
S1E2 - IPv4 Addressing

Nicholas Morrison nick@nanocat.net



IP: Internet Protocol

- Launched in 1974, by Vint Cerf and Bob Kahn.
- IP is the Layer 3 protocol supporting Layer 4 protocols such as:
 - TCP
 - UDP
 - GRE
 - ESP
 - ... and many more
- Other protocols that do a similar job to IP:
 - IPX/SPX
 - CLNS
- “Layer 3” and “Layer 4” refer to the OSI Layer Model

The OSI Layer Model (in short)

- OSI = Open Systems Interconnection
- Standard of the ISO (International Organisation for Standardization): <https://www.iso.org/about-us.html>
- Download it from here: <https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html> (search for the string “7498-1:1994”)

Quoting the standard:

” 6.1.4. The highest is the Application Layer and it consists of the application-entities that cooperate in the OSI Environment. The lower layers provide the services through which the application entities cooperate.

6.1.5. Layers 1-6, together with the physical media for OSI provide a step-by-step enhancement of communication services. The boundary between layers identifies a stage in this enhancement process at which an OSI service standard is defined while the functioning of the layers is governed by OSI protocol standards. ”

The Seven OSI Layers

Layer 7: Application	HTTP, HTTPS, IMAP, WebSockets
Layer 6: Presentation	MIME, SSL/TLS, ASCII, MPEG
Layer 5: Session	TCP Sockets, named pipes, RPC
Layer 4: Transport	TCP, UDP, GRE, ESP
Layer 3: Network	IP, ICMP, IPSEC, OSPF, RIP
Layer 2: Data Link	ARP, VLAN (802.1q), PPP, IS-IS
Layer 1: Physical	RJ45, RS-232, 802.3 PHY (10BASE-T,

100BASE-T, 1000BASE-T), DOCSIS,
DWDM

Tools for converting between bases

- <https://www.rapidtables.com/convert/number/decimal-to-binary.html>
- macOS Calculator (Programmer mode)
- linux, macOS: `bc` (`apt install bc` / `brew install bc`)

```
$ echo "obase=2;192" | bc
11000000
$ echo "ibase=2;11000000" | bc
192
$
```

IP Addresses

- IP addresses are used to identify Layer 3 interfaces on an IP network
- They have a fixed length of 32 bits,
- usually written as four groups of eight bits with a dot between each eight:

```
11000000.10101000.00000000.00000001
```

except converted to decimal:

```
192.    168.    0.    1
```

- That ^ is called a “dotted quad”
- You can also express it as a 32-bit integer, if you want to:

```
3232235521
```

IPv4 and IPv6

- IPv4 addresses: 32 bits long
- 32 bits = 4,294,967,295 combinations of on or off, 1 or 0
- => only 4,294,967,295 IPv4 addresses (minus reservations)
- ... but we have more devices than that
- tricks for sharing IPv4 addresses: NAT (Network Address Translation)
- IPv6 addresses: 128 bits long
- 128 bits = 340,282,366,920,938,463,463,374,607,431,768,211,456 (340 trillion trillion trillion) combinations of on or off
- => 340,282,366,920,938,463,463,374,607,431,768,211,456 IPv6 addresses
- There are maybe 400 billion stars in the milky way.

Network Masks

- IP addresses are often written along with a network mask.

192.168.0.1 255.255.255.0

or,

192.168.0.1/24

- 32 bits long (or 128 bits long for IPv6).

255.255.255.0 = 11111111.11111111.11111111.00000000

- Clarifies the address as “network part” and “host part”

Network Masks

- Same “network” part => communicate directly, no Router
- Different “network” part => must use a Router to communicate
- No gaps allowed in the bits in the network mask!

IP Address Example: 203.33.18.1/24

- 203.33.18.1/24

203 . 33 . 18 . 1 /24
11001011.00100001.00010010.00000001 /24

- the first **24** bits are the network part
- the remaining (32 - 24 = 8) bits are the host part

203 . 33 . 18 . 0 /24
11001011.00100001.00010010.00000000 /24

=====
=====

<= network part, written as 203.33.18.0/24

203 . 33 . 18 . 1 /24
11001011.00100001.00010010.00000001 /24

=====

<= host part, written as 203.33.18.1/24

IP Address Example: 192.168.56.43/16

- 192.168.56.43/16

```
192 . 168 . 56 . 43 /16
11000000.10101000.00111000.00101011 /16
```

- the first **16** bits are the network part
- the remaining (32 - 16 = 16) bits are the host part

```
192 . 168 . 0 . 0 /16
11000000.10101000.00000000.00000000 /16
=====.
```

<= network part, written as: 192.168.0.0/16

```
192 . 168 . 56 . 43 /16
11000000.10101000.00111000.00101011 /16
=====.
```

<= host part, written as: 192.168.56.43/16

IP Address Example: 150.101.5.6/23

- 150.101.5.6/23

```
150 . 101 . 5 . 6 /23
10010110.01100101.00000101.00000110 /23
```

- the first 23 bits are the network part
- the remaining (32 - 23 = 9) bits are the host part

```
150 . 101 . 4 . 0 /23
10010110.01100101.00000100.00000000 /23
=====.
```

<= network part, written as 150.101.4.0/23

```
150 . 101 . 5 . 6 /23
10010110.01100101.00000101.00000110 /23
=.
```

<= network part, written as 150.101.5.6/23

Finding your own IP address(es)

- Linux: ip a
- Windows: ipconfig
- macOS: ifconfig

Questions

- What is your local IP address?
- What is your network mask?
- How many unique IP addresses are available in your local network?

linux: `ip a` | windows: `ipconfig` | macos: `ifconfig`

Handy tool: `ipcalc`