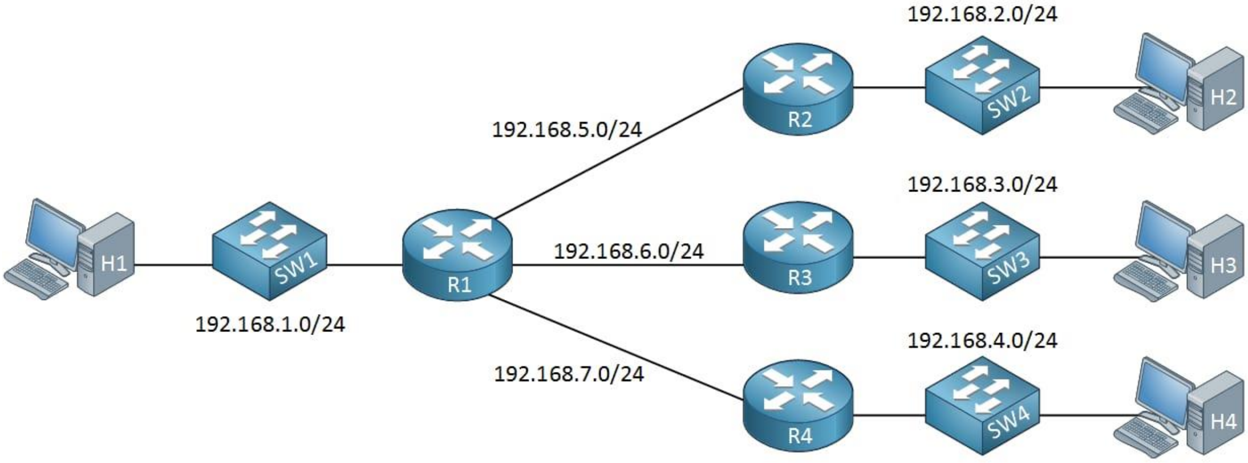
# What is Subnetting?

Subnetting means **dividing one big network into smaller networks**.



## Why? Because big networks have problems like:

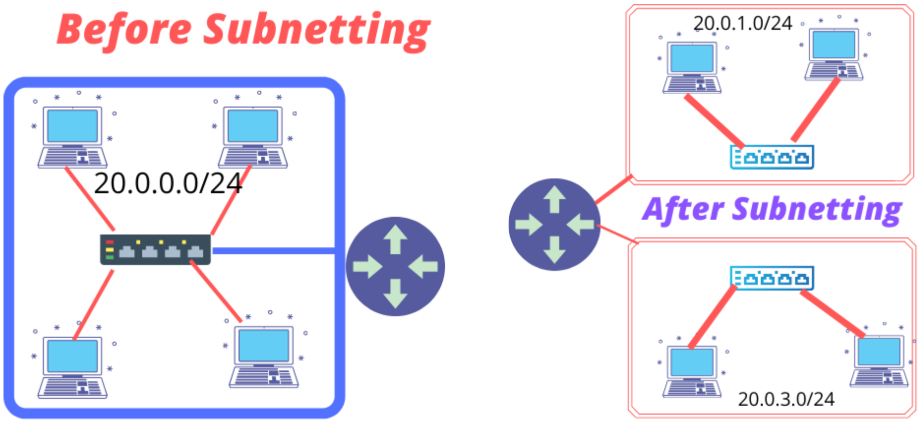
- Too many devices → more traffic → slow performance
- Difficult to manage
- Security issues (all devices in one network)

So, subnetting helps us create **smaller, organized networks** from one large network.

---

## Why do we use Subnetting?

- To **save IP addresses** (efficient use)
- To **separate departments** (e.g., HR, IT, Sales)
- To **improve security and performance**



---

## How does Subnetting work?

Every IP address has **two parts**:

- **Network part** (identifies the network)
- **Host part** (identifies the device)

192 . 168 . 0 . 2

An IP address consists of two parts.

Network address Host address.

255 . 255 . 255 . 0

A subnet mask reveals how many bits in the IP address are used for the network by masking the network portion of the IP address.

The **subnet mask** tells which part is network and which part is host.

Example of subnet masks:

- Class A → 255.0.0.0
- Class B → 255.255.0.0
- Class C → 255.255.255.0

---

## Subnetting Example

**Given:**
IP: **192.168.10.0/24** (Class C)
Subnet mask: **255.255.255.0**
Total IPs: 256 (192.168.10.0 – 192.168.10.255)

**We need 4 subnets**.

---

# Step 1: How many bits do we borrow?

Formula:
**$2^n \geq$ number of subnets needed**
$2^2 = 4$ → So, we borrow **2 bits** from host part.

---

# Step 2: New Subnet Mask

Default /24 → after borrowing 2 bits → **/26**
New subnet mask = **255.255.255.192**
(Binary: 11111111.11111111.11111111.11000000)

## Tips to Make it Easy

- **binary chart** :
  128 | 64 | 32 | 16 | 8 | 4 | 2 | 1
- Show how subnet mask changes from 11111111.11111111.11111111.00000000 (/24) to 11111111.11111111.11111111.11000000 (/26).
- Use colors or markers to highlight **network bits vs host bits**.

---

# Step 3: Subnet Increment

Increment = 256 - 192 = 64
So, each subnet increases by 64.

---

## Step 4: Calculate Subnets and Host Range

- Subnet 1: 192.168.10.0 – 192.168.10.63
  (Network: 192.168.10.0, Broadcast: 192.168.10.63, Usable: .1 to .62)
- Subnet 2: 192.168.10.64 – 192.168.10.127
  (Network: 192.168.10.64, Broadcast: 192.168.10.127, Usable: .65 to .126)
- Subnet 3: 192.168.10.128 – 192.168.10.191
  (Network: 192.168.10.128, Broadcast: 192.168.10.191, Usable: .129 to .190)
- Subnet 4: 192.168.10.192 – 192.168.10.255
  (Network: 192.168.10.192, Broadcast: 192.168.10.255, Usable: .193 to .254)

---

## Step 5: Hosts per Subnet

Formula:
2^host_bits - 2
Host bits left = 6
So: $2^6$ - 2 = **62 usable hosts per subnet**

---

### Summary

- Original Network: 192.168.10.0/24
- New mask: /26 (255.255.255.192)
- Subnets: 4
- Hosts per subnet: 62
- Subnets:
  - 192.168.10.0
  - 192.168.10.64
  - 192.168.10.128
  - 192.168.10.192

---

**In Short:**
Subnetting = Break one big network into smaller networks by changing subnet mask.

# Given Network

- **IP Address:** 10.0.0.0
- **Default Mask:** 255.0.0.0 (/8)
- **Total Addresses:** 16,777,216 ($2^{32}$ - $2^{8}$ for hosts)
- This is a **huge network**, so subnetting is required to create smaller networks.

---

## Example Scenario

We need to create **8 subnets** from 10.0.0.0.

---

## Step 1: Borrow Bits

Formula: $2^n \geq$ required subnets
We need 8 subnets → $2^3 = 8$
So, we borrow **3 bits** from the host portion.

- Default prefix = /8
- After borrowing 3 bits → /11
- New subnet mask = **255.224.0.0**
  (Binary: 11111111.11100000.00000000.00000000)

## Step 2: Calculate Increment

Increment = 256 - 224 = **32**
This increment is in the **second octet**.

---

## Step 3: Subnet Details

Now, we list the 8 subnets:

| Subnet No. | Network Address | First Host | Last Host | Broadcast Address |
|---|---|---|---|---|
| **1** | 10.0.0.0 | 10.0.0.1 | 10.31.255.254 | 10.31.255.255 |

| | | | | |
|---|---|---|---|---|
| **2** | 10.32.0.0 | 10.32.0.1 | 10.63.255.254 | 10.63.255.255 |
| **3** | 10.64.0.0 | 10.64.0.1 | 10.95.255.254 | 10.95.255.255 |
| **4** | 10.96.0.0 | 10.96.0.1 | 10.127.255.254 | 10.127.255.255 |
| **5** | 10.128.0.0 | 10.128.0.1 | 10.159.255.254 | 10.159.255.255 |
| **6** | 10.160.0.0 | 10.160.0.1 | 10.191.255.254 | 10.191.255.255 |
| **7** | 10.192.0.0 | 10.192.0.1 | 10.223.255.254 | 10.223.255.255 |
| **8** | 10.224.0.0 | 10.224.0.1 | 10.255.255.254 | 10.255.255.255 |

## Hosts per Subnet

- Remaining host bits = 32 - 11 = 21
- Hosts per subnet = $2^{21}$ - 2 = **2,097,150 usable hosts per subnet**
  (That's huge!)

## Summary

- Original Network: 10.0.0.0/8
- New Subnet Mask: 255.224.0.0 (/11)
- Number of Subnets: 8
- Hosts per Subnet: ~2 million

## Given Network

- **IP Address:** 10.0.0.0
- **Default Mask:** 255.0.0.0 (/8)
- **Total Addresses:** 16,777,216 ($2^{32} - 2^{8}$ for hosts)
- This is a **huge network**, so subnetting is required to create smaller networks.

### Example Scenario

We need to create **8 subnets** from 10.0.0.0.

---

## Step 1: Borrow Bits

Formula: $2^n \geq$ required subnets
We need 8 subnets → $2^3 = 8$
So, we borrow **3 bits** from the host portion.

- Default prefix = /8
- After borrowing 3 bits → /11
- New subnet mask = **255.224.0.0**
  (Binary: 11111111.11100000.00000000.00000000)

---

## Step 2: Calculate Increment

Increment = 256 - 224 = **32**
This increment is in the **second octet**.

---

## Step 3: Subnet Details

Now, we list the 8 subnets:

| Subnet No. | Network Address | First Host | Last Host | Broadcast Address |
|---|---|---|---|---|
| 1 | 10.0.0.0 | 10.0.0.1 | 10.31.255.254 | 10.31.255.255 |
| 2 | 10.32.0.0 | 10.32.0.1 | 10.63.255.254 | 10.63.255.255 |
| 3 | 10.64.0.0 | 10.64.0.1 | 10.95.255.254 | 10.95.255.255 |
| 4 | 10.96.0.0 | 10.96.0.1 | 10.127.255.254 | 10.127.255.255 |
| 5 | 10.128.0.0 | 10.128.0.1 | 10.159.255.254 | 10.159.255.255 |
| 6 | 10.160.0.0 | 10.160.0.1 | 10.191.255.254 | 10.191.255.255 |
| 7 | 10.192.0.0 | 10.192.0.1 | 10.223.255.254 | 10.223.255.255 |
| 8 | 10.224.0.0 | 10.224.0.1 | 10.255.255.254 | 10.255.255.255 |

## Hosts per Subnet

- Remaining host bits = 32 - 11 = 21
- Hosts per subnet = $2^{21}$ - 2 = **2,097,150 usable hosts per subnet**
  (That's huge!)

---

## Summary

- Original Network: 10.0.0.0/8
- New Subnet Mask: 255.224.0.0 (/11)
- Number of Subnets: 8
- Hosts per Subnet: ~2 million

8 8 8

Class 203·124·46·0̄ /24

255·255·255·0

128 68 32 16 8 4 2 1

203·124·46·0·0·0·0·0·0·0·0 /24

**1 bit subnit :- $2^n$ $2^1 = 2$ → Combination.**

128 | 64 32 16 8 4 2 1

203·124·46·0 | 0·0·0·0·0·0·0 /24

1) 203·124·46·0 | 0 0 0 0 0 0 0

2) 203·124·46·1 | 0 0 0 0 0 0 0

↳ In network ID these all bits will be off and in broadcost these all will be on.

(1) 203·124·46·0 /25 → N/w ID

46·1

46·2

:

203·124·46·127/25 → B·Cast ID

} 32-25=7 $2^7=128$

32-26=6 $2^6=64$

(2) 203·124·46·128 /25 → N·ID

46·29

46·30

203·124·46·255 /25 → B·ID

* If we want to borrow 2 bits subnet. Then according to formula ↷

$\Rightarrow 2^n = 2^2 = 4$

Then the Combinations will be 4.

(1) 203·124·46·0 | 0 | 0·0·0·0·0·0

(2) 203·124·46·1 | 0 | 0·0·0·0·0·0

(3) 203·124·46·0 | 1 | 0·0·0·0·0·0

(4) 203·124·46·1 | 1 | 0·0·0·0·0·0

## Class-C Network 2-bits borrow

205·5·5·0 /24 $2^2 = 4 \rightarrow 4$ Combinans

255·255·255·0 /24 $2^n \rightarrow n \rightarrow$ No of n/w bits.

205·5·5· $\overset{128}{0}\,\overset{64}{0}$·$\overset{32}{0}\,\overset{16}{0}\,\overset{8}{0}\,\overset{4}{0}\,\overset{2}{0}\,\overset{1}{0}$ /24

1) 205·5·5·0·0 | 0 0 0 0 0 0
2) 205·5·5·0·1 | 0 0 0 0 0 0
3) 205·5·5·1·0 | 0 0 0 0 0 0
4) 205·5·5·1·1 | $\overset{32}{0}\ \overset{16}{0}\ \overset{8}{0}\,\overset{4}{0}\ \overset{2}{0}\ \overset{1}{0}$

(1) 205·5·5·0 /26
205·5·5·1 → when host all bits are off mean 0
5·2 } 64
205·5·5·63 /26 → B/ID
↳ when all remaing bits are ON.

2) 205·5·5·64 /26 → N/ID
205·5·5·65
205·5·5·126 } 64
205·5·5·127/26 B/C ID

(3) 205·5·5·128 /26 → N/w ID
5·128
5·190 } 64
205·5·5·191 → B/C ID

(4) 205·5·5·192 /26
5·193
5·258 } 64
205·5·5·255 /26

So we Convert one n/w into 4 n/w