TCP/IP Network Addressing.

Networking tools for troubleshooting and monitoring

TCP/IP Network Addressing.

Модель OSI(Open Systems Interconnection)

Модель OSI має **сім рівнів**: від фізичного до прикладного. Вона є концептуальною моделлю для стандартизації взаємодії систем. Це - **ідеальний** довідник.

1. **Прикладний (Application)**
2. **Представлення (Presentation)**
3. **Сеансовий (Session)**
4. **Транспортний (Transport)**
5. **Мережевий (Network)**
6. **Канальний (Data Link)**
7. **Фізичний (Physical)**

Модель TCP/IP (Transmission Control Protocol/Internet Protocol)

Модель TCP/IP є **практичною** і має **чотири рівні**. Вона значно спрощує OSI і є тією моделлю, за якою реально працює інтернет:

- **Прикладний Рівень (Application Layer):** Тут ми маємо протоколи, які взаємодіють безпосередньо з додатками: **HTTP**, **DNS**, **SMTP**, **SSH**. Це рівень, на якому працює більшість наших мікро сервісів та веб-додатків.
- **Транспортний Рівень (Transport Layer):** Відповідає за *надійну* (TCP) або *швидку* (UDP) доставку даних між хостами. **TCP** (Transmission Control Protocol) забезпечує встановлення з'єднання, гарантовану доставку та нумерацію пакетів. **UDP** (User Datagram Protocol) швидший, але не гарантований - ідеально для потокового відео чи DNS-запитів.
- **Міжмережевий Рівень (Internet Layer):** Він відповідає за **логічну адресацію** (IP-адреси) та **маршрутизацію** пакетів між різними мережами. Тут панує протокол **IP** (Internet Protocol) та його супутник **ICMP**.
- **Рівень Мережевих Інтерфейсів (Network Access Layer):** Відповідає за фізичну передачу даних по мережевому середовищу (Ethernet, Wi-Fi). Тут ми бачимо **MAC-адреси** (Media Access Control) та протокол **ARP**.

Що таке IP-Адресація? IPv4 vs IPv6

IP-адреса — це унікальний числовий ідентифікатор пристрою у мережі. Це як **поштова адреса** для вашого комп'ютера. Без неї трафік не знайде свого отримувача. Ми повинні чітко розрізняти дві версії: **IPv4** та **IPv6**.

IPv4 (Internet Protocol Version 4)

- **Формат:** Це 32-бітне число, яке ми звикли бачити у вигляді **чотирьох октетів**, розділених крапками. Наприклад: `192.168.1.10`. Кожен октет – це 8 біт, що дає значення від 0 до 255.
- **Кількість Адрес:** Теоретично, близько 4.3 мільярда адрес.
- **Проблема:** Через вичерпання адрес (адресний голод) виникла необхідність у **NAT** (Network Address Translation).

IPv6 (Internet Protocol Version 6)

- **Формат:** Це 128-бітне число, записане у **вісім груп по чотири шістнадцяткові цифри**, розділених двокрапками. Наприклад: `2001:0db8:85a3:0000:0000:8a2e:0370:7334`
- **Кількість Адрес:** Величезна кількість адрес
- **Переваги:** Вирішує проблему адресного голоду, спрощує маршрутизацію (завдяки меншій потребі у NAT) та має вбудовану підтримку безпеки (IPsec).

Класифікація IPv4-Адрес та Приватні Діапазони

Хоча класова адресація **(Classful Addressing)**, де мережі поділялися на класи **A, B, C**, здебільшого замінена **CIDR (Classless Inter-Domain Routing)**, знати її необхідно для розуміння старих систем та принципів.

Класифікація IPv4 (Classful Addressing)

Клас	Перший Октет (Діапазон)	Призначення
A	1–126	Дуже великі мережі. Багато хостів, мало мереж.
B	128–191	Мережі середнього розміру.
C	192–223	Малі мережі. Багато мереж, мало хостів.
D	224–239	Мультикаст (Multicast) — для групової розсилки.
E	240–255	Резерв, експериментальне використання.

Приватні Діапазони (Private IP Ranges)

Клас	Приватний Діапазон	CIDR
A	10.0.0.0–10.255.255.255	10.0.0.0/8
B	172.16.0.0–172.31.255.255	172.16.0.0/12
C	192.168.0.0–192.168.255.255	192.168.0.0/16

Маска Підмережі (Subnet Mask)

Маска підмережі — це 32-бітне число, яке працює як **бінарний фільтр**. Вона дозволяє будь-якому пристрою визначити, яка частина його IP-адреси належить до **адреси мережі**, а яка — до **адреси хоста** (конкретного пристрою). Це фундаментальний принцип маршрутизації.

- **1 (у масці):** Відповідає біту в IP-адресі, який є частиною **ідентифікатора мережі** (Network ID).
- **0 (у масці):** Відповідає біту в IP-адресі, який є частиною **ідентифікатора хоста** (Host ID).

Щоб знайти мережеву адресу, ми виконуємо операцію **побітового "І" (AND)** між IP-адресою та маскою підмережі. Якщо результат операції збігається з мережевою адресою, пакет залишається в локальній мережі; якщо ні — він передається маршрутизатору.

Префікс CIDR

CIDR (Classless Inter-Domain Routing)

Сучасний підхід – це **CIDR**, або **бескласова міждоменна маршрутизація**. Він замінив стару класову систему A, B, C, зробивши використання IP-адрес набагато ефективнішим.

Префікс CIDR записується у форматі **IP-адреса/N**, де **N** — це **довжина префікса мережі**, або просто **кількість послідовних "одиниць"** у масці підмережі.

Приклад CIDR	Маска Підмережі	Опис
/8	255.0.0.0	8 біт на мережу, 24 на хост. Великі мережі.
/16	255.255.0.0	16 біт на мережу, 16 на хост. Типовий розмір VPC.
/24	255.255.255.0	24 біти на мережу, 8 на хост. Найпоширеніша домашня/офісна мережа.
/30	255.255.255.252	30 біт на мережу. Використовується для P2P-з'єднань маршрутизаторів.

Subnetting

Це процес поділу однієї великої мережі на менші, ефективніші підмережі. Це необхідно для **безпеки** (сегментація трафіку), **продуктивності** (зменшення широкомовного домену) та **економії IP-адрес**.

Як працює Subnetting?

Ми **«запозичуємо»** біти з частини **Host ID** і перетворюємо їх на частину **Network ID**.

Припустимо, у нас є мережа **192.168.1.0/24**.

- Маска: 255.255.255.0.
- На хости виділено **8 біт** (від 2^0 до 2^7). Це 2^8=256 адрес.

Якщо ми вирішимо розділити цю мережу на 4 менші підмережі, ми повинні запозичити **2 біти** з частини хоста, оскільки 2^2=4.

Нова довжина префікса буде 24+2=/26.

Subnetting

Нові Параметри /26

- **Нова Маска:** 255.255.255.192.
- **Розмір Блоку (Block Size):** 256−192=64 адреси.
- **Хостів на Підмережу:** 2^6−2=62 доступних хостів. (Мінус 2 адреси: мережева та широкомовна).

CIDR-блоки підмереж /26

CIDR	Broadcast	Діапазон хостів
192.168.1.0/26	192.168.1.63	192.168.1.1 – 192.168.1.62
192.168.1.64/26	192.168.1.127	192.168.1.65 – 192.168.1.126
192.168.1.128/26	192.168.1.191	192.168.1.129 – 192.168.1.190
192.168.1.192/26	192.168.1.255	192.168.1.193 – 192.168.1.254

Спеціальні Адреси: Мережева, Широкомовна, Шлюз

Адреса Мережі (Network ID)

- **Призначення:** Це перша адреса в підмережі. Використовується для **ідентифікації** самої підмережі.
- **Приклад:** Для підмережі 192.168.1.0/24, адреса мережі — **192.168.1.0**.
- **Важливо:** Маршрутизатори використовують цю адресу для записів у своїх **таблицях маршрутизації**.

Широкомовна Адреса (Broadcast ID)

- **Призначення:** Це остання адреса в підмережі. Використовується для надсилання пакета **всім** пристроям одночасно в цій підмережі.
- **Приклад:** Для підмережі 192.168.1.0/24, широкомовна адреса — **192.168.1.255**.

Адреса Шлюзу (Gateway Address)

- **Призначення:** Це адреса маршрутизатора (або шлюзу), який служить точкою виходу з локальної мережі до інших мереж (включаючи Інтернет).
- **Приклад:** Це, як правило, **перша** або **остання** доступна IP-адреса хоста в підмережі.
    - Часто використовується: **192.168.1.1** або **192.168.1.254** (для /24).

Мережеві інструменти для діагностики та моніторингу

Протокол ICMP: Ping та Traceroute

ICMP (Internet Control Message Protocol) працює на **Міжмережевому рівні (Network Layer)**,, його функція - не передача даних, а **контроль** та **діагностика**. ICMP використовується для надсилання повідомлень про помилки та операційну інформацію.

Утиліта **`ping`** вико

```
# Приклад використання ping
ping -c 5 google.com

# Вивід: 5 packets transmitted, 5 received, 0% packet loss, time 4004ms
# rtt min/avg/max/mdev = 8.012/8.035/8.051/0.019 ms
```

и (Latency) мережі.

Утиліта **`traceroute`** використовується для відображення **маршруту** (всіх проміжних маршрутизаторів, або **хопів**), який прохо

```
# Приклад використання traceroute
traceroute 8.8.8.8
```

Аналіз Мережевих З'єднань: netstat та ss

Коли ми діагностуємо проблеми з додатками (наприклад, чому мій веб-сервер не слухає порт 8080?), нам потрібно зазирнути на **Транспортний Рівень (Transport Layer)**. Для цього ми використовуємо утиліти **`netstat`** та її сучасну, швидшу альтернативу **`ss`** (Socket Statistics).

утиліта **netstat (Network Statistics)** виводить інформацію про мережеві з'єднання, таблиці маршрутизації, статистику мережевих інтерфейсів і, головне, **відкриті порти**.

утиліта **ss (Socket Statistics) у** сучасних Linux-системах (особливо на високопродуктивних серверах) є кращею, оскільки вона швидше отримує дані безпосередньо з ядра, що критично для систем з тисячами з'єднань.

```
# Використовуємо ss для виведення всіх активних TCP/UDP сокетів
# -l (listening), -t (TCP), -u (UDP), -n (numeric), -p (processes)
ss -lntu | grep 8080
```

Конфігурація Інтерфейсів: ifconfig, ip

Для конфігурації та первинної діагностики мережевих інтерфейсів ми використовуємо команду **`ip`**, яка є сучасною заміною застарілої утиліти **`ifconfig`**.

ifconfig (Interface Configuration) - це стара утиліта, яка досі зустрічається у деяких старих дистрибутивах Linux. Вона використовується для перегляду та конфігурації мережевих інтерфейсів.

ip (IP utility) є частиною пакету **iproute2** і є стандартним інструментом у сучасних системах. Вона універсальна і дозволяє керувати адресами, маршрутизацією, тунелями та мережевими пристроями.

```
# Перегляд IP-адрес інтерфейсів
ip addr show eth0
```

```
# Перегляд таблиці маршрутизації
ip route show
```

Глибока Діагностика: mtr та tcpdump

Для більш глибокої та комплексної діагностики, коли `ping` та `traceroute` недостатньо, ми звертаємося до двох потужних інструментів: **`mtr`** та **`tcpdump`**.

mtr (My Traceroute) - об'єднує функціональність **`ping`** та **`traceroute`** в одному інтерактивному, безперервному інструменті.

- Вона безперервно відправляє пакети і виводить статистику для **кожного хопу** в реальному часі.
- Ми бачим

```
# Приклад використання mtr для безперервної діагностики
mtr -c 100 -r google.com
```

го маршрутизатора.

`tcpdump` - це утиліта командного рядка для **захоплення та аналізу мережевих пакетів**, що проходять через інтерфейс. Вона працює на **Канальному (Data Link)** та **Мережевому (Network)**рівнях, дозволяючи бачити

```
# Приклад tcpdump: Слухаємо трафік на порту 80 (HTTP) на інтерфейсі eth0
sudo tcpdump -i eth0 'port 80'
```

DHCP (Dynamic Host Configuration Protocol)

DHCP є протилежністю статичній адресації, дозволяючи автоматично призначати IP-адреси та інші мережеві параметри пристроям.

DHCP Server автоматично надає клієнтам:

1. **IP-адресу** (з пулу доступних адрес).
2. **Маску Підмережі**.
3. **Адресу Шлюзу (Default Gateway)**.
4. **Адресу DNS-Сервера**.

Процес отримання IP-адреси клієнтом називається **DORA** (це абревіатура, яку потрібно запам'ятати):

1. **Discover (Відкриття):** Клієнт надсилає широкомовний DHCP-Discover запит: *«Я тут, мені потрібна IP-адреса!»*
2. **Offer (Пропозиція):** DHCP-сервер надсилає широкомовний DHCP-Offer: *«Я пропоную тобі IP 192.168.1.10.»*
3. **Request (Запит):** Клієнт надсилає широкомовний DHCP-Request, підтверджуючи, що він приймає пропозицію: *«Я беру 192.168.1.10.»*
4. **Acknowledge (Підтвердження):** DHCP-сервер надсилає одноадресный DHCP-ACK, підтверджуючи, що адреса присвоєна і видає **Час Оренди (Lease Time)**.

Універсальний Мережевий Інструмент: nc

`netcat` (або `nc`) часто називають **«мережевим швейцарським ножем»**. Це простий інструмент для читання та запису даних через мережеві з'єднання, використовуючи TCP або UDP.

Ключові Сценарії:

- **Тестування Відкритого Порту:** Проста перевірка, чи слухає порт.
- **Створення Базового Сервера/Клієнта:** Швидка передача файлів або імітація з'єднання.

```
# 1. Тестування, чи порт 443 відкритий на зовнішньому хості
nc -vz google.com 443

# 2. Створення простого TCP-сервера для перевірки зв'язку
# Хост 1 (Сервер): nc -l 8888
# Хост 2 (Клієнт): nc [IP_Хоста_1] 8888  (і може відправляти текст)
```

Взаємодія з мережею на рівні додатка

cURL — це потужний інструмент, який підтримує безліч протоколів (HTTP, HTTPS, FTP, SMTP тощо) і використовується для передачі даних. Це наш основний інструмент для тестування веб-сервісів.

- Тестування різних HTTP-методів (GET, POST, PUT, DELETE).
- Відправка JSON- або XML-даних (тіла запиту).
- Перевірка HT

```
# 1. Проста перевірка доступності та HTTP-статусу
curl -I https://api.myservice.com/health
```

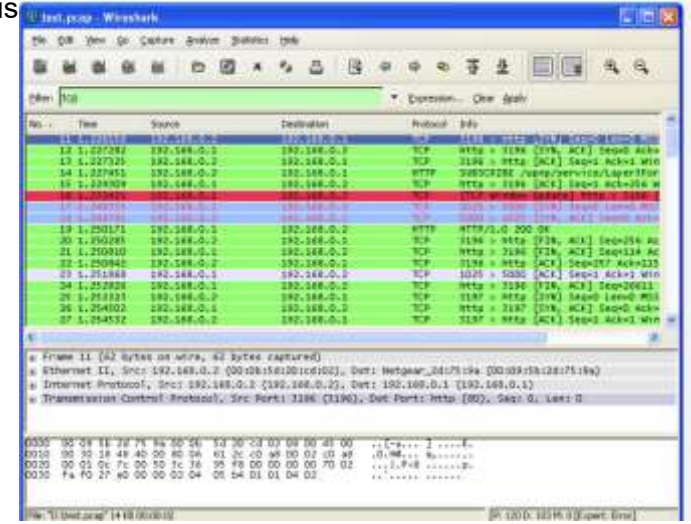
wget — в основному використовується для **рекурсивного завантаження** файлів або веб-сторінок. У DevOps ми часто використовуємо його в скриптах для завантаження конфігураційних файлів або бінарних файлів.

```
# Проста перевірка, чи можна завантажити файл (ігноруючи сертифікати SSL/TLS)
wget --no-check-certificate https://myinternalrepo.com/data.zip
```

Глибокий Аналіз Пакетів: Wireshark

Wireshark — це потужний аналізатор, який працює з `.pcap` файлами, захопленими за допомогою `tcpdump` або самого Wireshark. Його головні переваги:

1. **Протокольний Діссектор (Protocol Dissector):** Wireshark знає структуру тисяч мережевих протоколів (від TCP/IP до HTTP/2 та CNI Kubernetes) і відображає їх у зручній для читання ієрархічній структурі.
2. **Візуалізація:** Можливість графічно побудувати часові діаграми затримок, послідовності TCP-з'єднань та відстежувати retransmis



Q&A