

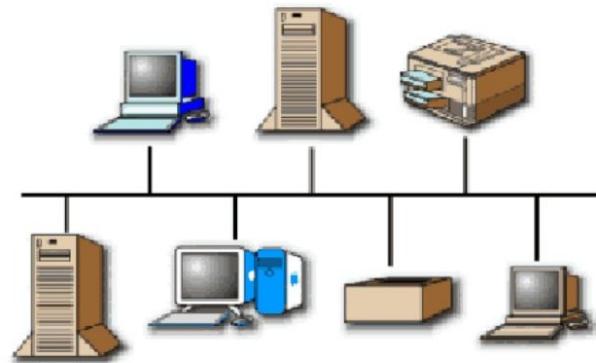
# Основи мереж

# Що таке комп'ютерна мережа?

Мережа — це набір комп'ютерів, принтерів, маршрутизаторів, комутаторів та інших пристроїв, які можуть обмінюватися даними один з одним через певне середовище передачі.

Зараз існує два основних типи мереж:

- Локальна мережа (LAN)
- Глобальна мережа (WAN)



# Локальна мережа (LAN)

Локальна мережа (LAN) — це група комп'ютерів і (LAN) мережевих пристройів зв'язку в межах обмеженої географічної області, наприклад офісної будівлі. *Тут немає участі третіх осіб.*

Вони характеризуються наступним:

- Висока швидкість передачі даних
- Як правило, менш дорогі технології
- Обмежена географічна зона

# Глобальна мережа (WAN)

Глобальна мережа (WAN) з'єднує локальні мережі. Вона не обмежена певною географічною територією та може бути пов'язана по всьому світу. *Задіяна стороння мережа.*

Вона характеризується наступним:

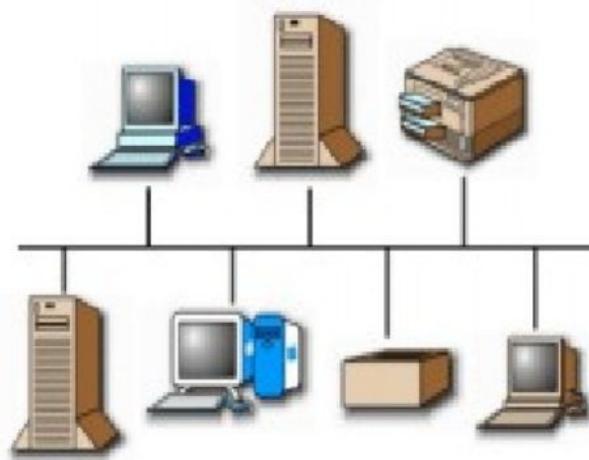
- Кілька взаємопов'язаних локальних мереж
- Як правило, дорожча технологія
- Більш складна у реалізації, ніж локальні мережі
- Існує в необмеженій географічній зоні
- Менша стійкість до помилок через відстань передачі

# Загальні топології локальної мережі

## Шинна архітектура

У шинній топології:

- один кабель з'єднує кожну робочу станцію лінійно, послідовно.
- сигнали транслюються на всі станції, але станції діють лише на адресовані їм кадри.

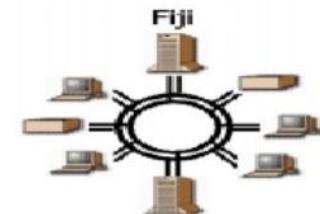


# Загальні топології локальної мережі

## Кільцева архітектура

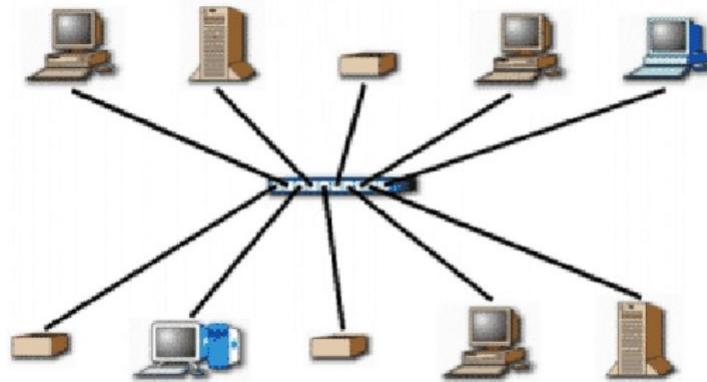
У кільцевій топології:

- Односпрямовані канали з'єднують передавальний бік одного пристрою з приймальним боком іншого пристрою.
- Пристрої передають кадри на наступний пристрій (нижчий учасник) у кільці.



# Зоряна топологія

У зіркоподібній топології кожна станція підключена до центрального концентратора або концентратора, який функціонує як багатопортовий повторювач. Кожна станція транслює на всі пристрої, підключені до концентратора. Фізичні топології локальної мережі зазвичай характеризуються як ..... обсяг мережі



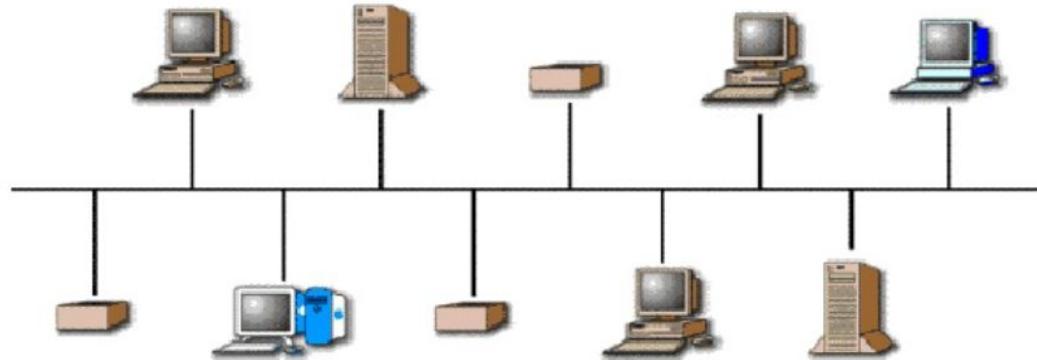
# Методи передачі по локальній мережі

Методи передачі по локальній мережі поділяються на 3 основні категорії:

- Одноадресна передача
- Багатоадресна передача
- Трансляція

# Одноадресна передача

При одноадресній передачі один пакет даних надсилається від джерела до одного адресата в мережі.



# Одноадресний процес

- Джерело звертається до пакета з адресою призначення.
- Пакет відправляється в мережу.
- Мережа доставляє пакет до місця призначення.

# Багатоадресна передача

Під час багатоадресної передачі один пакет даних копіюється та надсилається до певних місць призначення в мережі

## Багатоадресний процес

- Джерело адресує пакет за допомогою групової адреси.
- Пакет відправляється в мережу.
- Мережа копіює пакет.
- Копія доставляється до кожного адресата, який включено в групову адресу.

# Трансляція

Під час багатоадресної передачі один пакет даних копіюється та надсилається до певних місць призначення в мережі

## Процес трансляції

- Джерело адресує пакет широкомовною адресою.
- Пакет відправляється в мережу.
- Мережа копіює пакет.
- Копії пакетів доставляються в усі пункти призначення в мережі.

# Пристрої інфраструктури локальної мережі

Є багато пристроїв, пов'язаних з потоком інформації через локальну мережу. При з'єднанні вони створюють інфраструктуру функціональної локальної мережі.

Ці пристрої включають:

- Повторювачі
- Мости
- Хаби
- Перемикачі
- Маршрутизатори

# Протоколи

- Необхідні спільні дії
  - Комп'ютерна мережа полягає не тільки в обміні байтами
  - величезна система з кількома утилітами та функціями. Наприклад
    - виявлення помилок
    - Шифрування
    - Маршрутизація
    - тощо.
- Для належного спілкування сущності в різних системах повинні розмовляти однією мовою
  - повинні існувати взаємоприйнятні конвенції та правила щодо змісту, часу та основних механізмів
- Ці конвенції та відповідні правила називаються «ПРОТОКОЛАМИ»

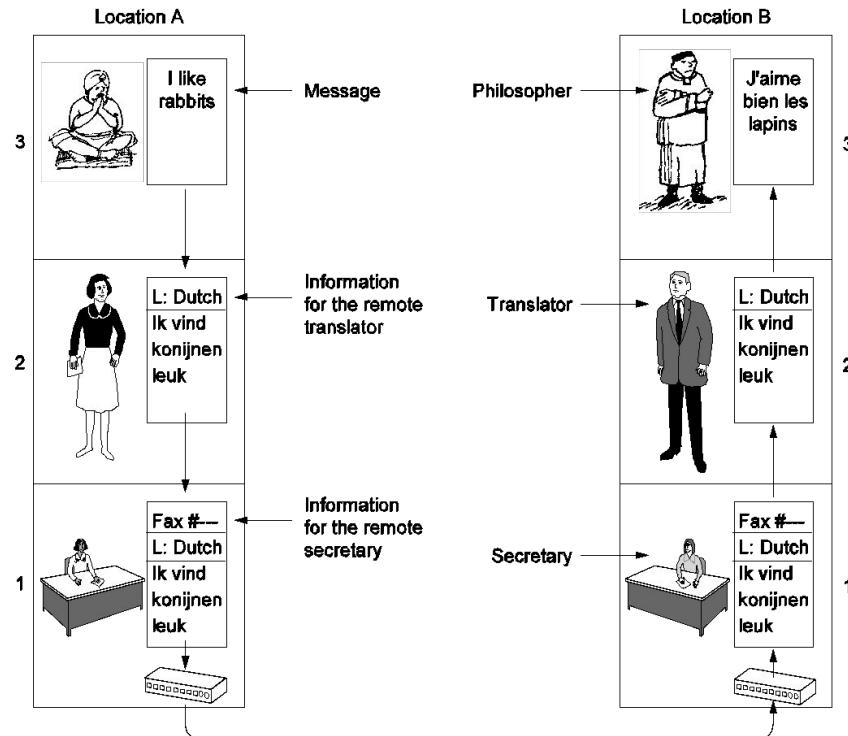
# Архітектура протоколу

- Завдання передачі даних розбито на кілька модулів
  - Чому?
  - Як ці модулі взаємодіють?
- Наприклад, передача файлів може використовувати три модулі
  - Програма для передачі файлів
  - Модуль обслуговування зв'язку
  - Модуль доступу до мережі

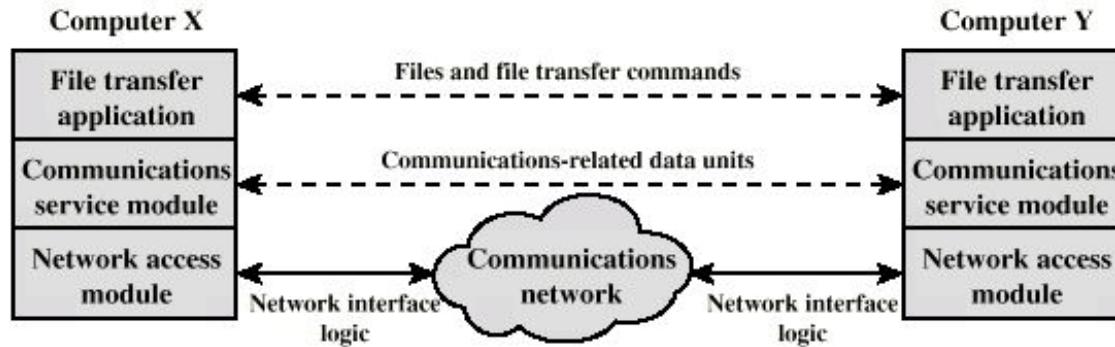
# Приклад реального світу для архітектури протоколу архітектура філософ-перекладач-секретар

## Проблеми:

- однорангові протоколи незалежні один від одного
  - наприклад, секретарі можуть змінити комунік. середній для електронної пошти
  - або перекладачі можуть домовитися про використання іншої спільної мови
- Кожен шар додає заголовок



# Спрощена архітектура передачі файлів



Прикладний рівень передачі файлів: команди, паролі та фактичні файли – дані високого рівня

Комунаційний сервісний модуль: надійна передача цих даних – виявлення помилок, впорядкована доставка пакетів даних тощо.

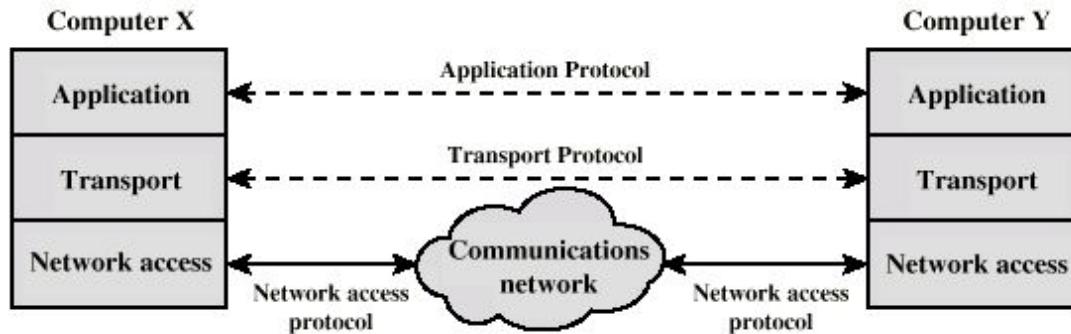
Мережевий модуль: фактична передача даних і робота з мережею – якщо мережа змінюється, це впливає лише на цей модуль, а не на всю систему

# Загальні принципи архітектури протоколу, які ми бачили досі

- Шарова структура
  - Стек протоколів
- Кожен рівень надає послуги верхньому рівню; очікування послуг від нижчого
  - Інтерфейси рівнів повинні бути чітко визначені
- Однорангові об'єкти спілкуються за допомогою власного протоколу
  - однорангові протоколи
  - незалежно від протоколів на інших рівнях
  - якщо один протокол змінюється, це не повинно вплинути на інші протоколи

# Загальна трирівнева модель

- Узагальніть попередній приклад для загальної програми
  - ми можемо мати різні програми (електронна пошта, передача файлів, ...)



- Рівень доступу до мережі
- Транспортний рівень
- Рівень програми

# Рівень доступу до мережі

- Обмін даними між комп'ютером і мережею
- Комп'ютер-відправник надає адресу призначення
  - щоб мережа могла маршрутизувати
- Різні методи комутації та мереж
  - Комутація ланцюга
  - Комутація пакетів
  - локальні мережі
  - тощо.
- Для цього рівня можуть знадобитися спеціальні драйвери та інтерфейсне обладнання залежно від типу використовуваної мережі.
- Але верхні шари не бачать цих деталей
  - власність незалежності

# Транспортний рівень

- Надійний обмін даними
  - щоб переконатися, що всі пакети даних надійшли в тому самому порядку, в якому вони надсилаються
  - Неотримані або отримані помилково пакети передаються повторно
- Незалежно від мережі, що використовується
- Незалежно від застосування

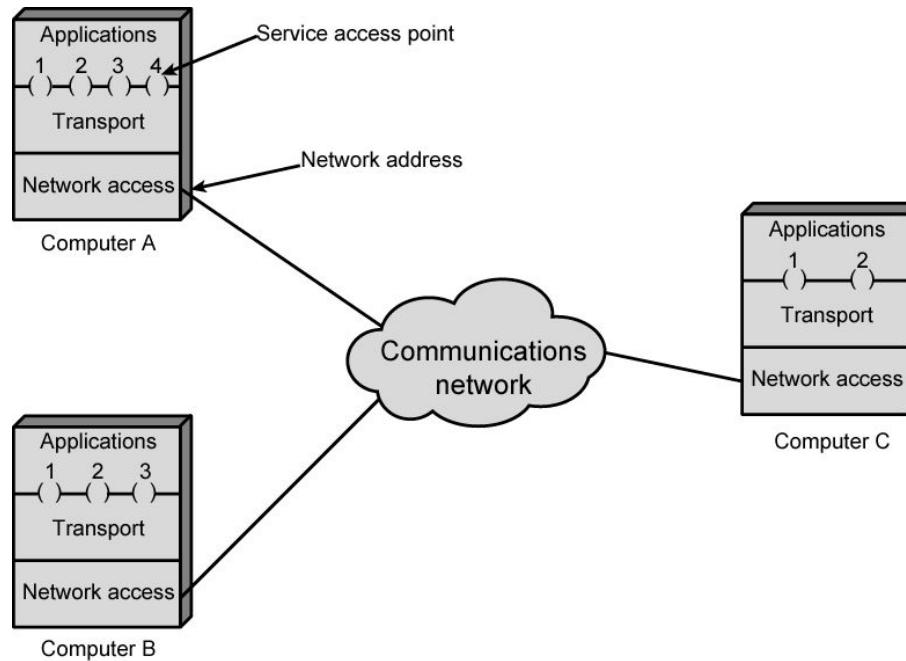
# Рівень програми

- Підтримка різних програм користувача
- напр. електронна пошта, передача файлів

# Адресація вимог

- Потрібні два рівні адресації
- Кожному комп'ютеру потрібна унікальна мережева адреса
- Кожна програма на (багатозадачному) комп'ютері потребує унікальної адреси на комп'ютері
  - Точка доступу до служби або SAP
  - Номер порту в стеку протоколів TCP/IP

# Архітектури протоколів і мережі



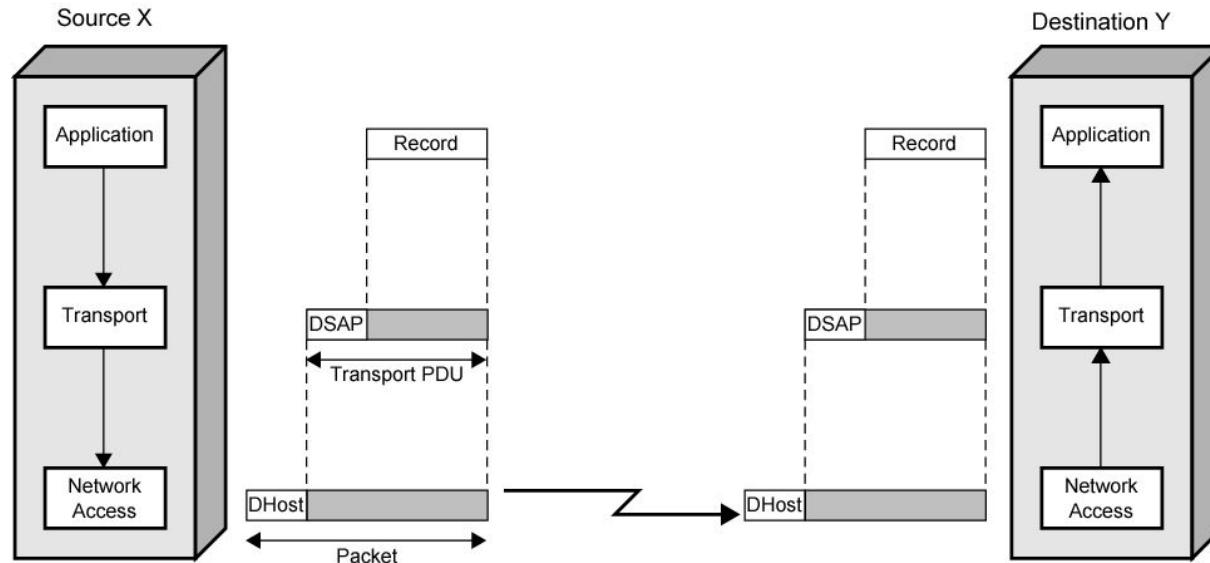
# Блоки даних протоколу (PDU)

- Дані користувача передаються від рівня до рівня
- Керуюча інформація додається/видаляється до/з даних користувача на кожному рівні
  - Заголовок (іноді трейлер)
  - кожен рівень має інший заголовок/кінець
- Дані + заголовок + трейлер = PDU (протокольний блок даних)
  - По суті, це те, що ми називаємо пакетом
  - кожен рівень має інший PDU

# Мережевий PDU

- Додає мережевий заголовок
  - мережева адреса комп'ютера призначення
  - додаткові засоби з мережі (наприклад, рівень пріоритету)

# Робота архітектури протоколу



DSAP = destination service access point  
DHost = destination host

# Стандартні архітектури протоколів

- Загальний набір умовностей
- Нестандартні проти стандартних протоколів
  - Нестандартні: К джерел і L приймачів ведуть до K\*L різних протоколів
  - Якщо використовується загальний протокол, ми проектуємо лише один раз
- Продукти від різних виробників взаємодіють
  - Клієнти не прив'язуються до конкретного постачальника
  - Якщо загальний стандарт не реалізовано в продукті, тоді ринок цього продукту обмежений; клієнтам подобаються стандартні продукти

# Стандартні архітектури протоколів

- Два підходи (стандарт)
  - Еталонна модель OSI
    - ніколи не використовувалась широко
    - але добре відома
  - Набір протоколів TCP/IP
    - Найбільш широко використовується
- Інший підхід (запатентований)
  - Системна мережева архітектура IBM (SNA)

# Еталонна модель OSI

- Взаємозв'язок відкритих систем (OSI)
- Еталонна модель
  - забезпечує загальну основу для стандартизації
  - визначає набір рівнів і послуг, що надаються кожним рівнем
  - Для кожного рівня можна розробити один або декілька протоколів
- Розроблено Міжнародною організацією стандартизації (ISO)
  - також опубліковано ITU-T (Міжнародний союз телекомунікацій)

# Еталонна модель OSI

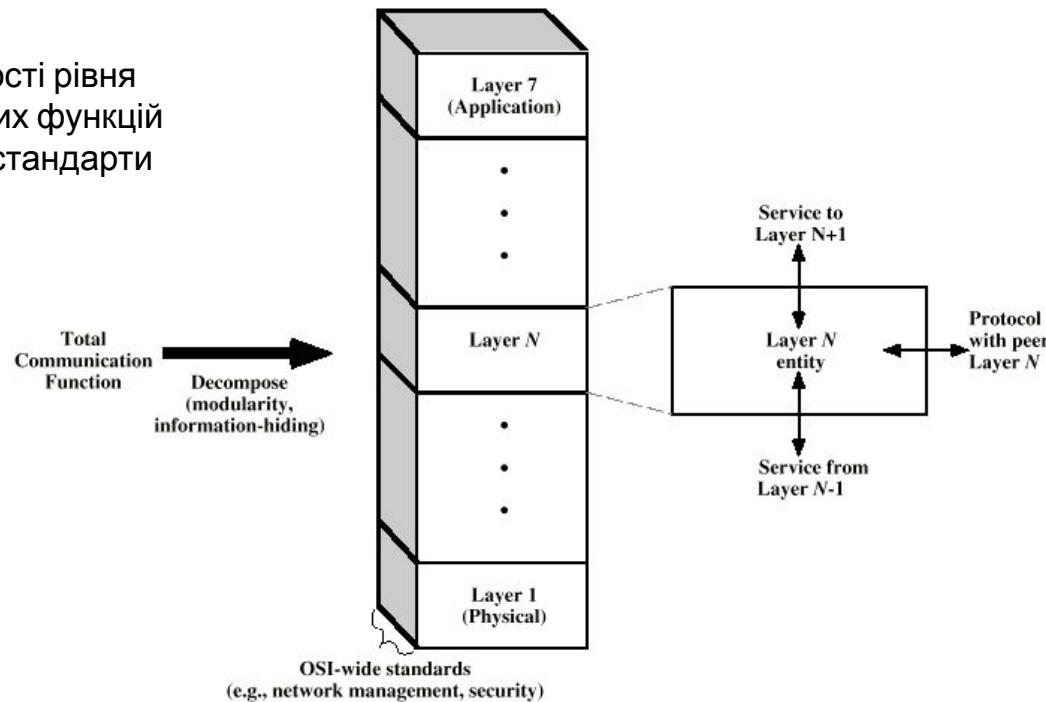
- Багатошарова модель
  - Сім шарів – сім представлено як оптимальна кількість шарів
- Поставлено занадто пізно (опубліковано в 1984 році)!
  - на той час TCP/IP почав ставати стандартом де-факто
- Хоча жоден протокол на основі OSI не зберігся, модель все ще діє (у підручниках)
  - Для канального рівня (який ми побачимо пізніше) протоколи OSI все ще дійсні

# OSI - модель рівня

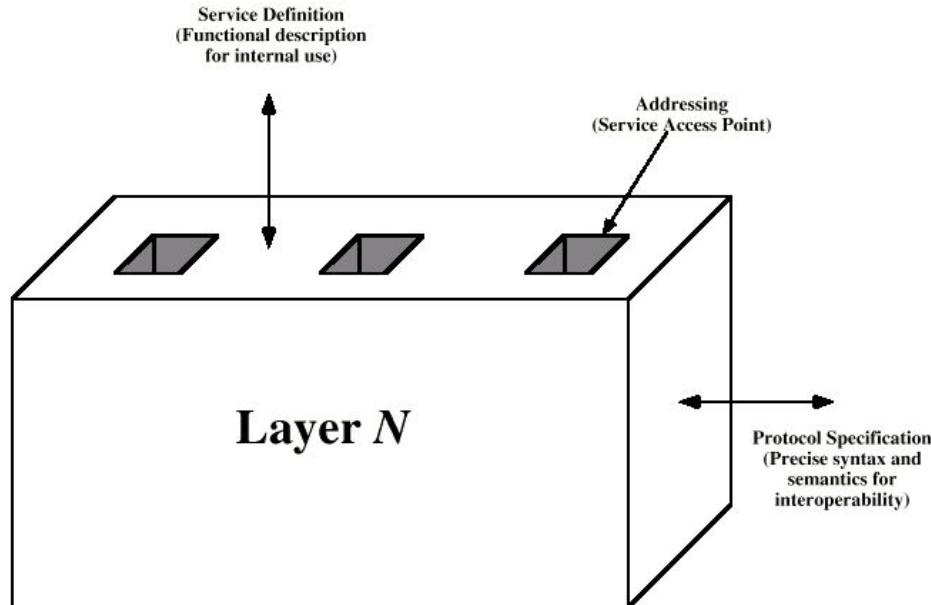
- Кожен рівень виконує підмножину необхідних комунікаційних функцій
- Кожен рівень покладається на наступний нижчий рівень для виконання більш примітивних функцій
- Кожен рівень надає послуги наступному вищому рівню
- Зміни в одному рівні не повинні вимагати змін в інших рівнях

# OSI як основа для стандартизації

функціональні можливості рівня описані ISO; на основі цих функцій можна розробити різні стандарти



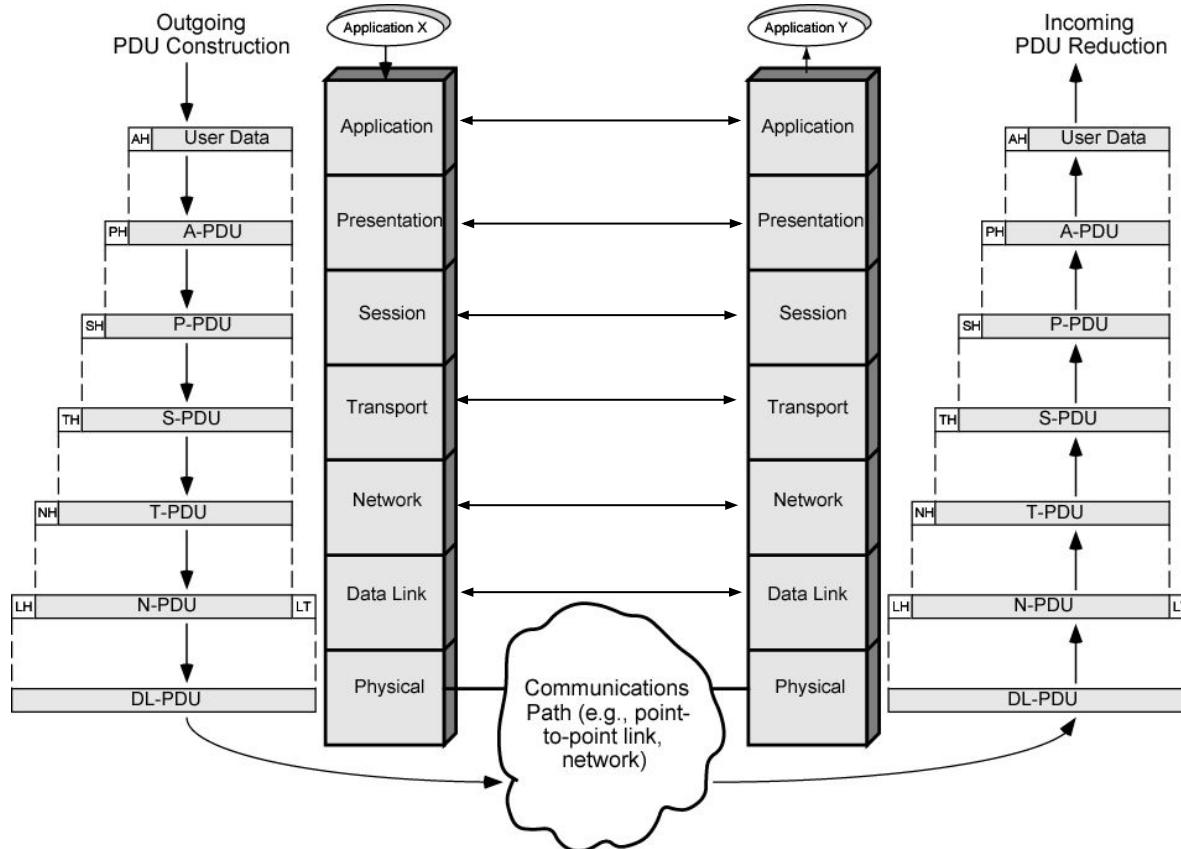
# Спеціальні стандарти рівня



# Елементи стандартизації

- Специфікація протоколу
  - Працює між одним рівнем у двох системах
    - Може включати різні платформи
  - Специфікація протоколу має бути точною
    - Формат одиниць даних
    - Семантика всіх полів
- Визначення послуги
  - Функціональний опис того, що надається наступному верхньому рівню
- Адресація
  - Посилається на SAP

# OSI навколошнє середовище



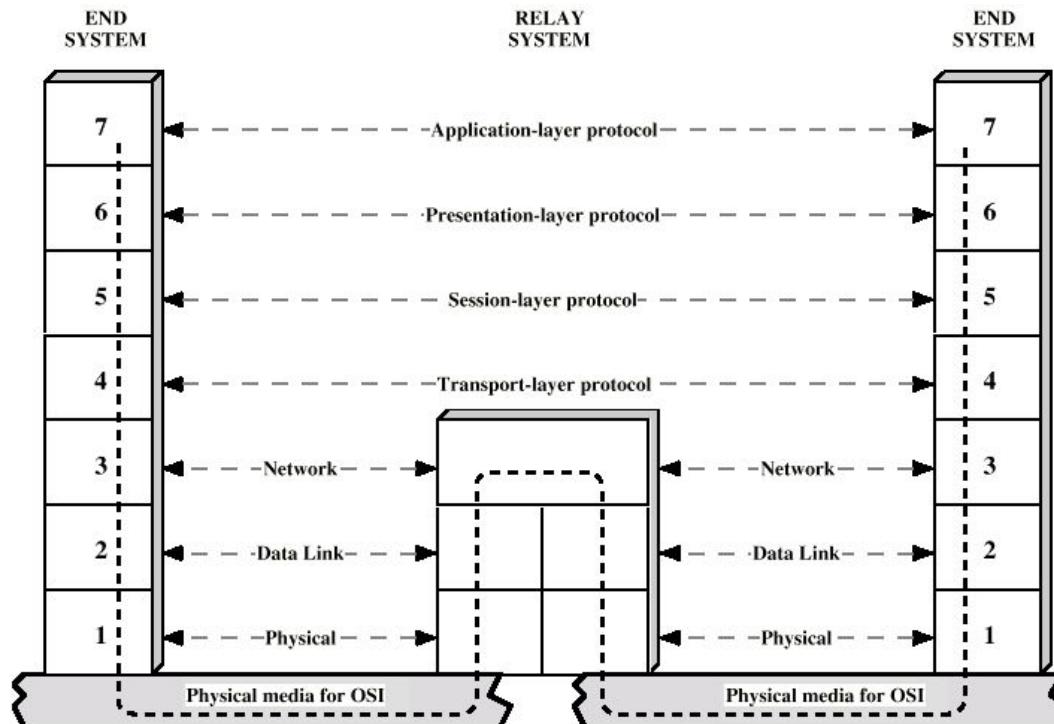
# OSI Рівні

- фізичний
  - Фізичний інтерфейс між пристроями
  - Характеристики
    - Механічні - характеристики інтерфейсу
    - Електричні – рівні напруги для бітів, швидкість передачі, кодування тощо.
- Передача даних
  - Базові послуги: виявлення та контроль помилок, контроль потоку на рівні каналу (точка-точка)
    - Більш високі рівні можуть передбачати передачу без помилок
  - Пізніше до рівня зв'язку даних додається підрівень
    - Підрівень MAC (Medium Access Control).
    - мати справу з мережами мовлення

# OSI Рівні

- Мережа
  - Передача інформації мережею зв'язку
    - проблеми, пов'язані з мережею
  - Вузли мережі (ретранслятори/маршрутизатори) повинні виконувати функції комутації та маршрутизації
  - QoS (якість обслуговування) і контроль перевантаження також розглядаються на цьому рівні
  - Декілька інших проблем з мережею
    - напр. відмінності в адресації, макс. довжина даних тощо.
  - Вищим рівням не потрібно знати базову мережеву технологію
  - Не потрібні прямі посилання

# Використання А Реле/маршрутизатор



# OSI Рівні

- Транспорт
  - Наскрізний обмін даними
  - Послідовно, без втрат, без дублікатів
  - Якщо необхідно, дані верхнього рівня розбиваються на менші блоки
- Сесія
  - Контроль діалогів
    - чия черга говорити?
    - Дисципліна діалогу (повний дуплекс, напівдуплекс)
  - Контрольні точки та відновлення

# OSI Рівні

- Презентація
  - Формати даних
  - Стиснення даних
  - Шифрування
- Застосування
  - Підтримка різних програм

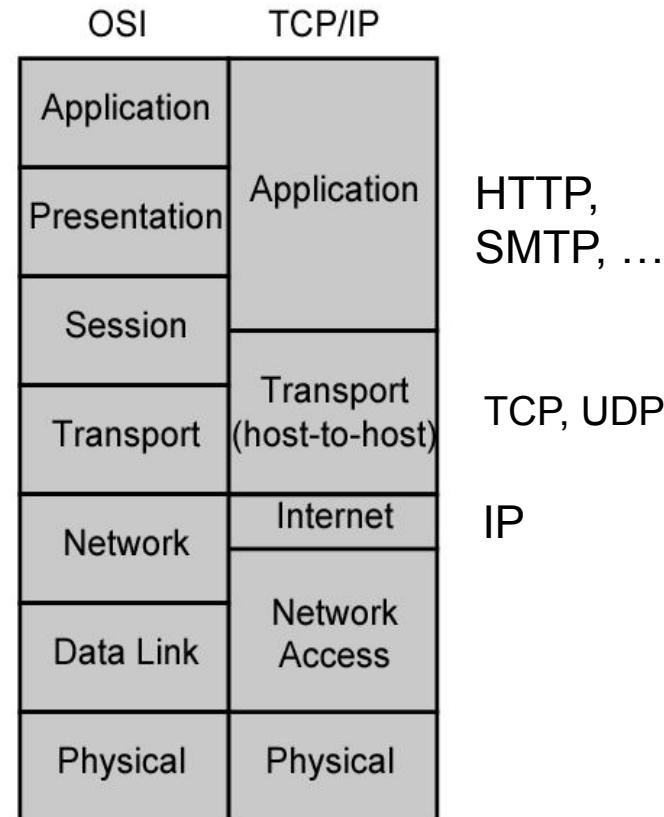
# Набір протоколів TCP/IP

- Найпоширеніша взаємодіюча архітектура мережевого протоколу
- Визначався і широко використовувався до OSI
  - OSI повільно розвивався на ринку
- Фінансується Агентством передових оборонних досліджень США (DARPA) для своєї мережі з комутацією пакетів (ARPANET)
  - Міністерство оборони автоматично створило величезний ринок для TCP/IP
- Використовується Інтернетом і WWW

# Набір протоколів TCP/IP

- TCP/IP не має офіційної структури рівнів
- Але протоколи мають на увазі одне
  - Прикладний рівень
  - Транспортний рівень (від вузла до хосту / від кінця до кінця).
  - Інтернет-рівень
  - Рівень доступу до мережі
  - Фізичний рівень
- Фактично еталонна модель TCP/IP була побудована на його протоколах
  - Ось чому ця еталонна модель призначена лише для набору протоколів TCP/IP
  - і ось чому не так важливо призначати ролі кожному рівню в TCP/IP; розуміння TCP, IP і прикладних протоколів буде достатньо

# OSI проти TCP/IP



# Доступ до мережі та фізичні рівні

- Еталонна модель TCP/IP не надто обговорює ці рівні
  - вузол повинен підключитися до мережі за таким протоколом, щоб він міг надсилати IP-пакети
  - цей протокол не визначений TCP/IP
  - в основному в апаратному забезпеченні
  - добре відомим прикладом є Ethernet

# Інтернет-рівень

- Протокол міжмережевих мереж без підключення «точка-точка» (використовує дейтаграмний підхід)
  - піклується про маршрутизацію в кількох мережах
  - кожен пакет подорожує в мережі незалежно один від одного
    - можуть не прийти (якщо проблема в мережі)
    - вони можуть прибути не в порядку
  - проектне рішення, яке виконує Міністерство оборони, щоб зробити систему більш гнучкою та чутливою до втрати деяких пристрійів підмережі
- Реалізовано в кінцевих системах і маршрутизаторах як Інтернет-протокол (IP)

# Транспортний рівень

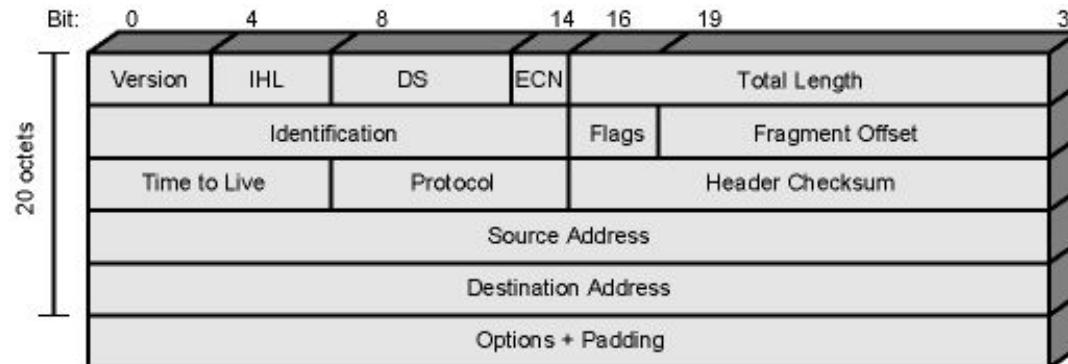
- Наскрізна передача даних
- Протокол керування передачею (TCP)
  - орієнтований на підключення
  - надійна доставка даних
  - замовлення доставки
- Протокол дейтаграм користувача (UDP)
  - послуга без підключення
  - доставка не гарантується
- Чи можете ви навести приклади програм, які використовують TCP і UDP?

# Рівень програми

- Support for user applications
- Окремий модуль дляожної програми
  - напр. HTTP, SMTP, telnet

# IP (Інтернет-протокол)

- Ядро набору протоколів TCP/IP
- Співіснують дві версії
  - v4 – широко використовуваний протокол IP
  - v6 – був стандартизований у 1996 році, але досі не отримав широкого розповсюдження
- IP(v4) заголовок мінімум 20 октетів (160 біт)



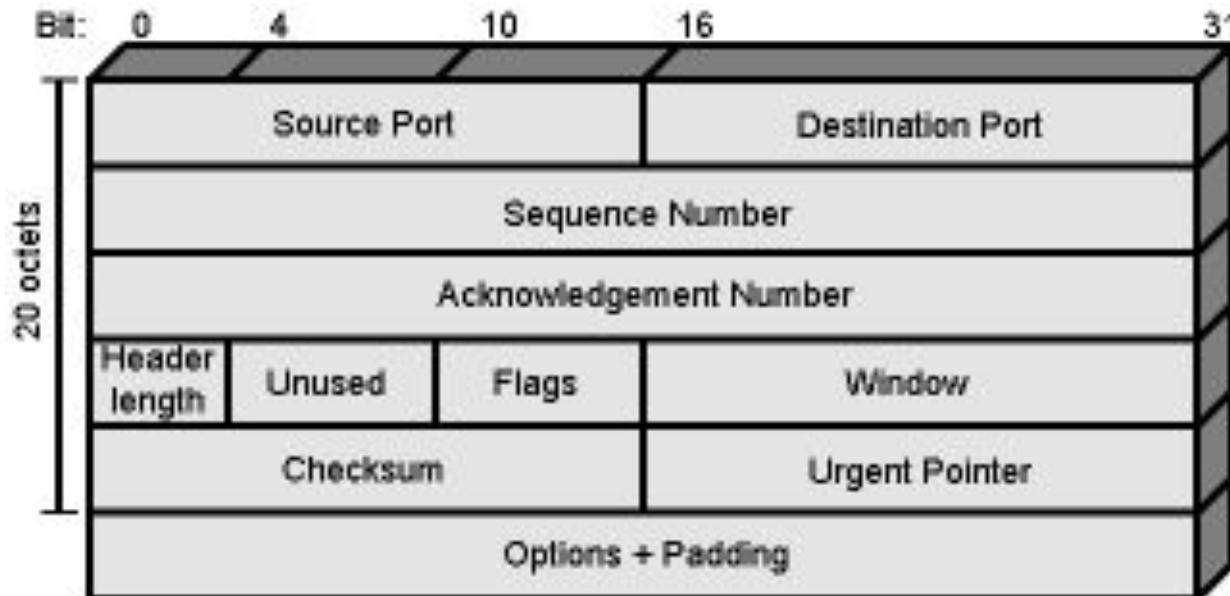
# IPv6

- IPv6
  - Удосконалення IPv4 для сучасних високошвидкісних мереж
  - Підтримка потоків мультимедійних даних
- Але рушійною силою версії 6 було збільшення адресного простору
  - 128-біт порівняно з 32-бітами версії 4
- Не має зворотної сумісності
  - все обладнання та програмне забезпечення повинні бути змінені

# TCP

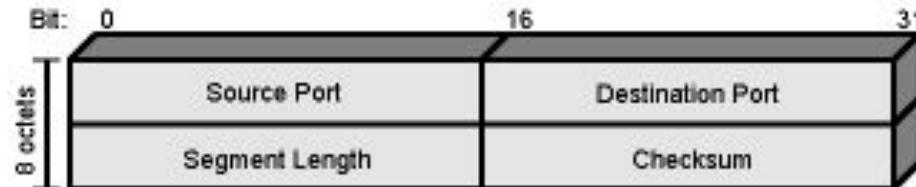
- Протокол керування передачею
  - наскрізний протокол
  - Надійне підключення = забезпечує контроль потоку та помилок
- У термінах TCP з'єднання - це  
*тимчасове об'єднання між суб'єктами в різних системах*
- TCP PDU
  - Називається «сегмент TCP»
  - Включає порт джерела та призначення
    - Визначити відповідних користувачів (додатки)
    - пара портів (разом з IP-адресами) однозначно ідентифікує з'єднання; така ідентифікація необхідна для того, щоб TCP відстежував сегменти між сутностями.

# TCP Заголовок



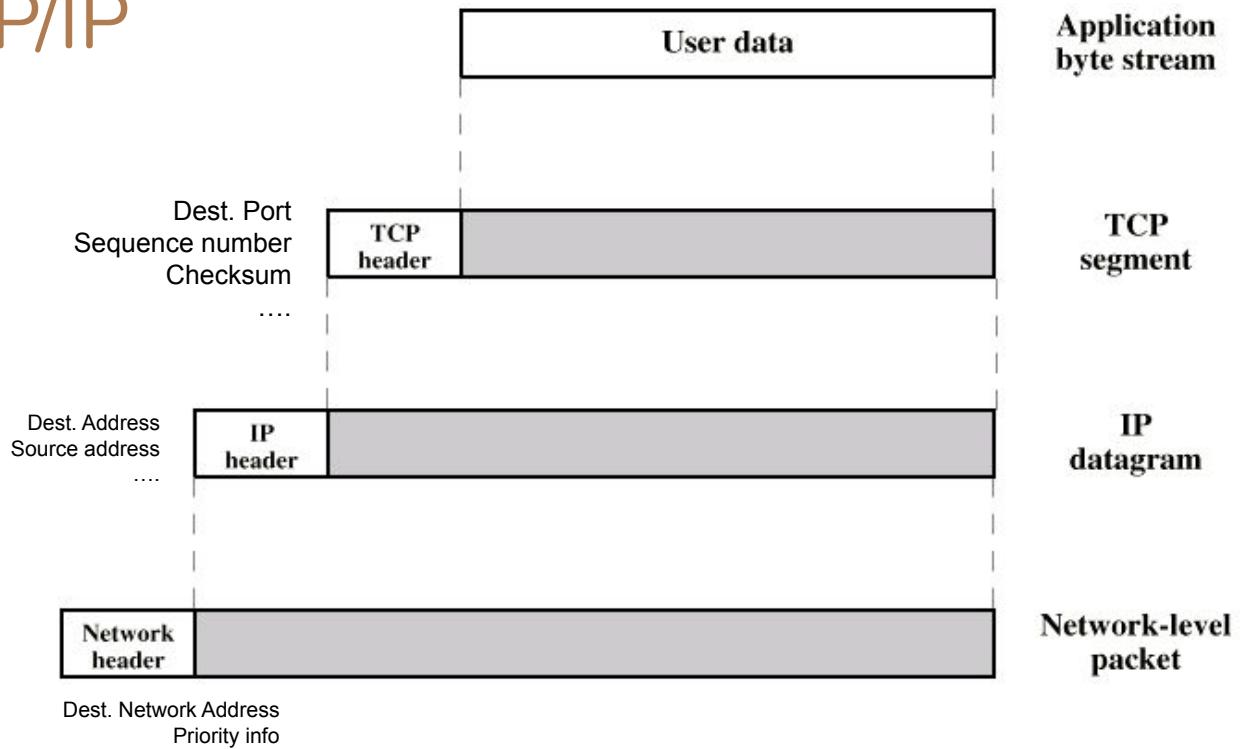
# UDP

- Протокол дейтаграм користувача
- Альтернатива TCP
  - наскрізний протокол
- Доставка не гарантована
- Без збереження послідовності
- Немає захисту від дублювання
- Мінімальні накладні витрати

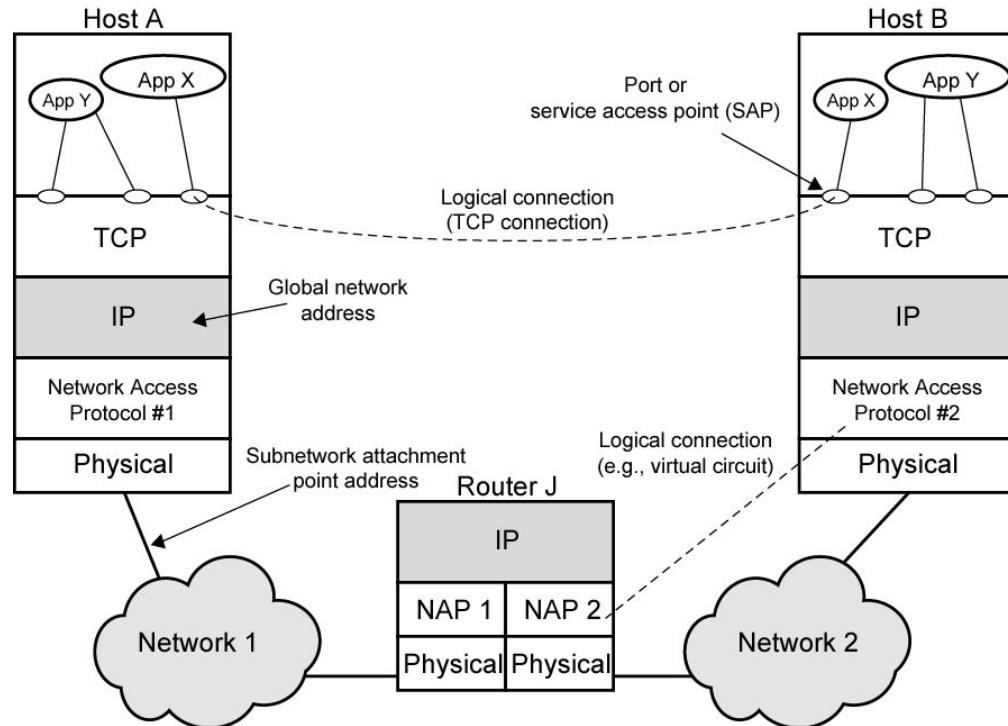


(b) UDP Header

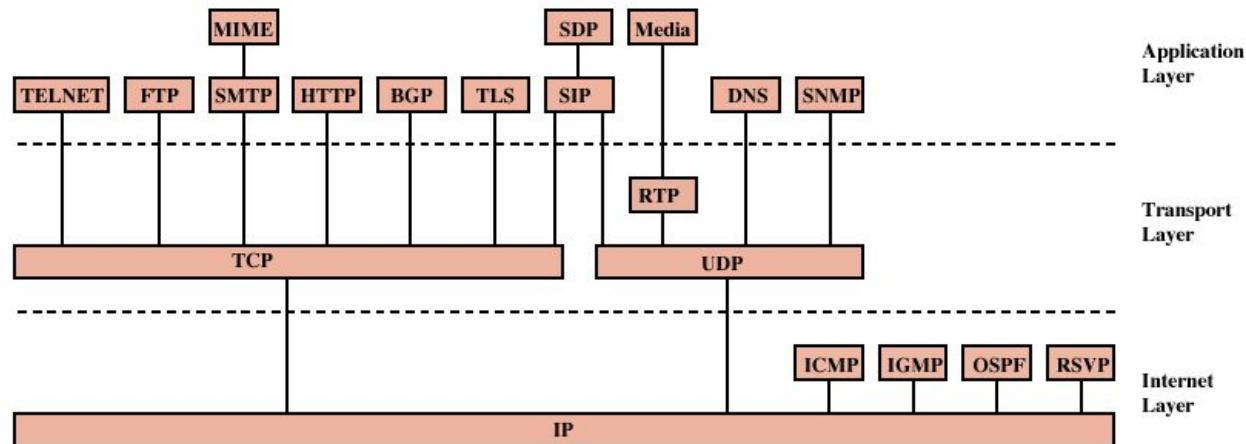
# PDU в TCP/IP



# Операція TCP та IP



# Деякі протоколи в TCP/IP Suite



BGP = Border Gateway Protocol  
DNS = Domain Name System  
FTP = File Transfer Protocol  
HTTP = Hypertext Transfer Protocol  
ICMP = Internet Control Message Protocol  
IGMP = Internet Group Management Protocol  
IP = Internet Protocol  
MIME = Multi-Purpose Internet Mail Extension  
OSPF = Open Shortest Path First

RSVP = Resource ReSerVation Protocol  
RTP = Real-Time Transport Protocol  
SDP = Session Description Protocol  
SIP = Session Initiation Protocol  
SMTP = Simple Mail Transfer Protocol  
SNMP = Simple Network Management Protocol  
TCP = Transmission Control Protocol  
TLS = Transport Layer Security  
UDP = User Datagram Protocol

# Міжмережеве з'єднання

- Взаємопов'язаний набір мереж
  - Може виглядати як велика мережа
- Кожна складова мережа є підмережею
- Вся конфігурація називається Інтернетом
  - не Інтернет
    - концептуально те саме, але під «Інтернетом» ми не маємо на увазі конкретну мережу
    - Інтернет є найважливішим прикладом мережі типу “Інтернет”

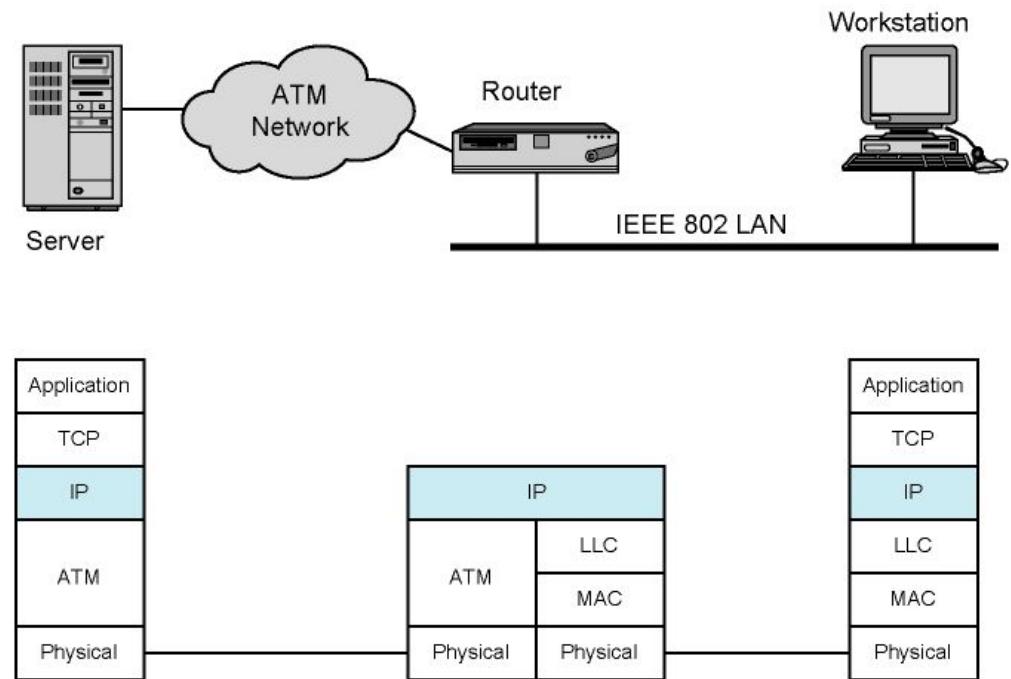
# Міжмережеве з'єднання Пристрой

- Кожна підмережа підтримує зв'язок між пристроями, підключеними до цієї підмережі
  - Кінцеві системи (ESs)
- Підмережі, з'єднані проміжними системами (ISs)
  - На практиці ISs— це маршрутизатори, які використовуються для ретрансляції та маршрутизації пакетів між різними підмережами
  - Якщо підмережі використовують різні протоколи доступу до мережі, маршрутизатор повинен підтримувати всі протоколи
  - Згідно з термінологією OSI, маршрутизатор працює на рівні 3 (мережевий рівень).

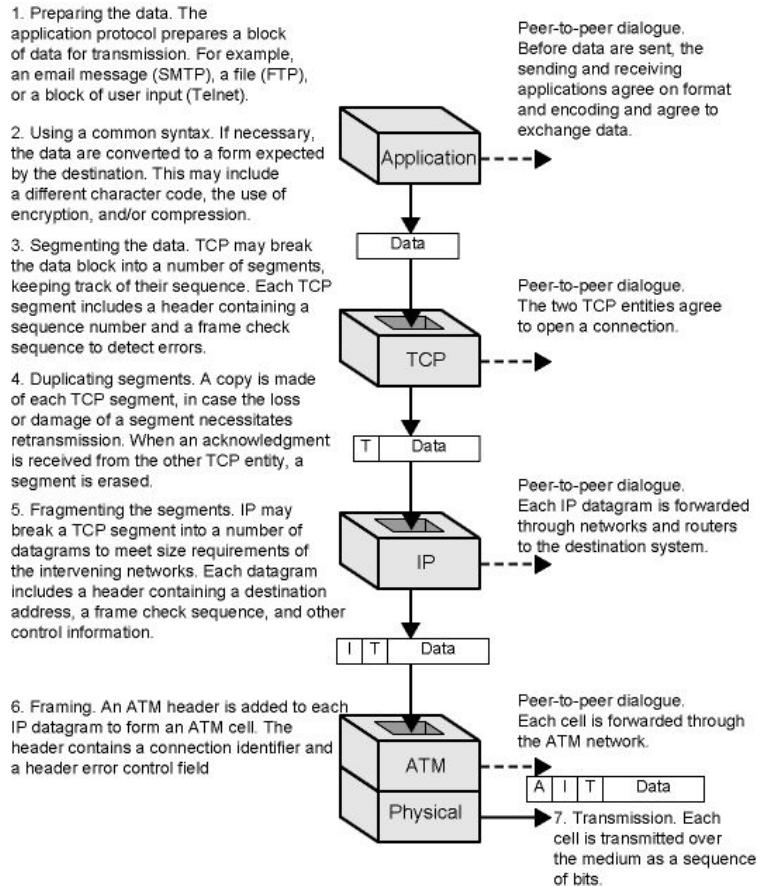
# Маршрутизатори

- З'єднуйте різноманітні підмережі без будь-яких змін в архітектурі підмереж
- Необхідно врахувати відмінності між мережами, наприклад
  - Схеми адресації
    - можливо, потрібно буде перекласти мережеві адреси
  - Максимальний розмір пакетів
    - якщо дві підмережі мають різні обмеження для макс. розмірів пакетів, то маршрутизатору може знадобитися фрагментувати/повторно зібрати пакети
- Ми бачили, що підмережі можуть мати різний доступ до мережі та фізичні рівні, але вони повинні використовувати один (між)мережевий протокол, реалізований у всіх кінцевих системах і маршрутизаторах
  - Найважливішим міжмережевим протоколом є протокол IP

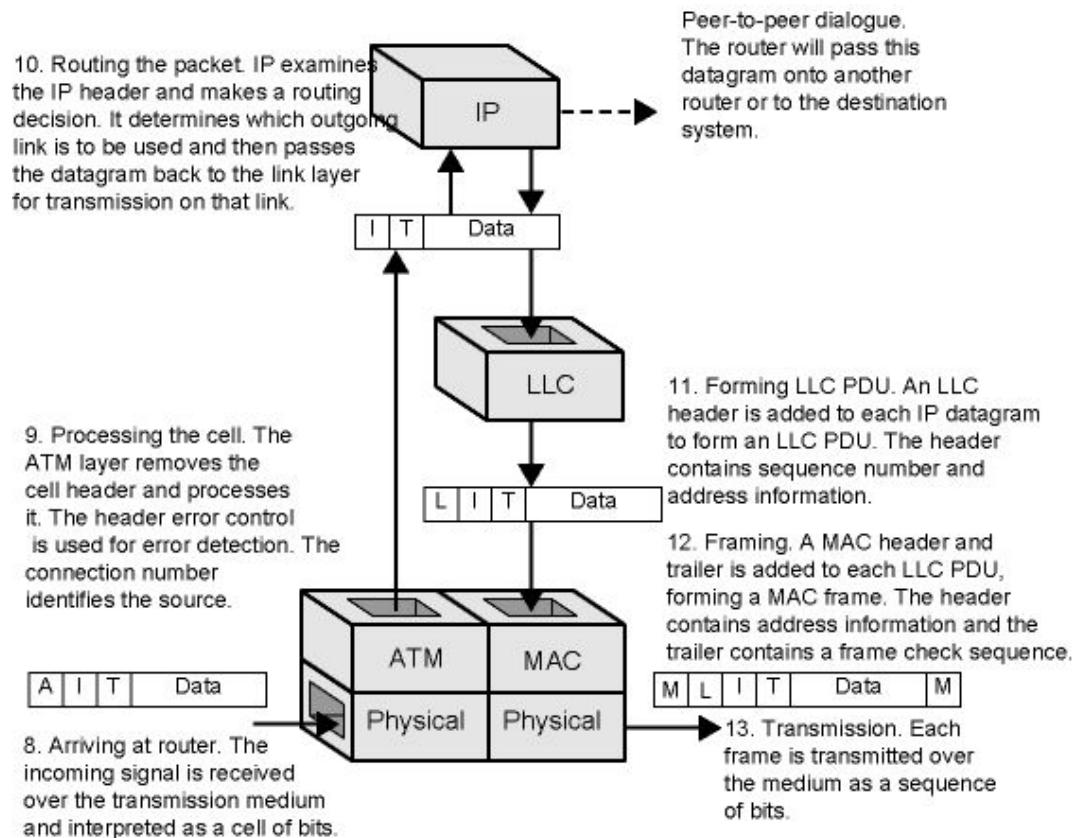
# Конфігурація для TCP/IP Приклад



# Дія Відправника



# Дія маршрутизатора



# Дія приймача

20. Delivering the data. The application performs any needed transformations, including decompression and decryption, and directs the data to the appropriate file or other destination.

19. Reassembling user data. If TCP has broken the user data into multiple segments, these are reassembled and the block is passed up to the application.

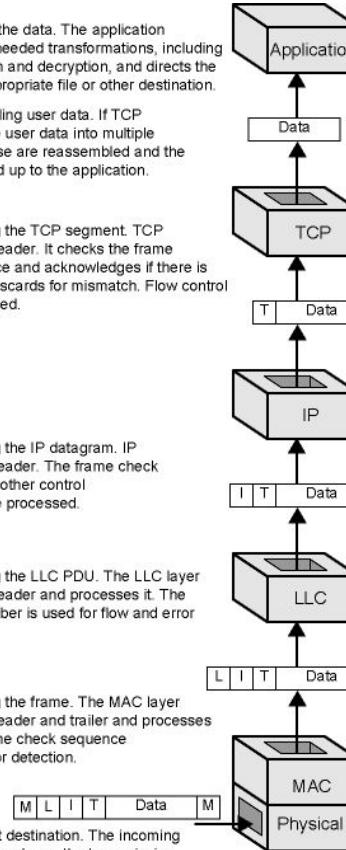
18. Processing the TCP segment. TCP removes the header. It checks the frame check sequence and acknowledges if there is a match and discards for mismatch. Flow control is also performed.

17. Processing the IP datagram. IP removes the header. The frame check sequence and other control information are processed.

16. Processing the LLC PDU. The LLC layer removes the header and processes it. The sequence number is used for flow and error control.

15. Processing the frame. The MAC layer removes the header and trailer and processes them. The frame check sequence is used for error detection.

14. Arriving at destination. The incoming signal is received over the transmission medium and interpreted as a frame of bits.



# Стандарти

- Необхідно для забезпечення сумісності між обладнанням
- Переваги
  - Забезпечує великий ринок обладнання та програмного забезпечення
  - Дозволяє продуктам від різних постачальників спілкуватися
- Недолік
  - Технологія заморожування (???)

# Організації стандартів у мережах

- Інтернет-суспільство
- ISO (Міжнародна організація стандартизації)
  - більш формальний
  - NGO, але більшість членів є представниками урядів
- ITU-T (раніше CCITT)
  - Міжнародний союз телекомунікацій
  - UN агенція
  - урядовий

# Інтернет-суспільство (ISOC)

- Розвиток і стандартизація Інтернету
- З суборганізації
  - IAB (Рада з архітектури Інтернету)
    - загальна архітектура Інтернету
  - IETF (Інженерна робоча група Інтернету)
    - розробка протоколів
  - IESG (Інженерна керівна група Інтернету)
    - стежить за зусиллями IETF зі стандартизації

# Організація IETF

- Згруповані в області
  - напр. програми, безпека, маршрутизація тощо.
  - у кожній області є регіональний директор, який також є членом IESG
- Кожна область має кілька робочих груп
  - робочі групи фактично роблять внесок у стандарти/протоколи тощо.
- Добровільна участь у робочих групах IETF
- Докладніше див
  - [www.ietf.org](http://www.ietf.org) або
  - RFC 3160 - The Tao of IETF - A Novice's Guide to the Internet Engineering Task Force

# Інтернет-чернетки та RFC

- Інтернет чернетка
  - Чорнові та тимчасові документи
  - закінчується через 6 місяців, якщо IESG не схвалить його як RFC
  - можна повторно подати
  - опубліковано онлайн
  - коментарі вітаються
- RFC (Запит на коментарі)
  - остаточна версія
  - може застаріти попередній RFC на ту саму тему
  - насправді RFC може бути документом будь-якого типу
    - не обов'язково стандарт
    - Найкращі поточні практики, експериментальні, інформаційні RFC
    - RFC 1 квітня ([http://en.wikipedia.org/wiki/April\\_1\\_RFC](http://en.wikipedia.org/wiki/April_1_RFC))
      - Мій улюблений IP через Avian Carriers (RFC 1149)

Network Working Group  
Request for Comments: 1149

D. Waitzman  
BBN STC  
1 April 1990

## A Standard for the Transmission of IP Datagrams on Avian Carriers

### Status of this Memo

This memo describes an experimental method for the encapsulation of IP datagrams in avian carriers. This specification is primarily useful in Metropolitan Area Networks. This is an experimental, not recommended standard. Distribution of this memo is unlimited.

### Overview and Rational

Avian carriers can provide high delay, low throughput, and low altitude service. The connection topology is limited to a single point-to-point path for each carrier, used with standard carriers, but many carriers can be used without significant interference with each other, outside of early spring. This is because of the 3D ether space available to the carriers, in contrast to the 1D ether used by IEEE802.3. The carriers have an intrinsic collision avoidance system, which increases availability. Unlike some network technologies, such as packet radio, communication is not limited to line-of-sight distance. Connection oriented service is available in some cities, usually based upon a central hub topology.

### Frame Format

The IP datagram is printed, on a small scroll of paper, in hexadecimal, with each octet separated by whitestuff and blackstuff. The scroll of paper is wrapped around one leg of the avian carrier. A band of duct tape is used to secure the datagram's edges. The bandwidth is limited to the leg length. The MTU is variable, and

# Трек стандартів Інтернету

- Ці кроки передбачають збільшення обсягу перевірки та тестування
- Крок 1: Чернетка в Інтернеті
- Крок 2: Запропонований стандарт
  - Інтернет-проект, схвалений IESG як RFC
  - має залишитися принаймні шість місяців для просування
- Крок 3: Проект стандарту
  - щонайменше дві незалежні та сумісні реалізації
  - має залишатися не менше 4 місяців
- Крок 4: Інтернет-стандарт
  - Значний досвід роботи
    - Ключова відмінність між ISOC та іншими організаціями стандартизації
  - Потрібен консенсус

# Орган з присвоєння номерів Інтернету (IANA)

- Організація ISOC, відповідальна за всі «унікальні номери» в Інтернеті
  - включаючи IP-адреси
- Майже всі протоколи працюють з числовими параметрами
  - напр. номери портів, коди помилок, коди стану, типи повідомлень, параметри тощо.
  - значення всіх числових кодів здебільшого визначені в RFC, але призначення номерів формалізовано IANA

# Брандмауер

Firewalld — це інструмент керування брандмауером для операційних систем Linux. Він надає функції брандмауера, діючи як інтерфейс для інфраструктури netfilter ядра Linux через утиліту простору користувача nftables (до v0.6.0 iptables backend), діючи як альтернатива програмі командного рядка nft. Назва firewalld дотримується конвенції Unix щодо іменування системних демонів шляхом додавання літери «d»

firewalld підтримує мережі IPv4 і IPv6 і може керувати окремими зонами брандмауера з різним ступенем довіри, як визначено в профілях зон. Адміністратори можуть налаштовувати Network Manager на автоматичне перемикання профілів зон на основі відомих Wi-Fi (бездротових) і Ethernet (дротових) мереж, але firewalld не може зробити це самостійно.

Служби та програми можуть використовувати інтерфейс D-Bus для запитів і налаштування брандмауера. firewalld підтримує тимчасові правила, тобто кількість з'єднань (або «звернень») до служби може бути глобально обмежена. Немає підтримки підрахунку звернень і подальшого відхилення підключення для кожної IP-адреси джерела; поширенна техніка, яка використовується для обмеження впливу грубого злому та розподілених атак типу «відмова в обслуговуванні».

Синтаксис команд firewalld подібний до інших інтерфейсів iptables, таких як Uncomplicated Firewall (ufw), але більш докладний. Інтерфейс командного рядка дозволяє керувати наборами правил брандмауера для протоколу, портів, джерела та призначення; або попередньо визначені послуги за назвою.



# Встановіть Firewalld

Стандартною системою брандмауера для Ubuntu є ufw, але за бажанням ви можете встановити та використовувати Firewalld. Встановіть Firewalld на Ubuntu 18.04 / Ubuntu 16.04, виконавши команди:

```
sudo apt-get install firewalld
```

За замовчуванням служба має бути запущена, якщо не працює, запустіть і ввімкніть її запуск під час завантаження:

```
sudo systemctl enable firewalld  
sudo systemctl start firewalld
```

# Встановіть Firewalld

Переконайтесь, що служба працює:

```
$ sudo firewall-cmd --state  
running
```

Якщо у вас увімкнено ufw, вимкніть його, щоб зробити firewalld стандартним брандмауером

```
sudo ufw disable
```

# Використовуйте Firewalld

Тепер, коли пакет встановлено та службу firewalld запущено, давайте розглянемо кілька прикладів використання

Нижче наведено приклади базового використання брандмауера.

## 1. Список усіх налаштованих

### правил брандмауера

`ssh` та `dhcipv6-client` служби ввімкнено за замовчуванням під час запуску служба брандмауера.

```
# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcipv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

# Використовуйте Firewalld

2. Отримайте список усіх служб, які можна ввімкнути за допомогою імені

```
sudo firewall-cmd --get-services
```

3. Увімкніть службу http

Параметр --permanent означає збереження правил проти перезавантаження сервера.

```
sudo firewall-cmd --add-service=http --permanent
```

# Використовуйте Firewalld

4. Увімкніть як http, так і https в одному рядку

```
sudo firewall-cmd --permanent --add-service={http,https} --permanent
```

5. Увімкніть порт TCP 7070

```
sudo firewall-cmd --add-port=7070/tcp --permanent
```

# Використовуйте Firewalld

6. Увімкніть порт UDP 514

```
sudo firewall-cmd --add-port=514/udp --permanent
```

7. Створіть нову зону

```
sudo firewall-cmd --new-zone=myzone --permanent
```

8. Увімкніть послугу в певній зоні

```
sudo firewall-cmd --zone=myzone --add-port=4567/tcp --permanent
```

9. Встановіть зону за замовчуванням

```
sudo firewall-cmd --set-default-zone=public --permanent
```

# Використовуйте Firewalld

## 10. Додати інтерфейс до зони

```
sudo firewall-cmd --get-zone-of-interface=eth0 --permanent  
sudo firewall-cmd --zone=<zone> --add-interface=eth0 --permanent
```

## 11. Дозволити доступ до порту з певної підмережі/IP

```
$ sudo firewall-cmd --add-rich-rule 'rule family="ipv4" service name="ssh" \  
source address="192.168.0.12/32" accept' --permanent  
$ sudo firewall-cmd --add-rich-rule 'rule family="ipv4" service name="ssh" \  
source address="10.1.1.0/24" accept' --permanent
```

# Використовуйте Firewalld

## 12. List rich rules

```
sudo firewall-cmd --list-rich-rules
```

## 13. Налаштuvати переадресацію портів

```
# Enable masquerading
$ sudo firewall-cmd --add-masquerade --permanent

# Port forward to a different port within same server ( 22 > 2022)
$ sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toport=2022 --permanent

# Port forward to same port on a different server (local:22 > 192.168.2.10:22)
$ sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toaddr=192.168.2.10 --permanent

# Port forward to different port on a different server (local:7071 > 10.50.142.37:9071)
$ sudo firewall-cmd --add-forward-port=port=7071:proto=tcp:toport=9071:toaddr=10.50.142.37 --permanent
```

# Використовуйте Firewalld

## 14. Видалення порту/служби

Замінити `--add` на `--remove`

# Питання та відповіді