56:2C:03:2F (Cisco Systems)
Nmap scan report for 100.103.0.11
Host is up (0.0031s latency).
MAC Address: 2A:FC:45:97:87:28 (Unknown)
Nmap scan report for 100.103.0.16
Host is up (0.0046s latency).
```

Network Layer

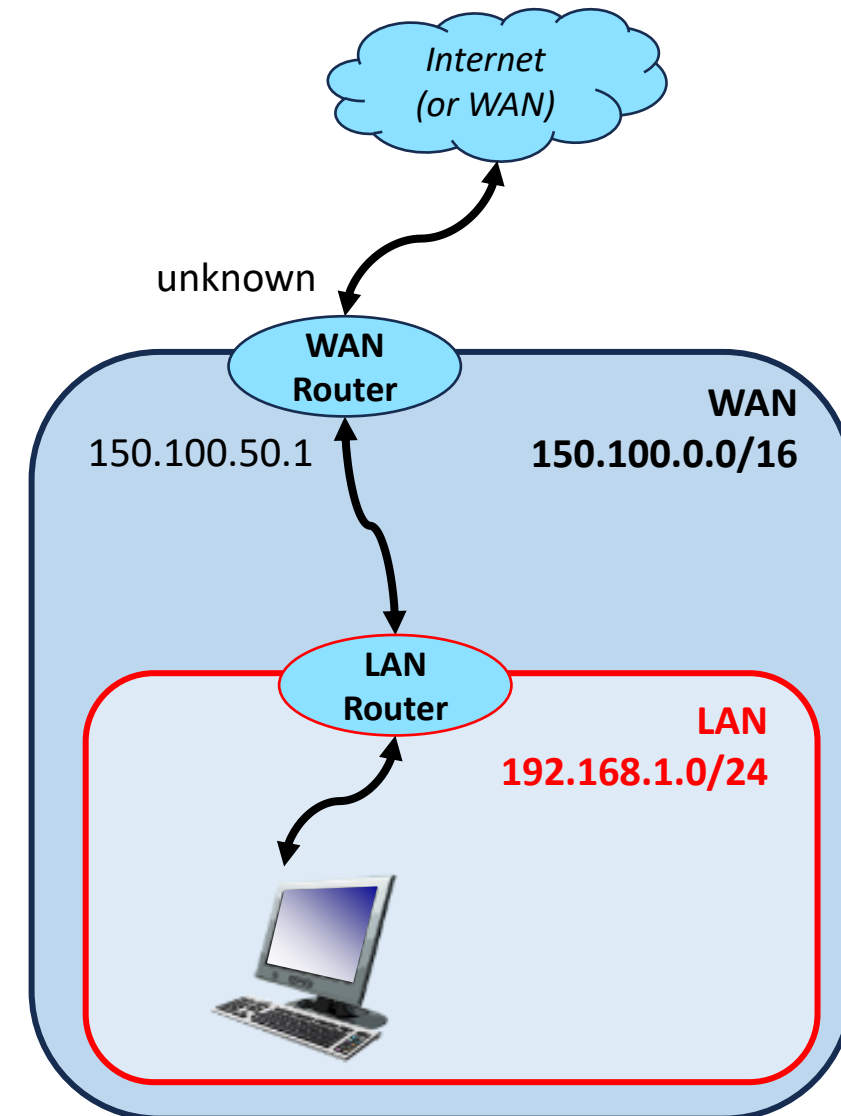
Internet Protocol: Reserved IP addresses

- Since **local administrators should be free to organize a private network** without interferences, by convention there are some **reserved IP blocks**, i.e., that ISPs/ICANN cannot assign to any organization. Some famous examples are:
 - 10.0.0.0 – 10.255.255.255 (reserved for private networks).
 - 192.168.0.0 – 192.168.255.255 (reserved for private networks).
 - 127.0.0.0 – 127.255.255.255 (indicate this machine, i.e., loopback).
 - Typically, 127.0.0.1 is used.
 - 0.0.0.0 (indicates current network).
 - Addresses having 0 in the host-part fall under this definition.
 - 255.255.255.255 (indicates broadcast).
 - Addresses having 255 in the host-part fall under this definition.
 - Etc.
- The necessity to reserve IPs is given by the fact that **private addresses may be the same as public (non-private) ones**, so routing within the network would be **ambiguous**.

Network Layer

Internet Protocol: LAN Setup Example

- Let's see an example in which we want to create a **new local subnetwork (LAN)** into a **pre-existing network (WAN)**.
- The **network administrator** of the WAN provides us the following information:
 - The **IP of the WAN**: 150.100.0.0/16.
 - The **IP of the gateway** (leading to external WAN/Internet): 150.100.50.1
 - A free **IP for our network**: 150.100.50.10
- We have now to configure our router and our devices to setup the new LAN.
 - We are considering a **NAT-enabled router**.
 - We decide that **the IP of the new LAN** will be: 192.168.1.0/24 (reserved for local networks).



Network Layer

Internet Protocol: LAN Setup Example

- Our **local router has 2 interfaces**, one for the WAN-side and another for the LAN-side.
- Let's start by configuring our local router's **WAN-side settings**.
- On the router we have to specify the **following parameters**:
 - The IP address of the router in the WAN.
 - The subnet of the WAN.
 - The gateway.
 - One or more DNS.
- Notice that we can also specify a **dynamic IP** if the WAN has a DHCP service.

Internet Connection

Set up an internet connection with the service information provided by your ISP (internet service provider).

Internet Connection Type:	Static IP	
	Select this type if your ISP provides specific IP parameters.	
IP Address:	150.100.50.10	
Subnet Mask:	255.255.0.0	
Default Gateway:	150.100.50.1	
Primary DNS:	8.8.8.8	
Secondary DNS:	4.4.4.4	(Optional)

Network Layer

Internet Protocol: LAN Setup Example

- Now let's configure the **LAN-side settings**.
- We have to **give an IP** to the LAN-side interface of the router:
 - Since our network will be addressed as 192.168.1.0/24, we will give 192.168.1.1/24 to the router (it is typical to give .1 to it).
- In this example, our router also provide **DHCP service**, we can then specify:
 - **Address pool** (for DHCP hosts).
 - **Lease time** (timeout of the IP assignment, after that IP can be reused).
 - **Gateway** and **DNS** to be communicated to the hosts.

LAN

View and configure LAN settings.

MAC Address: 48-22-54-16-18-7D

IP Address: 192.168.1.1

Subnet Mask: 255.255.255.0

DHCP Server

Dynamically assign IP addresses to the devices connected to the router.

DHCP Server: ☒ Enable

IP Address Pool: 192.168.1.100 - 192.168.1.250

Address Lease Time: 120 minutes

Default Gateway: 192.168.1.1 (Optional)

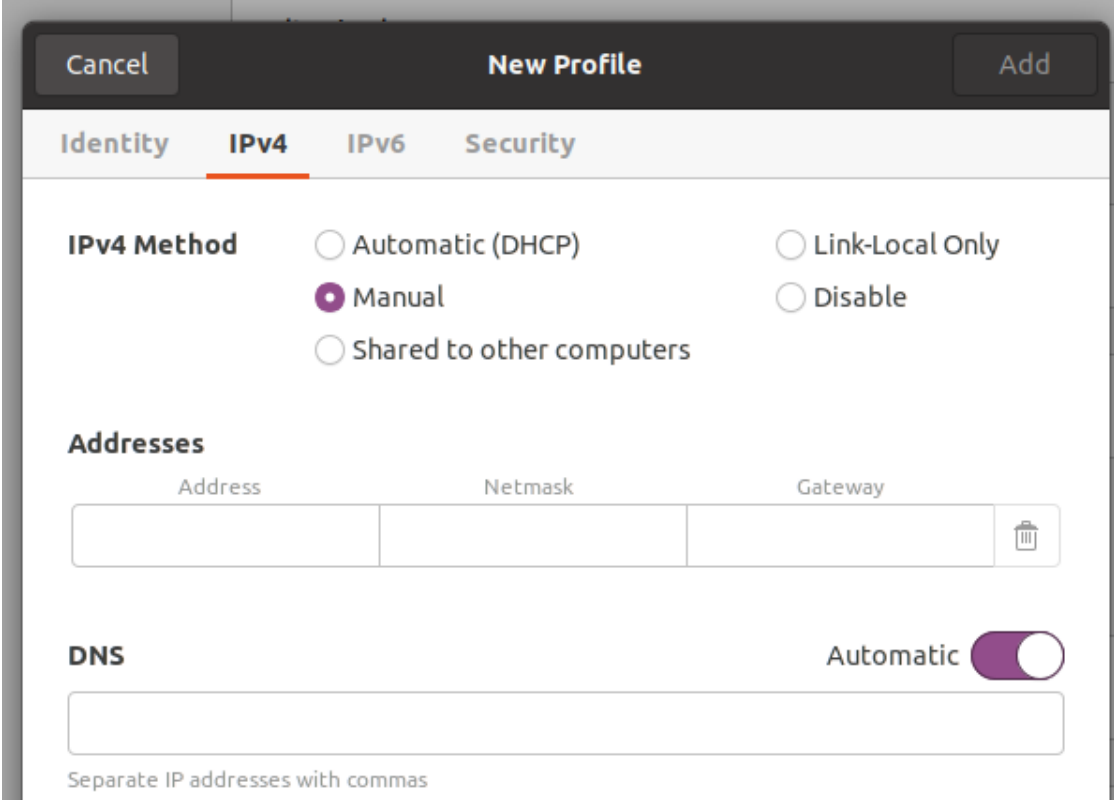
Primary DNS: (Optional)

Secondary DNS: (Optional)

Network Layer

Internet Protocol: LAN Setup Example

- This is a simple example of how to configure this connection on a **new host** (Ubuntu 20.04).
- There are different methods to choose:
 - **Automatic (DHCP)**: uses DHCP to automatically configure the connection (you will get an IP from 192.168.1.100 to 192.168.1.250).
 - **Manual**: setup the connection by manually setting the configuration.



The screenshot shows the 'New Profile' window for network configuration. The 'IPv4' tab is selected, showing the 'IPv4 Method' section with three radio buttons: 'Automatic (DHCP)', 'Manual' (which is selected), and 'Shared to other computers'. To the right, there are two more radio buttons: 'Link-Local Only' and 'Disable'. Below this is the 'Addresses' section with three input fields for 'Address', 'Netmask', and 'Gateway', and a trash icon. At the bottom, the 'DNS' section has a text input field and a toggle switch labeled 'Automatic' which is turned on. A note at the bottom says 'Separate IP addresses with commas'.

Network Layer

Internet Protocol: LAN Setup Example

- In a **manual setting** we need to know the network configuration (and the available IPs).
- We have to specify the main information for **connection setup**:
 - IP address of the host (static/fixed).
 - Subnet mask.
 - Gateway.
 - DNS (one or more).

The screenshot shows the 'New Profile' window for network configuration. The 'IPv4' tab is selected. Under 'IPv4 Method', the 'Manual' option is chosen. The 'Addresses' table contains one entry with IP 192.168.1.10, Netmask 255.255.255.0, and Gateway 192.168.1.1. The 'DNS' section has a text field containing '8.8.8.8' and an 'Automatic' toggle switch.

Address	Netmask	Gateway	
192.168.1.10	255.255.255.0	192.168.1.1	

DNS Automatic ☐

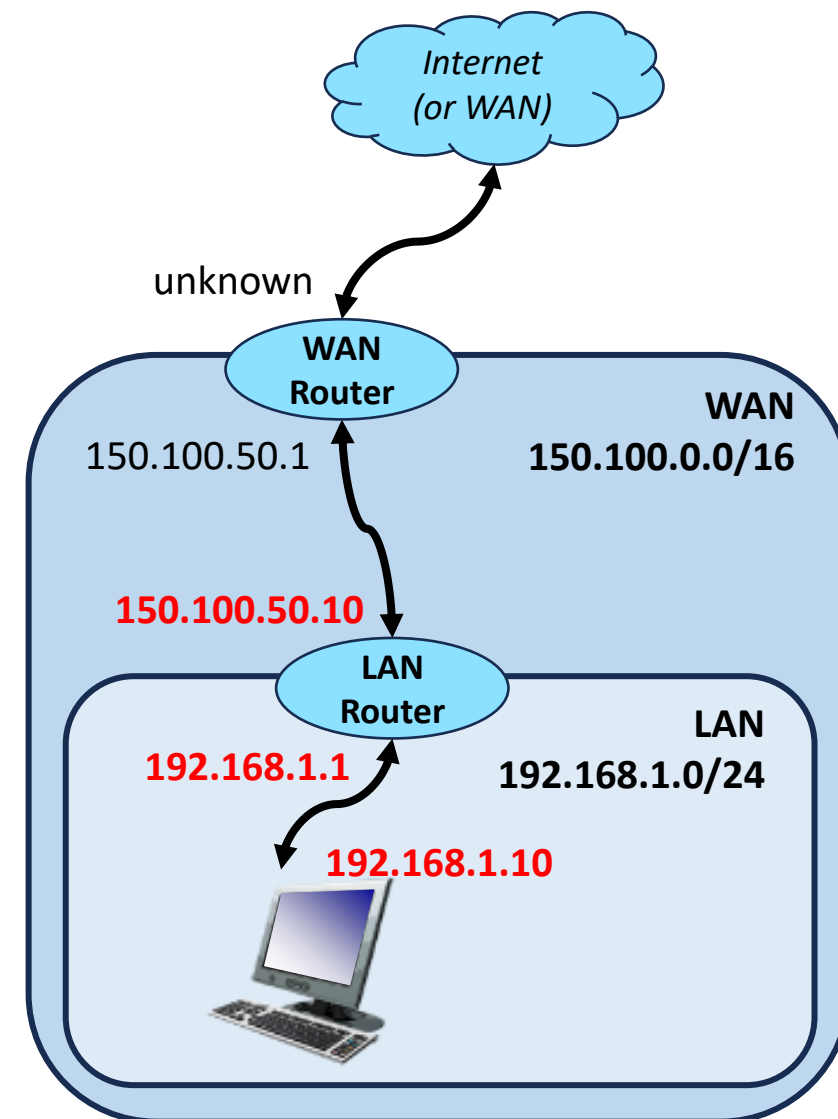
8.8.8.8

Separate IP addresses with commas

Network Layer

Internet Protocol: LAN Setup Example

- In this created LAN we can **connect new hosts to the LAN** by specifying a manual IP address (outside the DHCP pool) or an automatic IP address (inside the DHCP pool).
- The connected **hosts will send their packets** to the local router (unaware of the outside WAN).
- We have the **local router configured to forward packets** to the WAN router (and possibly to the Internet).



Network Layer

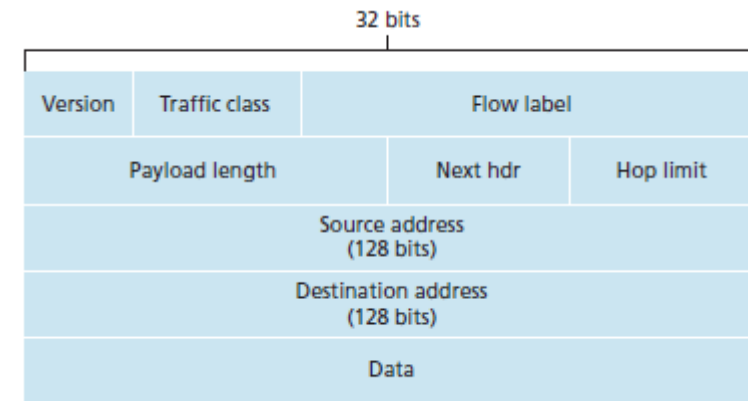
Internet Protocol: IPv6

- In the early 1990s, the Internet Engineering Task Force (IETF), which is an ONG founded in 1986, **began an effort to develop a successor to the IPv4** protocol to face the 32-bit **IPv4 address shortage**.
- IPv6 was designed to **ensure more addresses available**, but it was also an opportunity to **update other aspects of IPv4**, based on the accumulated operational experience.
- It is **not sure when exactly all the available IPv4 would be exhausted** (it was speculated to be 2008 or 2018, but we have still some IPs left).
- But **what about IPv5?** It was proposed in 1979 and was based on the experimental Internet Stream Protocol (ST). It was still relying on 32bit addresses, so it was **dropped mainly because of addresses shortage**.

Network Layer

Internet Protocol: IPv6 Datagram

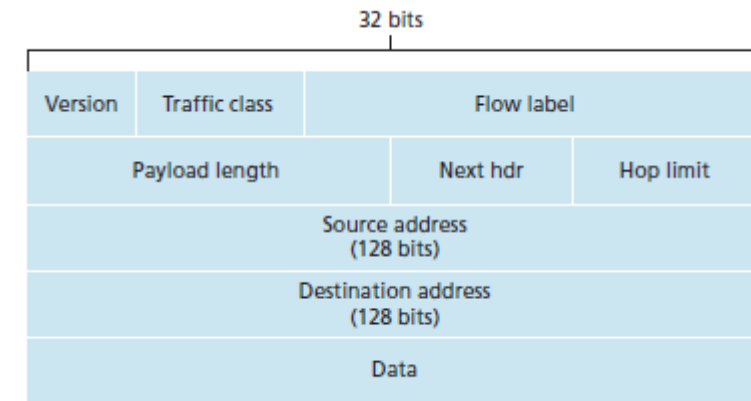
- IPv6 has the following **advantages**:
 - **Expanded addressing capabilities**: from 32 to 128 bits (i.e., $3.4028237e+38$), which ensures that the world won't run out of IP addresses, as every grain of sand on the planet can be IP-addressable.
 - **A fixed-length 40-byte header**: some of IPv4 fields have been dropped or made optional.
 - **Flow labeling**: packets from some applications (e.g., audio/video streaming) can be grouped into a unique with flow having specific identification.
 - The precise role and usage of flows is still discussed.



Network Layer

Internet Protocol: IPv6 Datagram

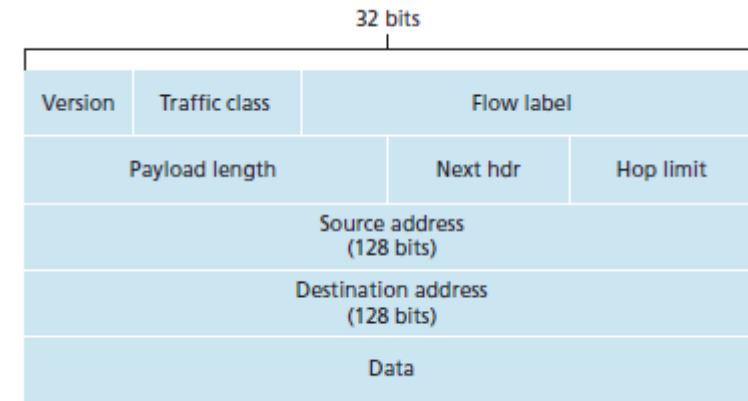
- IPv6 datagram is composed by the following fields:
 - **Version** (4 bits): identifies the **IP version** number. Not surprisingly, IPv6 carries a value of 6 in this field.
 - **Traffic class** (8 bits): like the TOS field in IPv4, can be used to give **priority to datagrams** (e.g., voice-over-IP over SMTP, etc.).
 - **Flow label** (20 bits): identify a **flow** of datagrams.
 - **Payload length** (16 bits): the **number of bytes of the payload** (40-bytes header excluded).
 - **Next header** (8 bits): identifies the **upper-level protocol** to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP). Similar to protocol field in the IPv4 header.
 - **Hop limit** (8 bits): Specifies the **time-to-live** as in TTL field of IPv4 (decremented every forward).
 - **Source and destination addresses** (128+128 bits): the IPv6 128-bit **addresses**.
 - **Data** (variable): the **payload** portion of the datagram.



Network Layer

Internet Protocol: IPv6 Datagram

- What is **missing** from IPv4:
 - **Fragmentation-related fields:** IPv6 does not allow fragmentation and reassembly on intermediate routers, but only on hosts.
 - If an IPv6 datagram received by a router is too large to be forwarded, the **router simply drops the datagram and sends a “Packet Too Big” error** message back to the sender. The sender can then resend the data, using a smaller IP datagram size.
 - **Header checksum:** it takes time on routers, and it is redundant as **link-layer protocols typically perform checksum** on the whole packet.
 - **Options:** are no longer a part of the standard IP header, **some options can be specified into next header** field (along with TCP/UDP identifiers).



Network Layer

Internet Protocol: From IPv4 to IPv6

- There is one big issue with IPv6: **it is not backward-compatible**. IPv4-capable systems are unable to handle IPv6 datagrams.
- How will Internet adopt IPv6?
 - **The flag day approach**: in a given time and date **all Internet machines would be turned off and upgraded from IPv4 to IPv6**. A similar approach have been used when TCP was introduced but it was a mess (even for the small internet network of the time). A flag day involving billions of devices is even more unthinkable today.
 - **The tunneling approach**: the use of specific routers (tunnels) in charge of **mapping IPv4 datagrams into IPv6 datagrams** and vice versa in almost-transparent way. This is probably the most realistic approach, already **used in practice**.
- While the adoption of IPv6 was initially slow, it is accelerating. Google reports that **about 45% of clients (in 2023) accessing Google** services use IPv6 (<https://www.google.com/intl/en/ipv6/statistics.html>).