Лекція №5 КМ

Модель TCP/IP. Стек протоколів TCP/IP. Протоколи прикладного рівня (DNS, HTTP та HTTPS, FTP, SMTP, POP3, IMAP4, Telnet та SSH, SNMP). Протоколи транспортного рівня (TCP, UDP). Порти транспортного рівня. Протоколи міжмережевого рівня (IP, DHCP, ICMP, ARP, RARP).

ПЛАН

1. Модель TCP/IP.
2. Стек протоколів TCP/IP.
3. Протоколи прикладного рівня (DNS, HTTP та HTTPS, FTP, SMTP, POP3, IMAP4, Telnet та SSH, SNMP).
4. Протоколи транспортного рівня (TCP, UDP).
5. Порти транспортного рівня.
6. Протоколи міжмережевого рівня (IP, DHCP, ICMP, ARP, RARP).
7. Вразливості найпоширеніших протоколів стеку TCP/IP мереж.

Література

1. Comer, Douglas E. "Essentials of Computer Networks." Chapman and Hall/CRC, 2nd edition, 2020.
2. Behrouz A. Forouzan, "Data Communications And Networking", 5th Edition.
3. James F. Kurose and Keith W. Ross "Computer Networking: A Top-Down Approach" .
4. W. Richard Stevens "TCP/IP Illustrated, Volume 1: The Protocols" .
5. Postel, J. "Transmission Control Protocol," RFC 793, September 1981.
6. Allman, M., Paxson, V., Stevens, W. "TCP Congestion Control," RFC 2581, April 1999.
7. Ramakrishnan, K. K., Floyd, S. "A Proposal to add Explicit Congestion Notification (ECN) to IP," RFC 3168, September 2001.
8. Jacobson, V., Braden, R., Borman, D. "TCP Extensions for High Performance," RFC 1323, May 1992.
9. Kent, S., Atkinson, R. "Security Architecture for the Internet Protocol," RFC 2401, November 1998.
10. "Understanding TCP/IP," Cisco Systems Inc., Networking Academy Program, 3rd Edition, 2005.
11. "The Illustrated Network," Walter Goralski, 2009, Elsevier Inc.
12. Mauro, Douglas, Kevin Schmidt. "Essential SNMP." O'Reilly Media, 2005.
13. Case, J., M. Fedor, M. Schoffstall, and J. Davin. "A Simple Network Management Protocol (SNMP)." RFC 1157, 1990.
14. SNMP Version 3 Working Group. "Simple Network Management Protocol Version 3 (SNMPv3)." RFC 3411, 2002.
15. SNMP.org. "SNMPv3 Information."
16. "Pro DNS and BIND 10" by Ron Aitchison
17. "DNS and BIND" by Cricket Liu and Paul Albitz

1. Модель TCP/IP.

Модель TCP/IP (Transmission Control Protocol/Internet Protocol) є фундаментальною основою для створення комп'ютерних мереж. TCP/IP є основними протоколами Інтернету. Ця модель визначає, як дані передаються мережами, забезпечуючи надійний зв'язок між пристроями. TCP/IP зазвичай працює з обома типами IP-адрес, тобто IPv4 та IPv6.

Модель TCP/IP була спроектована і розроблена Міністерством оборони США в 1960-х роках і базується на стандартних протоколах. Модель TCP/IP була розроблена одночасно зі створенням мережі ARPANET, яка згодом стала основою сучасного Інтернету. Модель TCP/IP була розроблена для вирішення практичних мережевих задач того часу. Модель TCP/IP розроблена так, щоб не залежати від фізичних носіїв передачі даних і забезпечувати передачу даних.

Важливість TCP/IP.

TCP/IP визначає спосіб обміну даними через Інтернет, забезпечуючи наскрізний зв'язок, який визначає, як вони мають бути розбиті на пакети, адресовані, передані, маршрутизовані та отримані в пункті призначення. TCP/IP не потребує централізованого керування та розроблений для забезпечення надійності мереж із можливістю автоматичного відновлення після збою будь-якого пристрою в мережі. TCP/IP є непатентованим і, як наслідок, не контролюється жодною окремою компанією. Таким чином, набір IP можна легко змінити. Він сумісний з усіма операційними системами (ОС), тому може спілкуватися з будь-якою іншою системою. Набір IP також сумісний з усіма типами комп'ютерного обладнання та мереж.

Розглянемо детальніше модель TCP/IP.

Що робить і як працює модель TCP/IP?

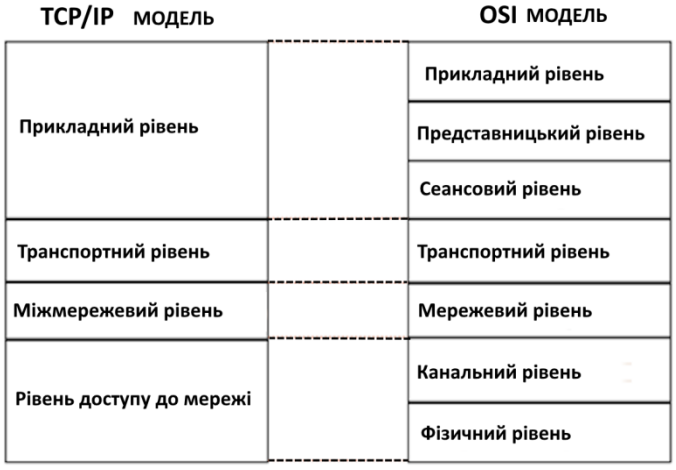
Основна робота TCP/IP полягає в передачі даних комп'ютера з одного мережевого пристрою на інший. Головна умова цього процесу - передати дані достовірними і точними, щоб одержувач отримував ту ж інформацію, яка відправляється відправником. 4-рівнева модель TCP/IP розділяє свої дані на пакети та об'єднує їх на пункті призначення. Модель TCP/IP використовується в Інтернет з широким спектром мережевих технологій та фізичних середовищ. Модель TCP/IP забезпечує гнучкість адаптації до різних фізичних реалізацій.

Різниця між TCP та IP

Ознака	TCP (протокол керування передачею)	IP (інтернет-протокол)
Мета	Забезпечує надійну, впорядковану та перевірену на помилки доставку даних між програмами.	Забезпечує адресацію та маршрутизацію пакетів між мережами.
Тип	Орієнтований на з'єднання	Не орієнтований на з'єднання
Функція	Керує передачею даних між пристроями, забезпечуючи цілісність і порядок даних.	Маршрутизує пакети даних від джерела до місця призначення на основі IP-адрес.
Обробка помилок	Має механізми для перевірки та відновлення помилок.	IP не обробляє помилки; покладається на протоколи верхнього рівня, такі як TCP.
Управління потоком	Має механізми регулювання потоку.	Не має механізмів регулювання потоку.
Керування перевантаженням	Керує перевантаженням мережі.	Не керує перевантаженням мережі.
Сегментація даних	Розбиває дані на менші пакети та збирає їх знову в місці призначення.	Розбиває дані на пакети, але не виконує їх повторне складання.
Розмір заголовка	Більший, 20-60 байт	Менший, зазвичай 20 байт
Надійність	Забезпечує надійну передачу даних	Не гарантує надійність доставки даних.
Підтвердження передачі	Так, підтверджує отримання пакетів даних.	Не підтверджує отримання пакетів даних.

Рівні моделі TCP/IP.

Діаграмне порівняння моделей **TCP/IP і OSI** виглядає так(таблиця 1):



Таблиця 1. Діаграмне порівняння моделей **TCP/IP і OSI** .

Модель TCP/IP має 4 рівні:

1. **Рівень доступу до мережі**: обробляє фізичну передачу даних через мережу.
2. **Міжмережевий рівень**: керує маршрутизацією пакетів даних по мережі.
3. **Транспортний рівень**: забезпечує надійну передачу даних між пристроями.

4. **Прикладний рівень**: надає протоколи для конкретних служб передачі даних на рівні від процесу до процесу.

Переваги TCP/IP моделі комп'ютерної мережі над OSI моделлю:

1. **Простота.** TCP/IP модель простіша і більш легка для розуміння, ніж модель OSI.
2. **Практичність і використання в реальному світі.** Модель TCP/IP практичніша, більш широко використовувана в сучасних мережах в порівнянні з більш теоретичною OSI моделлю.
3. **Надійність.** TCP/IP модель розроблена щоб бути стійкою і працювати в різному оточенні комп'ютерних мереж, що робить її придатною для динамічної природи мереж.
4. **Сумісність.** TCP/IP підтримує широкий діапазон протоколів і технологій, які полегшують сумісність різних систем і мереж.
5. **Широка підтримка і документація.** TCP/IP є основою мережі Інтернет, має широку підтримку громадськості, документацію, ресурси, полегшує пошук несправностей мереж та їх розвиток.
6. **Гнучкість протоколів.** TCP/IP забезпечує додавання нових протоколів без руйнування вже працюючих протоколів, що сприяє інноваціям і адаптованості до мережевих технологій.
7. **Наскрізний зв'язок .** TCP/IP модель підтримує наскрізний зв'язок(end-to-end communication), який спрощує конструювання і впровадження додатків і сервісів, пов'язаних з підключенням до мережі.

Ці переваги сприяють широкому використанню TCP/IP моделі в комп'ютерних мережах, особливо в інтернет-додатках.

Нижче наведено основні відмінності між моделлю OSI та моделлю TCP/IP.

Різниця між моделлю OSI та моделлю TCP/IP.

Модель OSI (Open System Interconnection)	**Модель TCP/IP (Transmission Control Protocol / Internet Protocol)**
1. OSI — це незалежний від протоколу загальний стандарт, що працює як шлюз зв'язку між мережею та кінцевим користувачем.	1. Модель TCP/IP базується на стандартних протоколах, на яких розвивався Інтернет. Це комунікаційний протокол, який дозволяє підключати хости через мережу.
2. У моделі OSI транспортний рівень гарантує доставку пакетів.	2. У моделі TCP/IP транспортний рівень не гарантує доставку пакетів, але забезпечує з'єднання. Проте модель TCP/IP надійніша.
3. Дотримується вертикального підходу.	3. Дотримується горизонтального підходу.
4. Модель OSI має окремі представницький рівень та сеансовий рівень.	4. TCP/IP не має окремого представницького рівня або сеансового рівня.
5. OSI є еталонною моделлю, інструментом орієнтування, за якою будуються мережі.	5. Модель TCP/IP є певною мірою реалізацією моделі OSI.
6. Мережевий рівень моделі OSI забезпечує обслуговування, орієнтоване на з'єднання, і без нього.	6. Міжмережевий рівень у моделі TCP/IP забезпечує обслуговування без підключення.
7. Модель OSI має проблему доопрацювання протоколів до моделі.	7. Модель TCP/IP не потребує доопрацювання для жодного протоколу.
8. Протоколи приховані в моделі OSI та легко замінюються у міру зміни технології.	8. У TCP/IP замінити протокол нелегко.
9. Модель OSI чітко визначає сервіси, інтерфейси та протоколи та робить чітке розмежування між ними. Вона не залежить від протоколу.	9. У TCP/IP сервіси, інтерфейси та протоколи чітко не розділені. TCP/IP також залежить від протоколу.
10. Вона має 7 рівнів	10. Вона має 4 рівня

Переваги і недоліки моделі TCP/IP

Переваги моделі TCP/IP	**Недоліки TCP/IP**
1. **Сумісність**: модель TCP/IP дозволяє різним типам комп'ютерів і мереж обмінюватися даними один з одним. Має архітектуру клієнт/сервер. 2. **Масштабованість**: TCP/IP має високу масштабованість, що робить її придатним як для малих, так і для великих мереж, від локальних мереж (LAN) до глобальних мереж (WAN), таких як Інтернет. Можна використовувати для встановлення з'єднання між двома комп'ютерами. 3. **Стандартизація**: вона базується на відкритих стандартах і протоколах, що гарантує, що різні пристрої та програмне забезпечення можуть працювати разом без проблем сумісності. 4. **Гнучкість**: модель підтримує низку протоколів маршрутизації, типи даних і методи зв'язку, що робить її адаптованою до різних потреб мережі. 5. **Надійність**: TCP/IP включає функції перевірки	1. **Складна конфігурація**: Налаштування та керування мережею TCP/IP може бути складним, особливо для великих мереж з великою кількістю пристроїв. Така складність може призвести до помилок конфігурації. Замінити протокол непросто. 2. **Питання безпеки**: TCP/IP спочатку не був розроблений з урахуванням безпеки, а протоколи безпеки (наприклад, SSL/TLS), додаються поверх базової моделі TCP/IP, що може призвести до вразливостей. 3. **Неефективність для малих мереж**: Для дуже малих мереж накладні витрати та складність моделі TCP/IP можуть бути непотрібними та неефективними порівняно з простішими мережевими протоколами. 4. **Обмежено адресним простором**: хоча IPv6

помилок і повторної передачі, які забезпечують надійну передачу даних навіть на великі відстані та через різні умови мережі.	вирішує цю проблему, старіша система IPv4 має обмежений адресний простір, що може призвести до проблем із вичерпанням адреси у великих мережах. 5. **Накладні витрати на передачу даних**: TCP, транспортний протокол, включає значну кількість накладних витрат для забезпечення надійної передачі. Це може знизити ефективність, особливо для невеликих пакетів даних або в мережах, де швидкість має вирішальне значення. 6. TCP/IP **не чітко розділяє протоколи, сервіси та інтерфейси.**

2. Стек мережевих протоколів TCP/IP.

Інформаційно-комунікаційні технології істотно впливають на рівень економічної конкурентоспроможності та національної безпеки держави. Мережеві інфраструктури державних установ і приватних підприємств переважно організовані на технології Ethernet і підключені до глобальної мережі Інтернет, що функціонує на основі стеку протоколів TCP/IP.

У мережевому пристрої для обміну даними через мережу повинні бути реалізовані відповідні протоколи. Протоколи отримують дані від передаючої програми, форматують ці дані перед їх відправкою, і перетворюють їх в послідовність бітів, які потім перетворюються у відповідні оптичні, електричні або радіочастотні сигнали в середовищі передачі. На приймальному пристрої відбувається зворотне перетворення послідовності бітів назад у дані, які потрібно передати приймаючій програмі.

Розглянемо, що таке стек протоколів. Вирішення складної задачі передачі даних через мережу досягається шляхом організації функціональних рівнів в моделі TCP/IP і забезпечення надсилання даних з рівня на рівень для обробки перед відправкою на середовище для передачі. Ці послідовні функціональні рівні можна уявити як набір процесів, розташованих один над одним. На кожному рівні працюють свої протоколи. Концептуально дані з програми комп'ютера-відправника проходять вниз крізь обробку протоколами і потрапляють в середовище передачі. У програму комп'ютера-приймача дані надходять із середовища передачі вгору через обробку протоколами в зворотньому порядку (рис. 1). *Сукупність протоколів, які забезпечують взаємодію двох систем і передавання повідомлень між ними, утворює* ***стек протоколів.***

Стек мережевих протоколів TCP/IP.

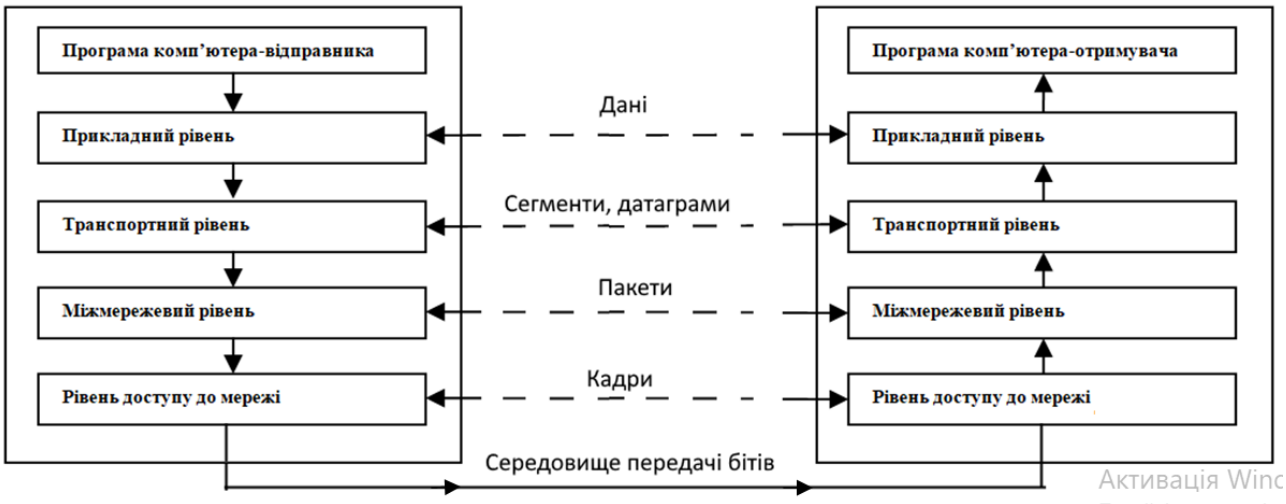


Рис. 1. Чотирирівнева модель TCP/IP.

Перевага стека протоколів TCP/IP полягає у його орієнтації на об'єднані гетерогенні мережі. Поступово, зі зростанням обчислювальної потужності комп'ютерів, стек TCP/IP суттєво потіснив інші стеки протоколів у локальних мережах і зараз є домінуючим у корпоративних мережах. Саме можливості стека протоколів TCP/IP дають змогу об'єднувати окремі підмережі (в Інтернет їх називають мережами), побудовані на різних технологіях. Це можуть бути локальні мережі Ethernet, Token Ring, а також глобальні мережі X.25, Frame Relay тощо.

Отже, найпоширенішим стеком протоколів в мережах LAN і WAN на сьогодні є стек TCP/IP. Існують реалізації відповідних протоколів моделі TCP/IP для багатьох платформ. Окремі протоколи з стека TCP/IP використовуються і в інших мережах (деякі протоколи прикладного рівня і протоколи маршрутизації).

2.1 Прикладний рівень TCP/IP (Application layer)

На цьому рівні дані з програмних додатків надсилаються в стек мережевих протоколів і передаються лініями зв'язку. На комп'ютері-одержувачі його прикладний рівень передає отримані дані програмі-одержувачу. Прикладний рівень додає відповідний заголовок і трейлер до даних відповідно до використовуваного

протоколу, перш ніж вони будуть відправлені на транспортний рівень. Прикладний рівень відповідає за наскрізний зв'язок і безпомилкову доставку даних. Він захищає програми верхнього рівня від складності даних нижчих рівнів.
Приклади протоколів прикладного рівня:
HTTP (HyperText Transfer Protocol) – керує зв'язком між веб-браузерами та веб-серверами.
HTTPS (HTTP Secure) – передача веб-сторінок (захищена версія HTTP з використанням TLS/SSL(Secure Socket Layer)). Використовується, коли в браузері потрібно заповнювати форми, входити в систему, проходити аутентифікацію і проводити банківські транзакції.
FTP (File Transfer Protocol) – передача файлів.
SMTP (Simple Mail Transfer Protocol) – передача електронної пошти.
Telnet – віддалений доступ до пристроїв.
SSH - подібний до Telnet, але підтримує зашифроване з'єднання (захищений віддалений доступ до пристроїв).
DNS (Domain Name System) – перетворення доменних імен в IP-адреси.
NTP(Network Time Protocol - протокол мережевого часу) - синхронізує годинники на нашому комп'ютері з одним стандартним джерелом часу. Протокол важливий для проведення банківських операцій.
SNMP (Simple Network Management Protocol) – протокол управління мережею (обмін даними між мережевими керуючими станціями і комп'ютерами, роутерами, шлюзами, серверами).
та інші.

2.2 Транспортний рівень TCP/IP(Transport layer)

Транспортний рівень розбиває дані програми на сегменти або датаграми. Протоколи транспортного рівня TCP/IP обмінюються даними, підтверджують їх отримання та повторно передають втрачені пакети для доставки пакетів без помилок. Це наскрізна комунікація. На цьому рівні є два протоколи:
TCP (Transmission Control Protocol) – протокол **надійної** передачі даних із встановленням з'єднання.
UDP (User Datagram Protocol) – протокол **ненадійної** передачі даних без встановлення з'єднання (потокові дані). UDP не перевіряє з'єднання між хостом-відправником і хостом-отримувачем. Програми, які передають невеликі обсяги даних, використовують UDP, а не TCP, оскільки це усуває процеси встановлення та перевірки з'єднань.

2.3 Міжмережевий рівень(Internet layer)

Міжмережевий рівень моделі TCP/IP повністю відповідає функціям мережевого рівня моделі OSI.
Міжмережевий рівень формує дані в IP-пакет для передачі по Інтернет, додає IP-адреси джерела та призначення в заголовок пакету. Адреса призначення в заголовку IP-пакета використовується маршрутизаторами для вибору маршруту в Інтернет. Міжмережевий рівень визначає протоколи, які відповідають за логічну передачу даних по всій мережі. Міжмережевий рівень передає дані на рівень доступу до мережі.
Приклад**:** відправка з комп'ютера електронного листа. Коли ви натискаєте «надіслати», електронний лист розбивається на менші пакети даних, які потім надсилаються на міжмережевий рівень для маршрутизації. Цей рівень призначає IP-адресу кожному пакету та використовує таблиці маршрутизації для визначення найкращого маршруту пакета до комп'ютера-отримувача. Потім пакет пересилається до наступного переходу (hop) на його маршруті, доки він не надійде на пункт призначення. Коли всі пакети буде доставлено, комп'ютер-отримувач збере їх у вихідне повідомлення електронної пошти.
Приклади основних протоколів міжмережевого рівня:
IP(Internet Protocol) - Інтернет-протокол, що відповідає за доставку пакетів від вихідного хоста до хоста призначення, використовуючи IP-адреси в заголовках пакетів. IP має 2 версії:

- **IPv4** (Internet Protocol) – четверта версія IP-протоколу - використовується більшістю веб-сайтів.
- **IPv6** (Internet Protocol) – шоста версія IP-протоколу з розширеним адресним простором (поступово впроваджується).

ICMP (Internet Control Message Protocol) – протокол повідомлень керування Інтернетом. Він відповідає за надання хостам інформації про проблеми з мережею, виконує контроль і діагностику помилок у мережі.
ARP (Address Resolution Protocol) – протокол визначення адрес (перетворення IP-адрес на MAC-адреси) - знаходить апаратну MAC-адресу хоста за його відомою IP-адресою.
OSPF (Open Shortest Path First) — протокол маршрутизації, що розраховує найкоротший шлях до мережі призначення за алгоритмом SPF(shortest path first).
BGP (протокол граничного шлюзу) —протокол маршрутизації пакетів із мережі в мережу шляхом обміну інформацією про маршрутизацію та доступність між периферійними маршрутизаторами.
EIGRP (Enhanced Interior Gateway Routing Protocol) - протокол маршрутизації, який використовується в комп'ютерній мережі для автоматизації рішень щодо маршрутизації та налаштування.

2.4 Рівень доступу до мережі (Network access layer).

Рівень доступу до мережі відповідає за передачу IP-пакетів через кожну з окремих ліній зв'язку, що складають шлях між двома комп'ютерами, що взаємодіють. Ці лінії зв'язку можуть бути комбінацією Ethernet, WiFi, 4G, супутникового або оптоволоконного зв'язку.IP-пакет буде переміщатися по великій кількості різних фізичних та/або бездротових каналів зв'язку між пристроєм-джерелом і пристроєм-призначенням. Кожен окремий тип фізичного каналу використовує різну технологію та має власний протокол. Кожен з протоколів додає до IP-пакетів свої заголовки для створення кадрів. Ці кадри потім використовуються для передачі IP-пакетів через цей конкретне середовище зв'язку. Отже, цей рівень відповідає за генерацію даних і запит підключень.

Основним протоколом рівня доступу до мережі є **Ethernet**. Він використовується в локальних мережах LAN і регулює спосіб передачі пакетів даних по мережі. Рівень доступу до мережі на пристрої-джерелі зв'язується з рівнем доступу до мережі на пристрої-призначення, перетворюючи кадр на електричний, електромагнітний (бездротовий) або світловий сигнал і потім надсилаючи його на носій.

Приклади основних протоколів рівня доступу до мережі:

Ethernet – стандарт для локальних мереж LAN.

PPP (Point-to-Point Protocol) – для зв'язку "точка-точка".

ATM - протокол для передачі голосу, відео та даних у мережах із високим трафіком.

Frame Relay – протокол з посиленим коригуванням помилок на кінцевих точках мережі з підтримкою комутації каналів.

Token Ring — протокол локальної комп'ютерної мережі, який має механізм передачі токенів для керування доступом у логічній кільцевій топології мережі.

HDLC(High-Level Data Link Control) —комунікаційний протокол, що забезпечує кадрування, виявлення та виправлення помилок і керування потоком.

SDLC (Synchronous Data Link Control) —біт-орієнтований протокол, що забезпечує напівдуплексну або повнодуплексну роботу, багатоточкову адресацію, керування потоком, виявлення помилок і відновлення, можливість надсилати більше одного повідомлення до отримання відповіді.

та інші.

Модель TCP/IP не має окремого фізичного рівня, як модель OSI.

Фізичний рівень в моделі OSI не охоплюється моделлю TCP/IP, оскільки рівень доступа до мережі розроблений таким чином, щоб не залежати від фізичних носіїв передачі даних. Це дозволяє TCP/IP бути гнучким і адаптуватися до різних типів фізичних з'єднань, таких як Ethernet, Wi-Fi, оптоволокно або навіть до старих технологій, таких як комутовані модеми. Фізичний рівень зазвичай забезпечується і обробляється апаратними компонентами та стандартами, специфічними для фізичного носія, що використовується, такими як кабелі Ethernet або радіохвилі для Wi-Fi.

2.5 Найвживаніші протоколи TCP/IP

Сьогодні набір протоколів TCP/IP включає багато протоколів і продовжує розвиватися для підтримки нових сервісів. Деякі з найвживаніших протоколів TCP/IP показані в таблиці 2.

Таблиця 2. Найвживаніші протоколи TCP/IP*.

Рівні TCP/IP	*Протокол системи доменних імен*	*Протоколи конфігурації хоста*	*Протоколи електронної пошти*	*Протоколи передачі файлів*	*Протоколи веб і веб-сервісу*
Прикладний рівень	**DNS**	**DHCPv4**	**SMTP**	**FTP**	**HTTP**
		DHCPv6	**POP3**	**SFTP**	**HTTPS**
		SLAAC BOOTP	**IMAP**	**TFTP**	**REST**
Транспортний рівень	*Протокол керування передачею*		*Протокол без перевірки передачі*		
	TCP		**UDP**		
Міжмережевий рівень	*Інтернет-протокол*	*Протоколи керуючих повідомлень*		*Протоколи маршрутизації*	
	IPv4	**ICMPv4**		**OSPF**	
	IPv6	**ICMPv6**		**BGP**	
	NAT	**ICMPv6 ND**		**EIGRP**	
Рівень доступу до мережі	*Протокол визначення адрес*		*Протоколи передачі даних лініями зв'язку*		
	ARP, RARP		**Ethernet**	**Token Ring**	**WLAN**

*Червоним кольором позначено TCP/IP протоколи.

Фізичний рівень в моделі OSI не охоплюється моделлю TCP/IP, оскільки рівень доступа до мережі розроблений таким чином, щоб не залежати від фізичних носіїв передачі даних. Це дозволяє TCP/IP бути гнучким і адаптуватися до різних типів фізичних з'єднань, таких як Ethernet, Wi-Fi, оптоволокно або навіть до старих технологій, таких як комутовані модеми. Фізичний рівень зазвичай забезпечується і обробляється апаратними компонентами та стандартами, специфічними для фізичного носія, що використовується, такими як кабелі Ethernet або радіохвилі для Wi-Fi.

1. Прикладний рівень включає мережеві програми та протоколи для пристроїв, що використовують мережу, в тому числі:

1) *Протокол системи доменних імен*.

1.1 **DNS:** Domain Name System - *система доменних імен* - перекладає доменні імена (наприклад cisco.com) на IP-адреси. DNS був створений для того, щоб не запам'ятовувати числові адреси інтернет-серверів. DNS завжди слухає порт 53 і перетворює доменні імена алфавітної мережі на числові IP-адреси і навпаки.

2) *Протоколи конфігурації хоста:*

2.1 **DHCPv4:** Dynamic Host Configuration Protocol version 4 - протокол динамічної конфігурації хосту для IPv4. Сервер DHCPv4 динамічно призначає інформацію про адреси IPv4 клієнтам DHCPv4 під час запуску та дозволяє повторно використовувати адреси, коли вони більше не потрібні.

2.2 **DHCPv6:** протокол динамічної конфігурації хоста для IPv6. DHCPv6 схожий на DHCPv4. Сервер DHCPv6 динамічно призначає інформацію про адресу IPv6 клієнтам DHCPv6 під час запуску.

2.3 **SLAAC:** Stateless address autoconfiguration - автоконфігурація адреси без збереження стану. SLAAC дозволяє пристрою отримувати інформацію про адресу IPv6 без використання сервера DHCPv6.

3) *Протоколи електронної пошти*:
   3.1 **SMTP:** Simple Mail Transfer Protocol - простий протокол передачі пошти. SMTP дозволяє клієнтам надсилати електронну пошту на поштовий сервер, а серверам – на інші сервери.
   3.2 **POP3**: Post Office Protocol version 3 - протокол поштового офісу версії 3. POP3 дозволяє клієнтам отримувати електронну пошту з поштового сервера та завантажувати електронну пошту в локальну поштову програму клієнта.
   3.3 **IMAP**: Internet Message Access Protocol - протокол доступу до повідомлень Інтернету. IMAP дозволяє клієнтам отримувати доступ до електронної пошти, що зберігається на поштовому сервері, а також підтримувати електронну пошту на сервері.
4) *Протоколи передачі файлів:*
   4.1 **FTP**: File Transfer Protocol - протокол передачі файлів. FTP встановлює правила, які дозволяють користувачеві на одному хості отримувати доступ і передавати файли на інший хост і з нього через мережу. FTP — це надійний, орієнтований на підключення та визнаний протокол доставки файлів.
   4.2 **SFTP**: SSH File Transfer Protocol - протокол передачі файлів SSH. Як розширення протоколу Secure Shell (SSH), SFTP використовується для встановлення безпечного сеансу передачі файлів, у якому передача файлів зашифрована. SSH — це метод безпечного віддаленого входу, який зазвичай використовується для доступу до командного рядка пристрою.
   4.3 **TFTP**: Trivial File Transfer Protocol - простий протокол передачі файлів без встановлення з'єднання з оптимальною доставкою файлів без підтвердження. Він менше витратний, ніж FTP.
5) *Протоколи веб і веб-сервісу:*
   1.1 **HTTP**: Hypertext Transfer Protocol - протокол передачі гіпертексту. HTTP – це набір правил для обміну текстом, графічними зображеннями, звуком, відео та іншими мультимедійними файлами у Всесвітній павутині.
   1.2 **HTTPS:** HTTP Secure - безпечна форма HTTP, яка шифрує дані, якими обмінюються через Всесвітню павутину.
   1.3 **REST:** Representational State Transfer - передача репрезентативного (самоописуваного) стану. REST — це метод використання інтерфейсів прикладного програмування (API) і HTTP-запитів для створення веб-додатків.

2.Транспортний рівень

Транспортний рівень надає послуги зв'язку між хостами, включаючи орієнтовані на підключення. Має 2 протоколи:

6) *Протокол керування передачею*
   6.1 **TCP***:* Transmission Control Protocol - протокол керування передачею. TCP забезпечує надійність зв'язку між процесами, що працюють на окремих хостах і *забезпечує надійні передачі з підтвердженням успішної доставки*.
7) *Протокол без перевірки передачі*
   6.2 **UDP**: User Datagram Protocol - *протокол без перевірки передачі*. UDP - це протокол дейтаграм користувача. UDP дозволяє процесу, що виконується на одному хості, надсилати пакети процесу, що виконується на іншому хості. Однак UDP не підтверджує успішну передачу датаграми.

3.Міжмережевий рівень

Інтернет-рівень використовується для транспортування пакетів із вихідного джерела до кінцевого пункту призначення. Інтернет рівень містить:

8) *Інтернет-протоколи:*
   8.1 **IPv4:** Internet Control Message Protocol version 4 - Інтернет-протокол версії 4. IPv4 отримує сегменти повідомлень від транспортного рівня, пакує повідомлення в пакети та адресує пакети для наскрізної доставки через мережу. IPv4 використовує 32-розрядну адресу.
   8.2 **IPv6**: IP версії 6. IPv6 схожий на IPv4, але використовує 128-бітну адресу.
   8.3 **NAT**: Network Address Translation - трансляція мережевих адрес. NAT перетворює IPv4-адреси з приватної мережі на глобально унікальні публічні IPv4-адреси.
9) *Протоколи обміну повідомленнями:*
   9.1 **ICMPv4**: Internet Control Message Protocol version 4 - протокол керуючих повідомлень Інтернету версії 4. ICMPv4 забезпечує зворотний зв'язок від хоста призначення до хосту джерела про помилки в доставці пакетів.
   9.2 **ICMPv6**: протокол керуючих повідомлень Інтернету версії 6. ICMPv6 за функціями подібний до ICMPv4, але використовується для пакетів IPv6.
   9.3 **ICMPv6 ND**: ICMPv6 Neighbor Discovery (протокол керуючих повідомлень Інтернету версії 6 виявлення дублікатів адрес). ICMPv6 ND містить чотири повідомлення протоколу, які використовуються для визначення адреси та виявлення дублікатів адрес.
10) *Протоколи маршрутизації*
   10.1 **OSPF**: Open Shortest Path First - першим відкрити найкоротший шлях. OSPF — це протокол маршрутизації за станом зв'язку, який використовує ієрархічну структуру на основі областей. OSPF — це відкритий стандартний протокол внутрішньої маршрутизації.
   10.2 **EIGRP:** Enhanced Interior Gateway Routing Protocol - розширений внутрішній протокол маршрутизації шлюзу. EIGRP — це власний протокол маршрутизації Cisco, який використовує складену метрику на основі пропускної здатності, затримки, навантаження та надійності.

10.3 **BGP:** Border Gateway Protocol - протокол граничного шлюзу. BGP — це відкритий стандартний протокол маршрутизації зовнішнього шлюзу, який використовується між постачальниками послуг Інтернету (ISP). BGP також широко використовується між провайдерами та їхніми великими приватними клієнтами для обміну інформацією про маршрутизацію.

4.Рівень доступу до мережі

Рівень доступу до мережі надає послуги зв'язку через фізичну мережу, як правило, від однієї картки мережевого інтерфейсу (NIC) до іншої NIC у тій самій мережі. Рівень доступу до мережі включає:

11) *Протокол визначення адреси*

11.1 **ARP**: Address Resolution Protocol - протокол визначення адрес. ARP є протоколом рівня доступу до мережі і забезпечує динамічне встановлення відповідності між IPv4-адресою пристрою та його апаратною адресою(MAC-адресою). Коли пристрої у мережі спробують взаємодіяти, вони використовують IP-адреси для ідентифікації один одного на рівні мережі Інтернет (наприклад, IPv4 або IPv6). Однак для фізичної передачі даних в мережі вони використовують MAC-адреси. Тому ARP перетворює IP-адреси пристроїв у відповідні їм MAC-адреси. Коли пристрою в мережі необхідно надіслати пакет даних на визначену IP-адресу, він відправляє ARP-запит на отримання MAC-адреси пристрою з відповідною IP-адресою. Запрошується відповідь від пристрою з відповідною IP-адресою, який повертає свою MAC-адресу. Після отримання відповіді пристрій може створити відображення між IP-адресою і MAC-адресою в своєму кеші ARP і використовувати його для подальшої комунікації з цим пристроєм. ARP використовується в IPv4-мережах, але аналогічний протокол, відомий як ICMPv6 ND: ICMPv6 Neighbor Discovery протокол, використовується в IPv6-мережах.

12) *Протоколи передачі даних лініями зв'язку:*

12.1 **Ethernet**: Ethernet визначає правила підключення та стандарти сигналізації рівня доступу до мережі.

12.2 **WLAN**: Wireless local-area network - бездротова локальна мережа. Протоколи WLAN визначають правила бездротової передачі сигналів на радіочастотах 2,4 ГГц і 5 ГГц.

Отже, стек TCP/IP протоколів регулює взаємодію різних рівнів. Ключовим поняттям тут є протоколи, які утворюють стек, спираючись один на одного для передачі даних. TCP/IP модель має спрощену архітектуру в порівнянні з OSI. Сама TCP/IP модель залишається незмінною, а стандарти протоколів можуть бути оновлені, що робить роботу з TCP/IP ще простішою. Стек TCP/IP отримав широке поширення і використовувався спочатку як основа для створення глобальної мережі, а пізніше і для створення стабільного інтернету.

3. Протоколи прикладного рівня (DNS, HTTP та HTTPS, FTP, SMTP, POP3, IMAP4, Telnet, SSH та SSL, SNMP).

Як працює протокол?

Весь технічний процес передачі даних через мережу розбитий на окремі кроки.

1. На кожному кроці відбуваються певні дії, які не можуть відбутися на жодному іншому кроці. Кожен крок містить власні правила та процедури або протокол.
2. Етапи протоколу виконуються в узгодженому порядку, який є однаковим для кожного комп'ютера в мережі.
3. На комп'ютері-відправнику ці кроки виконуються зверху вниз. В комп'ютері-одержувачі, ці кроки виконуються знизу вгору.
4. Комп'ютери-відправники та комп'ютери-одержувачі виконують кожен крок однаково, щоб дані мали таку саму структуру при отриманні, як і при відправленні.

Вимоги до мережевого протоколу

Мережеві протоколи мають багато основних рис. Окрім ідентифікації джерела та призначення, вони визначають деталі того, як повідомлення передається через мережу. Загальні комп'ютерні протоколи містять такі вимоги:

2. *Кодування повідомлень*
3. *Форматування та інкапсуляція повідомлень*
4. *Розмір повідомлення*
5. *Час повідомлення*
6. *Варіанти доставки повідомлень*

3.1 Протокол DNS (система доменних імен) — це протокол, який використовується для перетворення IP-адрес в легко читані людиною імена доменів для спілкування через мережу. За допомогою протоколу DNS ієрархічний набір DNS-серверів спільно зберігає всі відповідності доменного імені та IP-адреси, а DNS-сервери спрямовують користувачів до законного DNS-сервера для успішного визначення DNS-імені. Також протокол DNS підтримує адресацію IPv6. DNS завжди слухає порт 53 і перетворює доменні імена мережі на числові IP-адреси і навпаки.

DNS — це глобальна мережа, яка перетворює зручні доменні імена в числові IP-адреси, які використовуються комп'ютерами для ідентифікації веб-сайтів в Інтернеті. Ієрархія DNS складається із серверів, які разом утворюють величезну глобальну базу даних. Ієрархія DNS (простір доменних імен) має структуру дерева гілками вниз (перевернуте дерево). DNS використовує ієрархію для керування системою розподілених баз даних.

Домен — це мітка дерева DNS. Кожен вузол на дереві DNS представляє домен. Доменне ім'я – це просто список усіх доменів на шляху від локального домену до кореневого. Кожна мітка в доменному імені розділена крапкою.

Що таке DNS-сервери? DNS-сервери, які також функціонують як веб-сервери, відіграють вирішальну роль у цьому процесі перекладу, перетворюючи доменні імена, введені у веб-браузерах, у відповідні IP-адреси.

Що таке повне доменне ім'я (FQDN)? Повне доменне ім'я— це термін, пов'язаний із DNS, який визначає точне розташування домену в ієрархії. Він складається з імені хоста та імені домену, наприклад «**mail.mydomain.com**». Повне ім'я домена пропонує чіткий і однозначний метод визначення розташування ресурсів Інтернету. Маючи в назві імена хоста і доменні імена, наприклад «**mail**» і «**mydomain.com**», повне доменне ім'я полегшує точну ідентифікацію ресурсу. По суті, імена доменів забезпечують чіткий шлях для навігації розподіленою мережею Інтернету. Використання повного доменного імені забезпечує точне спілкування та пошук ресурсів, сприяючи загальній ефективності та надійності мережі.

Ієрархія DNS . Існує п'ять рівнів ієрархії DNS:

1**. Кореневий рівень (Root Level)** є вершиною ієрархії DNS, відіграючи головну роль у процесі визначення домену. Ієрархія серверів DNS починається на кореневому рівні, де міститься коренева зона DNS, якою керують кореневі сервери імен DNS. Сервери кореневої зони DNS перенаправляють запити до відповідних DNS верхнього рівня (TLD - Top Level Domain - домени верхнього рівня), позначаючи початок перекладу зрозумілих людині доменних імен на IP-адреси.

2. **Домени верхнього рівня .** Домени верхнього рівня знаходяться безпосередньо під кореневим рівнем в ієрархії DNS. Це широко визнані розширення, кожне з яких відображає організаційну ієрархію (наприклад, .com, .net, .org, .edu) або географічні відмінності (наприклад, .ca, .uk, .fr, .ua). Домени в доменах верхнього рівня представляють окремі організації або установи. Сервери доменних імен верхнього рівня :• «com» для комерційних веб-сайтів (наприклад, novell.com). • «org» для організаційних веб-сайтів(наприклад, wto.org). • «edu» для освітніх веб-сайтів (наприклад, mit.edu). • «net» для мережевих організацій(наприклад, ukr.net). • «gov» для урядових веб-сайтів (наприклад, mfa.gov). • «mil» для військових сайтів (наприклад, army.mil). • «int» для веб-сайтів міжнародних організацій (наприклад, coe.int).

3**. Домени другого рівня .** Безпосередньо під доменами верхнього рівня знаходяться домени другого рівня (Second Level Domain). Ці домени є специфічними для організацій або організацій і служать основними ідентифікаторами в веб-адресах (рис.2).

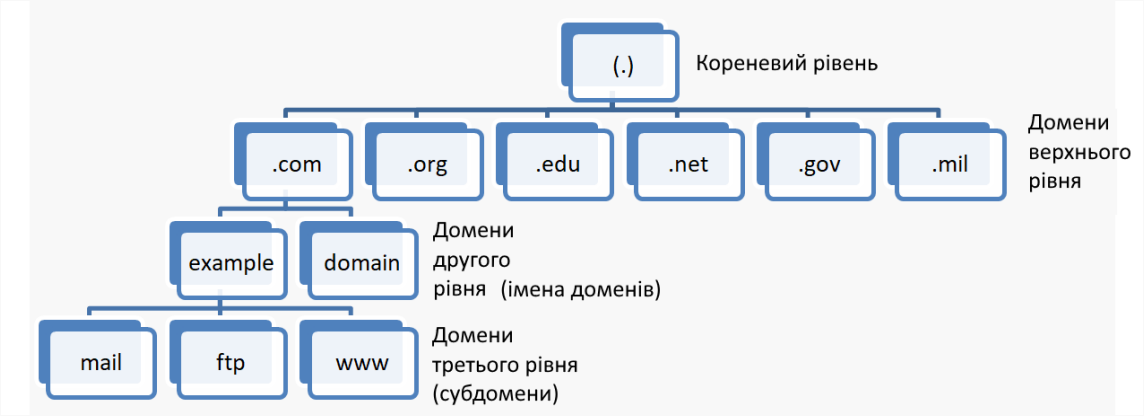


Рис.2 Ієрархія DNS.

Керовані реєстраторами доменів домени другого рівня знаходяться під адмініструванням організації або особи, якій вони призначені. Приклади включають «**example**» у «**example.com**», а домени другого рівня відіграють вирішальну роль в організації та категоризації вмісту в домені.

4. **Субдомени (піддомени)**. Безпосередньо під доменами другого рівня в ієрархії DNS знаходяться субдомени, які полегшують адміністрування хост-комп'ютерів організації. Вони формують додаткову організаційну структуру веб-сайту, підвищуючи гнучкість у дизайні та управлінні контентом. Наприклад , в «**blog.example.com**» домен «**blog**» є субдоменом «**example.com**». Мережевий адміністратор компанії може створити окремий субдомен для кожного підрозділу компанії. Будь-який домен у піддереві є частиною всіх доменів над ним. Таким чином, «**blog.example.com**» є частиною домену «**example.com**», і обидва є частиною домену «**.com**».

5. **Хости**. Розділ хосту повного домену використовується для ідентифікації конкретного пристрою, як правило, виділеного сервера.

Сервер кореневої зони

Ієрархія DNS спирається на 13 розподілених серверів кореневої зони в усьому світі, на відміну від одноточкового зображення. Призначені ICANN (The Internet Corporation for Assigned Names and Numbers) такі організації, як **Verisign, University of Maryland** і **NASA**, керують цими серверами. Вони відіграють вирішальну роль у перетворенні доменних імен на IP-адреси. Географічна розпорошеність і резервування цих серверів підвищують стійкість і ефективність системи DNS.

Щоб полегшити широку обізнаність про сервери кореневої зони, імена хостів і відповідні IP-адреси жорстко закодовані в програмному забезпеченні DNS і розповсюджені на хости та резолвери. 13 кореневих серверів, ідентифікованих іменами хостів, як-от **a.root-servers.net** до **m.root-servers.net**, управляються різними організаціями, включаючи державні установи, навчальні заклади та приватні компанії. Ця жорстко закодована інформація служить початковою точкою відліку для вирішення DNS, дозволяючи хостам і резолверам ініціювати процес запиту до авторитетних серверів, щоб остаточно перевести доменні імена в IP-адреси. Навмисне розподілення та резервування цих кореневих серверів сприяє стабільності та надійності глобальної інфраструктури DNS.

Як працює протокол DNS. Важливо знати, що є не один DNS, а їх вибір, який допомагає визначити IP-адресу. Комп'ютер має власний кеш даних DNS – локальне зіставлення IP-адреси з URL-адресою – який він використовує для швидких посилань. Але він не може зберігати всю інформацію для кожного веб-сайту чи пристрою. Звідси потреба в ієрархії DNS –серверів.

Коли користувач просить дозволу зайти на сайт, браузер спочатку перевіряє локальний DNS-кеш на правильність дозволу (від URL до IP-адреси) і, якщо не знаходить, відбувається:

1. Зверненняя до **рекурсивного DNS-сервера**, який зазвичай використовується місцевим провайдером Інтернету, щоб перевірити, до якого кореневого DNS-сервера потрібно звернутися, щоб знайти відповідь. Як тільки адреса знайдена, йде перехід на...
2. Один із **кореневих DNS-серверів**, якими керують близько 13 незалежних організацій, для пошуку адреси правильного DNS-сервера верхнього рівня (TLD) для запиту адреси залежно від того, чи це сайт «.com», «.org» або «.net», наприклад. Кореневі DNS знаходяться на вершині ієрархії DNS. Тут слід зазначити, що кожна адреса веб-сайту або URL-адреса має неявне «.» в кінці, навіть якщо ми його не вводимо. Цей символ "." позначає (або вказує на) кореневі сервери імен DNS у верхній частині ієрархії DNS, де він може знайти інформацію про правильний сервер TLD, до якого слід перейти.
3. **На сервері верхнього** рівня, яких налічується близько 1000 по всьому світу, запит спрямовується на правильний авторитетний DNS, де насправді зберігається необхідна інформація (IP-адреса).
4. Крім того, на **авторитетному DNS-сервері**, який містить широкий спектр інформації про IP-адресу, відповідна IP-адреса надсилається назад на рекурсивний DNS, який пересилає її на клієнтську машину, і користувач (ви) може перейти на веб-сайт.

Щоразу, коли робиться запит, зіставлення IP-адреси з URL кешується в кожному DNS для наступного користувача замість того, щоб щоразу повертатися назад і шукати інформацію. Це допомагає синхронізувати та оновлювати сервери, а також призводить до швидшого часу відгуку.

Переваги і недоліки DNS

Переваги	Недоліки
1. *Без DNS не існувало б інтернету*. Загалом DNS – це єдина система у всьому світі, яка може допомогти вам переглядати веб-сторінки. Стає все більш важливим, щоб DNS-сервери продовжували підтримуватися. 2. *Немає необхідності запам'ятовувати IP-адреси* - DNS-сервери надають гарне рішення для перетворення доменних або субдоменних імен в IP-адреси (не треба запам'ятовувати IP-адреси Twitter, Facebook, Google та інших щодня відвідуваних сайтів). DNS система також дозволяє пошуковим системам легко класифікувати та архівувати інформацію. 3. *Підвищена безпека* - DNS-сервери гарантують, що спроби злому вашого серверного середовища будуть зірвані перед входом на ваші машини. Однак для великих організацій важливе додаткове налаштування безпеки. 4. *DNS-сервери мають швидке інтернет-з'єднання*. 5. *DNS-сервери також мають основне та додаткове підключення*. Це дозволяє вам мати час безвідмовної роботи Інтернету, навіть коли один із серверів не працює (на технічному обслуговуванні).	1. Одним з головних недоліків DNS є те, що *його реєстром може керувати тільки ICANN* - некомерційна організація. 2. *DNS-запити зазвичай не несуть жодної інформації про клієнтів*, які їх ініціювали. Це одна з причин, чому DNS користується популярністю серед хакерів. Це пов'язано з тим, що сторона сервера бачить лише IP-адресу, звідки надійшов запит, і якою можуть маніпулювати хакери. 3. *DNS-сервери засновані на принципі відносин slave-master*. Це означає, що якщо головний сервер зламаний або ним якимось чином маніпулюють, то буде важко отримати доступ до веб-сторінки або бази даних, яка була розміщена на сервері. Хакери також використали це на свою користь. Націлившись на серверну машину та роблячи перенаправлення на інші сторінки, вони знаходять способи фішингу інформації.

Додатково:

RFCs:

1. RFC 1034 – “Domain Names – Concepts and Facilities” – Provides fundamental concepts of DNS, relevant to understanding Master Name Servers.

2. RFC 5936 – “DNS Zone Transfer Protocol (AXFR)” – Details the protocol used for zone transfers from Master Name Servers.

3.2 Протоколи передачі гіпертексту HTTP і HTTPS - це протоколи, що використовуються для передачі веб-сторінок. Протокол HTTP забезпечує незахищений зв’язок. Протокол HTTPS забезпечує захищене спілкування за допомогою шифрування SSL/TLS.

3.2.1 Протокол HTTP (Протокол передачі гіпертексту)– це стандартний протокол Інтернету, який визначає процеси взаємодії клієнта та сервера між веб-браузерами, такими як Microsoft Internet Explorer або Chrome, і веб-серверами, такими як Microsoft Internet Information Services (IIS) або Apache. HTTP дотримується моделі клієнт-сервер, коли клієнт відкриває з'єднання, робить запит, а потім чекає, поки не отримає відповідь. Протокол HTTP описано в RFC 7540 – HTTP/2.

Як працює HTTP

HTTP— це протокол без стану, за допомогою якого веб-браузер встановлює з'єднання з веб-сервером, завантажує відповідний файл, а потім розриває з'єднання. Браузер зазвичай запитує файл за допомогою методу HTTP GET запиту на TCP-порту 80, який складається з ряду заголовків HTTP-запиту, що визначають метод транзакції (GET, POST, HEAD і так далі) і вказує серверу на можливості клієнта. Сервер відповідає серією заголовків відповіді HTTP, які вказують, чи була транзакція успішною, тип даних, що надсилаються, тип сервера і, нарешті, запитувані дані.

Команди HTTP-запиту.

1. **GET**: метод GET запитує представлення вказаного ресурсу. Запити, що використовують GET, повинні отримувати лише дані.
2. **HEAD**: Метод HEAD запитує відповідь, ідентичну відповіді GET-запиту, але без тіла відповіді.
3. **POST**: Метод POST надсилає сутність на вказаний ресурс, що часто спричиняє зміну стану або побічні ефекти на сервері.
4. **PUT**: метод PUT замінює всі поточні представлення цільового ресурсу корисним навантаженням запиту.
5. **DELETE:** метод DELETE видаляє вказаний ресурс.
6. **CONNECT**: метод CONNECT встановлює тунель до сервера, визначеного цільовим ресурсом.
7. **TRACE**: метод TRACE виконує тест зворотного зв'язку повідомлень на шляху до цільового ресурсу.
8. **PATCH**: Метод PATCH використовується для застосування часткових модифікацій до ресурсу.

Основні можливості протоколу HTTP:

- **Постійні зв'язки:** Сервер HTTP може тримати TCP-з'єднання відкритими після передачі файлу, усуваючи необхідність відкривати та закривати з'єднання щоразу під час передавання файлу.
- **Тунелювання(Pipelining):** клієнт HTTP може надсилати кілька пакетів Інтернет-протоколу (IP) на сервер, не чекаючи, поки сервер відповість на кожен пакет.
- **Буферизація:** дозволяє декільком HTTP-запитам клієнта бути буферизованими в один пакет і відправлятися на сервер, що призводить до швидшого часу передачі, оскільки використовується менше і більше пакетів.
- **Заголовки хостів:** дає змогу веб-серверу, сумісному з HTTP , розміщувати кілька веб-сайтів за допомогою однієї IP-адреси.
- **Команди http put і http delete:** дозволяють веб-браузерам завантажувати та видаляти файли з веб-серверів за допомогою протоколу HTTP.

Заголовки HTTP дозволяють клієнту та серверу передавати додаткову інформацію за допомогою HTTP-запиту або відповіді.

Список кодів стану HTTP

Коли користувач відвідує веб-сайт за допомогою інтернет-браузера, браузер зв'язується із сервером, який обробляє запити браузера. Сервер відповідає на запит певною дією, дозволяючи веб-сайту завантажитися або відображає повідомлення про помилку. Такі дії визначаються протоколом передачі гіпертексту (HyperText Transfer Protocol, HTTP). Хоча всього існує 63 коди стану HTTP, деякі коди досить поширені (наприклад, класична помилка 404 «Not Found - Не знайдено»), тоді як інші з'являються рідко. Ось деякі з найпоширеніших кодів стану HTTP, які нумеруються стандартом Internet Engineering Task Force (IETF) 9110.

Категорії кодів стану HTTP

5 категорій кодів стану HTTP:

1xx: статус інформаційний: Інформація, отримана сервером у відповідь на запит.
2xx: статус успішний: Запит виконано успішно.
3xx: статус перенаправлення: URL-адресу веб-сайту було перенаправлено.
4xx: статус помилка клієнта: Сторінку неможливо знайти.
5xx: статус помилка сервера: Клієнт (в даному випадку браузер) зробив формально правильний запит, але сервер не може його виконати.

Коди стану HTTP:

1. **200 (OK):** Запит успішно зрозумілий і прийнятий сервером.
2. **301 (переміщено назавжди):** веб-адресі призначено нову URL-адресу, посилання на яку іноді міститься в повідомленні про помилку.
3. **302 (знайдено):** веб-адреса тимчасово знаходиться під іншою URL-адресою.
4. **303 (Дивіться інше):** користувач спрямовується сервером на інший веб-сайт або ресурс.

5. **304 (Не змінено):** якщо веб-сторінка не була змінена з моменту останнього доступу користувача, для сервера може бути економніше відображати кешовану версію або останню збережену копію сторінки. Щоб визначити, чи була сторінка змінена, сервер надсилає умовний запит. Якщо сторінка не була змінена, сервер надсилає код HTTP 304 і показує кешовану веб-сторінку користувачеві. Якщо сторінка оновилася, сервер надсилає стандартний код 200.
6. **400 (неправильний запит):** сервер не може обробити запит через очевидну помилку клієнта.
7. **401 (неавторизовано):** користувач не може отримати доступ до URL-адреси, оскільки в нього немає прийнятних облікових даних для автентифікації.
8. **403 (заборонено):** Подібно до помилки 401 це означає, що користувач не має доступу до URL-адреси. Ця помилка виникає після прийнятних облікових даних автентифікації, але вказує на відсутність дозволу.
9. **404 (Не знайдено):** сервер наразі не може знайти сторінку за URL-адресою. Це відрізняється від 410 (Зникла), яке означає, що сторінка зникла назавжди.
10. **500 (внутрішня помилка сервера):** на сервері сталася неочікувана помилка.
11. **503 (служба недоступна):** сервер недоступний, можливо, через перевантаження, але це не означає постійну недоступність.

3.2.2 Протокол HTTPS (Hyper Text Transfer Protocol Secure) є еволюцією HTTP, протоколу, який полегшує передачу даних в Інтернеті. Літера «S» у HTTPS, що означає «Secure - Захищено», працює на базі TLS (Transport Layer Security) або його попередника, SSL (Secure Sockets Layer). Ці криптографічні протоколи шифрують дані, які пересилаються між браузером користувача та сервером. Робота SSL, TLS починається з обміну асиметричними публічними ключами для гарантії, що обидві сторони дійсно є тими, за кого вони себе видають. Тут обидві сторони — зазвичай веб-браузер і сервер — домовляються про те, який алгоритм шифрування використовувати. Ефективність TLS і SSL ґрунтується на криптографічних алгоритмах від RSA і Діффі-Хеллмана для обміну ключами до AES для шифрування даних. Це шифрування гарантує, що будь-який зловмисник, який перехопить дані, їх не зрозуміє. В результаті, незалежно від того, вводите ви дані кредитної картки, використовуєте платформи електронної комерції, електронний банкінг або навіть просто переглядаєте веб-сторінки, HTTPS гарантує, що дані будуть захищені в Інтернеті. HTTPS залишається сучасним на тлі кіберзагроз, що швидко розвиваються. HTTPS використовує TCP відомий порт з номером 443 замість порту 80, який використовується HTTP.

Різниця між HTTP і HTTPS

HTTP	HTTPS
HTTP розшифровується як HyperText Transfer Protocol (протокол передачі гіпертексту).	HTTPS для безпечного протоколу передачі гіпертексту.
У HTTP URL починається з "http://".	У HTTP URL починається з "https://".
HTTP використовує порт номер 80 для зв'язку.	HTTP використовує номер 443 порту для зв'язку.
HTTP вважається небезпечним.	HTTP вважається безпечним.
HTTP працює на прикладному рівні.	HTTPS працює на транспортному рівні.
У протоколі HTTP шифрування немає.	Шифрування є в HTTPS.
HTTP не вимагає жодних сертифікатів.	Для роботи з HTTPS потрібні SSL-сертифікати.
HTTP використовують старі текстові веб-сайти	HTTPS використовують всі сучасні сайти
HTTP швидше, ніж HTTPS	HTTPS повільніше, ніж HTTP
HTTP не використовує хештеги даних для захисту даних.	HTTPS має дані перед їх відправкою і повертає їх у початковий стан на стороні одержувача.
У HTTP дані передаються як відкритий текст.	У HTTPS дані передаються як зашифрований текст.
Пошукові системи не віддають перевагу небезпечному веб-сайту.	Поліпшення репутації сайту в пошуковій системі.
HTTP не вимагає SSL/TLS або сертифікатів	HTTPS вимагає впровадження SSL/TLS із сертифікатами.
У HTTP користувачі не впевнені в безпеці своїх даних.	У HTTPS Користувачі впевнені в безпеці своїх даних.

Додатково:

1. World Wide Web Consortium (W3C)
2. Ви можете дізнатися більше про протокол HTTP: TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols (paperback) (Addison-Wesley Professional Computing Series)
3. Безпека/Сервер на стороні TLS, mozilla wiki

3.3 Протокол FTP(File Transfer Protocol - Протокол передачі файлів**) -** це стандартний протокол прикладного рівня TCP/IP, який можна використовувати для передачі файлів між хостами в Інтернет .
Протокол FTP (рис. 3) є одним із найперших протоколів Інтернету і досі використовується для завантаження та вивантаження файлів між клієнтами та серверами. FTP має архітектуру клієнт-сервер. Користувачі FTP можуть автентифікуватися за допомогою протоколу входу з відкритим текстом, зазвичай у формі імені користувача та пароля, але можуть підключатися анонімно, якщо сервер відповідно налаштовано. Для безпечної передачі з шифруванням вмісту FTP використовує протокол SSL/TLS (FTPS) або протокол SSH (SFTP).
FTP-клієнт — це програма, яка може віддавати команди FTP-серверу, а FTP-сервер — це сервіс або демон, запущений на сервері, який відповідає на команди FTP - клієнта. Команди FTP використовують для зміни каталогів, зміни режимів передачі між двійковим кодом і ASCII, завантаження файлів, та інші.

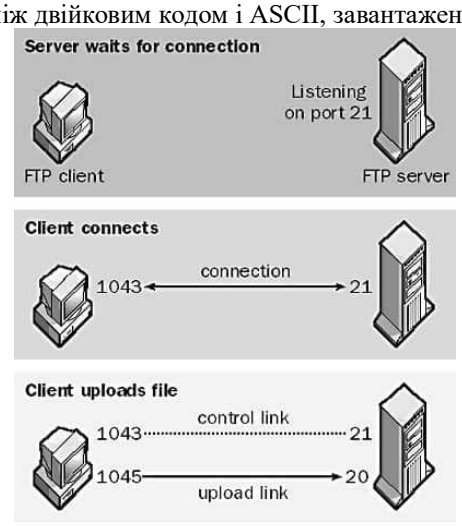


Рис. 3. Протокол передачі файлів FTP.[File Transfer Protocol (FTP)]
FTP використовує протокол керування передачею TCP для надійного мережевого зв'язку, встановлюючи сеанс перед початком передачі даних. TCP-**порт номер 21** на FTP-сервері прослуховує спроби з'єднання з FTP-клієнтом і є портом управління для встановлення з'єднання між клієнтом і сервером, для дозволу клієнту відправити FTP-команду на сервер і для повернення відповіді FTP-сервера на команду.
Після встановлення контрольного з'єднання FTP-сервер відкриває **порт номер 20** для нового з'єднання з клієнтом для передачі фактичних даних під час завантажень і вивантажень.

Багато FTP-хостів з метою надання оновлень програмного забезпечення, дозволяють здійснювати анонімні входи.
Додатково:ПРОТОКОЛ ПЕРЕДАВАННЯ ФАЙЛІВ RFC 959 (FTP)

3.4 Протоколи електронної пошти POP3, IMAP4 і SMTP.

Протоколами електронної пошти є протокол поштового відділення POP3, протокол доступу до пошти в Інтернеті IMAP4 і простий протокол передачі пошти SMTP. Ці протоколи дуже різні.
Основні протоколи: короткий огляд

- **POP3**: призначений для завантаження повідомлень з поштового сервера на локальний клієнт за вимогою. Працює через порт 110 або 995 (SSL/TLS).
- **IMAP4**: надає доступ кільком пристроям до однієї поштової скриньки, пропонує складніші функції - керування папками та позначення електронної пошти. Працює через порт 143 або 993 (SSL/TLS).
- **SMTP**: доставляє електронні листи від клієнта до сервера або від сервера до сервера. Він не отримує повідомлення і працює через порт 25, 587 або 465 (SSL/TLS).

Коли що використовувати?

1. **POP3** - для налаштувань на одному пристрої, з обмеженими ресурсами сервера та потребою в автономному доступі до електронної пошти.
2. **IMAP** - для користувачів з доступом до електронної пошти з кількох пристроїв з узгодженим поданням запитів.
3. **SMTP** - завжди використовується у поєднанні з POP3 або IMAP для обробки надсилання електронних листів.

Розуміння основних функцій і відмінностей між протоколами POP3, IMAP і SMTP має важливе значення для конфігурації електронної пошти.

Рекомендації до використання POP3 та IMAP.
Якщо треба доступ в автономному режимі та маєте один пристрій для електронної пошти, використовуйте протокол POP3. Якщо вам потрібні розширені функціональні можливості або доступ до електронної пошти з кількох пристроїв, використовуйте протокол IMAP.

3.4.1 Протокол, POP3 (Post Office Protocol, 3) POP3 — це один з основних протоколів прикладного рівня, який використовується у світі для обміну електронною поштою. Сеанси POP3 передбачають обмін повідомленнями електронної пошти між клієнтом і сервером. Протокол POP3 описано в RFC 1939. За замовчуванням служба POP3 доступна через порт 110. Найширше використовується протокол POP3. Сеанс складається з двох типів повідомлень: *команд і відповідей.*

Сеанс POP3 проходить у кілька етапів:

1. *З'єднання*: клієнт відкриває з'єднання, а сервер відповідає позитивним привітанням, наприклад, S: +OK Сервер POP3 встановив з'єднання і готовий приймати запити від користувача.
2. *Авторизація*: після встановлення з'єднання сервер вимагає від клієнта пройти автентифікацію. Для цього доступні два механізми. Один із них передбачає шифрування та використовується, коли користувач не хоче надсилати незашифровані паролі. Другий механізм - незашифровані паролі.
3. *Транзакція*: користувач перевіряє, отримує або видаляє повідомлення. Ця стадія може тривати невизначений час.
4. *Оновлення*: за запитом користувача QUIT сервер оновлює файли (наприклад, видалення). Якщо з'єднання втрачається до того, як користувач дає команду QUIT, сервер не оновлює файли і залишає їх такими, якими вони були до початку сеансу користувачем.

Протокол POP3 дає змогу поштовим клієнтам — наприклад, Microsoft Outlook, Apple Mail або Thunderbird — отримувати повідомлення з віддаленого SMTP-сервера електронної пошти на їх локальний пристрій. Працюючи через порт 110 для незашифрованих сеансів і порт 995 для сеансів із шифруванням SSL/TLS, POP3 є одним із основних протоколів, що забезпечують роботу електронної пошти.
Коли ви підключаєте поштовий клієнт до сервера POP3, сервер відображає список нових повідомлень, якщо такі є. Потім ваш поштовий клієнт завантажує ці повідомлення та зазвичай видаляє їх із сервера. Тепер ваші електронні листи зберігаються на локальному пристрої, доступні у режимі офлайн.

Основні функції POP3:

- *Отримання повідомлення*: основне завдання POP3. Він доставляє електронні листи з сервера на ваш локальний комп'ютер.
- *Використання порту*: стандартний незашифрований POP3 працює через порт 110, а POP3 з SSL/TLS (з шифруванням) працює через порт 995.
- *Операція без постійного з'єднання***:** POP3 не підтримує жодного постійного з'єднання між клієнтом і сервером. Кожен сеанс є незалежним і завжди завершується видаленням завантажених повідомлень із сервера.
- *Автентифікація*: до відправки вам ваших електронних листів POP3 переконається, що це дійсно ви. Ваше ім'я користувача та пароль нададуть доступ до POP3-сервера (можна використовувати і більш просунуті функції безпеки).

Протокол SMTP надає базовий механізм для надсилання повідомлень електронної пошти через Інтернет, але не має механізмів зберігання повідомлень та їх пошуку. Протокол POP3 надає механізми зберігання повідомлень, надісланих кожному користувачеві та отриманих за допомогою SMTP у контейнері - поштовій скриньці. Сервер POP3 зберігає повідомлення для кожного користувача, доки користувач не підключиться до них для завантаження та читання за допомогою клієнта POP3, наприклад Microsoft Outlook, Microsoft Outlook Express або Microsoft Mail and News.

Алгоритм роботи POP3.

Крок 1. Ініціалізація сеансу (рукостискання)
Сеанс починається, коли поштовий клієнт ініціює TCP-з'єднання із сервером. Сервер відповідає привітанням `+OK` - сигналом про початок транзакції. Будь-який збій на цьому кроці припиняє сеанс.

Крок 2. Аутентифікація
Перед обміном даними ви повинні пройти аутентифікацію (надіслати ім'я користувача та пароль). Для більшої безпеки застосовують протокол автентифікованого поштового відділення APOP для обміну хешованими обліковими даними.

Крок 3. Транзакції
Після успішної аутентифікації настає фаза транзакції.

Ключові команди POP3:

1. **STAT:** отримує статистику поштових скриньок.
2. **LIST**: список номерів і розмірів повідомлень.
3. **RETR:** Отримує певне повідомлення.
4. **DELE:** Позначає повідомлення для видалення.
5. **NOOP**: Команда без роботи, щоб підтримувати зв'язок.
6. **QUIT:** завершує сеанс POP3.

Кожна команда клієнта має відповідь від сервера, яка або підтверджує успішне виконання `+OK`, або вказує на збій `-ERR`.

Крок 4.Оновлення

Закриття сеансу – це фаза оновлення. Після виконання команди `QUIT` клієнт переводить сервер у фазу оновлення і всі позначені для видалення повідомлення назавжди видаляються з поштової скриньки. Після цього сервер надсилає відповідь `+OK` і закриває TCP-з'єднання.

Важливою особливістю POP3 є те, що сервер надсилає дані клієнту за його вимогою.

Переваги та недоліки POP3

Переваги	Недоліки
1. *Автономний доступ*: електронні листи зберігаються локально. 2. *Ефективність використання ресурсів*: Звільняє місце на сервері, видаляючи повідомлення після завантаження (допомагає уникнути квот на поштові скриньки). 3. *Безпека:* Для додаткової безпеки при автентифікації підтримуються безпечні з'єднання через SSL/TLS. 4. *Простота та швидкість*: завдяки простому набору команд POP3 може бути швидшим і простішим, ніж IMAP, при роботі з великими поштовими скриньками. 5. *Ефективність пропускної здатності*: оскільки електронні листи завантажуються та зберігаються локально, послідовні перевірки нових електронних листів споживають менше пропускної здатності, ніж протоколи, які зберігають повідомлення на сервері	1. *Немає синхронізації з кількома пристроями*: повідомлення прив'язуються до одного пристрою. 2. *Обмежене керування на стороні сервера*: не має розширених функцій (організація папок, , позначення та категоризація електронних листів). 3. *Відсутність синхронізації з декількома пристроями*: на відміну від IMAP, POP3 не підтримує синхронізацію на кількох пристроях. Якщо ви прочитали електронний лист на одному пристрої, він не буде позначений як прочитаний на інших. 4. *Ризик втрати даних:* оскільки електронні листи завантажуються в локальне сховище, апаратний збій може призвести до втрати даних, якщо не використовується стратегія резервного копіювання. 5. *Операція без постійного з'єднання:* сеанси POP3 не мають постійного з'єднання, що означає, що видалення повідомлення є незворотним після завершення сеансу. 6. *Відсутність вбудованого шифрування*: Хоча можна використовувати SSL/TLS, POP3 не надає наскрізного шифрування, що вимагає додаткових заходів безпеки.

Безпека POP3.

Вразливості POP3	Заходи для покращення безпеки POP3
1. *Автентифікація у вигляді незашифрованого тексту*: POP3 передає імена користувачів і паролі без шифрування, вони відкриті для прослуховування. 2. *Відсутність наскрізного шифрування*: Хоча SSL/TLS може захистити з'єднання між клієнтом і сервером, POP3 не має наскрізного шифрування для вмісту електронної пошти. 3. *Ризики на стороні сервера*: оскільки електронні листи завантажуються та видаляються з сервера, скомпрометований сервер може призвести до втрати непрочитаних або незбережених повідомлень.	1. *SSL/TLS*: увімкнення Secure Sockets Layer або Transport Layer Security забезпечує зашифровану передачу даних між клієнтом і сервером. 2. *APOP*: використовує хешування MD5 для захисту облікових даних. 3. *Брандмауери та системи виявлення вторгнень (IDS)*: попереджають адміністраторів про будь-які підозрілі дії зі службами електронної пошти. 4. *Регулярні оновлення програмного забезпечення електронної пошти та сервера*: гарантує, що ви отримаєте вигоду від останніх виправлень безпеки. 5. *Двофакторна автентифікація (2FA):* POP3 не підтримує *2FA* за замовчуванням, але 2FA може бути додатковим рівнем безпеки під час входу в систему.

Додаткова література:

1. RFC 1939 – "Post Office Protocol – Version 3": Детальний технічний огляд POP3, автором якого є Internet Engineering Task Force (IETF). (Читати Далі)
2. "Архітектура, дизайн та реалізація електронної пошти" Кевіна Томаса: У цій книзі обговорюється архітектура служб електронної пошти та протоколів, включаючи POP3, а також пропонуються найкращі практики для конфігурації та безпеки. (Читати Далі)

3.4.2 Протокол IMAP4 (Internet Mail Access Protocol version 4) – стандартний протокол прикладного рівня моделі TCP/IP для збереження та отримання повідомлень електронної пошти з SMTP в мережевому поштовому відділенні. Це протокол доступу до інтернет-пошти. IMAP4 описно в RFC 1730. Сервер IMAP4 зберігає повідомлення, отримані кожним користувачем, доки користувач не підключиться до них для завантаження та читання за допомогою клієнта IMAP4, наприклад Microsoft Outlook або Microsoft Outlook Express.
Протокол IMAP4 надає функції, подібні до протоколу поштового відділення POP3.

Функції IMAP4:

1. Доступ до кількох папок, включно зі спільними папками
2. Створення ієрархій папок для зберігання повідомлень
3. Залишення повідомлення на сервері після прочитання, щоб клієнти могли знову отримати доступ до повідомлень з іншого місця
4. Пошук у поштовій скриньці певного повідомлення для завантаження
5. Позначення повідомлень як прочитаних
6. Вибіркове завантаження лише частин повідомлень або вкладень
7. Перегляд заголовків повідомлень перед їх завантаженням

Алгоритм отримання клієнтом повідомлення з сервера IMAP4.

1. встановлення сеансу за допомогою протоколу керування передачею TCP через TCP-порт 143.
2. ідентифікація клієнта на сервері.
3. надсилання серверу серії команд IMAP4:

- **LIST:** Отримання списку папок у поштовій скриньці клієнта.
- **SELECT:** Вибір певної папки для доступу до своїх повідомлень.
- **FETCH (ВИБІРКА):** Отримання індивідуальних повідомлень.
- **LOGOUT (ВИХІД):** Завершення сеансу IMAP4.

IMAP4 підтримується сервером Microsoft Exchange Server. Оскільки клієнти IMAP4 можуть залишати прочитані повідомлення на сервері IMAP4, IMAP4 особливо корисний для користувачів мобільних пристроїв, які комутуються та отримують доступ до своєї пошти з кількох місць. Недоліком є те, що сервери IMAP4 вимагають більше ресурсів, ніж сервери POP3, оскільки користувачі зазвичай залишають велику кількість повідомлень на сервері.

Щоб виправити неполадки з віддаленими серверами IMAP4, використовуйте Telnet для підключення до порту 143. Потім виконйте різні команди IMAP4 і перевірте результати.

Переваги та недоліки IMAP

Переваги	Недоліки
1. *Синхронізація з кількома пристроями*: отримуйте доступ до однакових електронних листів і структури папок на різних пристроях. 2. *Організація на стороні сервера*: Широкі можливості для категоризації та керування електронними листами.	1. *Потрібен постійний доступ до Інтернету*: для перегляду нових повідомлень і папок. 2. *Споживає більше ресурсів сервера*: повідомлення та папки зберігаються на сервері.

3.4.3 Протокол SMTP (SMTP - Simple Mail Transfer Protocol - Простий протокол передачі пошти) - надсилає електронну пошту. Працює з протоколами POP3 та IMAP4 для отримання електронної пошти. Це стандартний протокол прикладного рівня для доставки електронної пошти через міжмережевий зв'язок TCP/IP, наприклад через Інтернет. Протокол передачі пошти SMTP описано в RFC 821.

Як працює SMTP

SMTP визначає формат повідомлень, що надсилаються між хостами TCP/IP в Інтернеті. SMTP використовує простий 7-бітний текст ASCII для надсилання повідомлень електронної пошти та надсилання SMTP-команд хостам-одержувачам. Багатоцільові розширення Інтернет-пошти (MIME) зазвичай використовуються для кодування багаточастинних двійкових файлів, включно з вкладеннями, у форму, яку може обробляти SMTP. SMTP надає механізм для пересилання електронної пошти з одного хоста TCP/IP на інший через Інтернет. Служби SMTP, що працюють на хості TCP/IP, спочатку встановлюють з'єднання з віддаленим хостом через TCP-порт 25 протоколу керування передачею. Потім сеанс SMTP ініціюється надсиланням команди helo та отриманням відповіді OK. Після цього комп'ютер-відправник використовує такі команди для надсилання повідомлень:

1. **Mail fr:** Ідентифікує хост-відправника хоста-одержувача
2. **Rcpt to:** Ідентифікує одержувача цільового повідомлення для хоста-одержувача за допомогою формату системи доменних імен (DNS) user@DNSdomain
3. **Data:** Ініціює надсилання тексту повідомлення у вигляді серії рядків тексту ASCII, що закінчуються окремою крапкою (.) у рядку
4. **Quit:** Закриває SMTP-з'єднання

Транспортування лише між SMTP-хостами. SMTP забезпечує передачу повідомлень тільки з одного SMTP-хоста на інший. Підтримку зберігання повідомлень у поштових скриньках забезпечують протоколи POP3 та IMAP4.

Впровадження SMTP на сервері Microsoft Exchange Server

Інсталюйте та настройте службу SMTP. Щоб виправити неполадки з віддаленими SMTP-серверами, використовуйте Telnet для підключення до порту 25, виконайте різні SMTP-команди і перевірте результати. Служба SMTP функціонує лише як агент доставки пошти SMTP (SMTP-хост) і не створює поштові скриньки користувачів.

***Переваги та недоліки* SMTP**

Переваги	Недоліки
1. *Ефективна маршрутизація*: використовує систему доменних імен (DNS) для оптимальної маршрутизації.	1. *Односпрямований*: виключно для надсилання електронних листів, а не для пошуку. 2. *Відсутність вбудованого шифрування*: функції

2. *Підтримує пакетні операції*: здатний надсилати кілька електронних листів за один сеанс.	безпеки часто потребують додаткових налаштувань.

*Додатково:*RFC 821 – простий протокол передачі пошти, серпень 1982 р.

3.4.4 Протокол Telnet. У наборі протоколів TCP/IP це протокол прикладного рівня, який надає можливості емуляції терміналу та дозволяє користувачам входити у віддалену мережу зі своїх комп'ютерів. Telnet описано в RFC 854. Це стандартний протокол TCP/IP для запуску програм на віддалених хостах. Telnet працює за принципом клієнт-сервер. Користувач, який використовує клієнтське програмне забезпечення telnet, може з командного рядка інтерактивно запускати програми командного рядка на віддаленому хості, на якому запущено службу або демон telnet. Користувач вводить інформацію в клієнті telnet, ця інформація обробляється на сервері telnet і її результат обробки повертається користувачеві. Термін «telnet» також відноситься до програмного забезпечення (клієнтського або серверного компонента), яке реалізує цей протокол на певній платформі чи системі. Синтаксис команди підключення до сервера Telnet з командного рядка такий: telnet hostname port. Наприклад, telnet avalon-rpg.com 23.
Віддалений вхід – це процес, під час якого користувачі можуть увійти на віддалений сайт, тобто комп'ютер, і користуватися послугами, доступними на віддаленому комп'ютері. За допомогою віддаленого входу користувач отримує можливість зрозуміти результат передачі результату обробки з віддаленого комп'ютера на локальний комп'ютер.(рис.4)

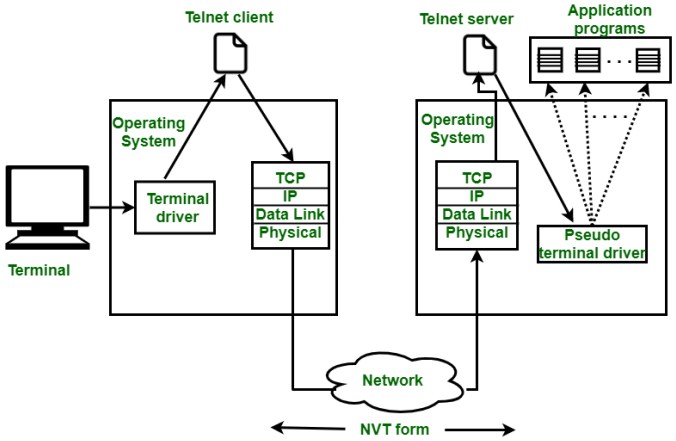


Рис.4. Віддалений вхід в TELNET [Introduction to TELNET].

Процедура віддаленого входу в систему

1. Коли користувач набирає текст на локальному комп'ютері, локальна операційна система приймає цей символ.
2. Локальний комп'ютер не інтерпретує символи, він відправляє їх клієнту TELNET.
3. Клієнт TELNET перетворює ці символи в універсальний набір символів під назвою Network Virtual Terminal (NVT) і передає їх в локальний стек протоколу TCP/IP.
4. Команди або текст, які мають форму NVT, передаються через Інтернет і надходять у стек TCP/IP на віддаленому комп'ютері.
5. Потім символи доставляються в операційну систему, а потім передаються на сервер TELNET.
6. Потім сервер TELNET змінює ці символи на символи, які можуть бути зрозумілі віддаленому комп'ютеру.
7. Віддалена операційна система отримує символи з драйвера псевдотерміналу, який є частиною програмного забезпечення, щоб символи сприймалися як такі, що надходять із терміналу.
8. Потім операційна система передає символ відповідній прикладній програмі.

Telnet не є безпечним. Облікова інформація (імена користувачів і паролі), що надається через telnet, не зашифрована і, отже, вразлива до крадіжки особистих даних. З часом використання Telnet зменшилося з міркувань безпеки, і тепер для безпечного віддаленого керування віддають перевагу альтернативам, таким як SSH. Telnet корисний для віддаленого адміністрування, діагностики мережі, навчальних цілей і взаємодії з застарілими системами.
Більш детально: RFC 854 - Telnet Protocol Specification
3.4.5 Протокол Secure Shell (SSH) – це протокол, який дозволяє клієнту віддалено видавати команди серверу. Успіх SSH зумовлено низькою вартістю впровадження технології та корисністю протоколу в організаціях. Протокол найбільше використовується для обміну даними всередині організацій: клієнти в основному отримують доступ до серверів організацій, до яких вони вже належать. Технологія SSH вводить автентифікацію серверів для клієнтів, які бажають до них отримати доступ. Клієнти повинні знати, чи використовувати захищений і автентифікований протокол (SSH) чи незахищений протокол (telnet) під час ініціювання підключення до сервера. Клієнти SSH відстежують, чи автентифікуються сервери. Якщо перше підключення до сервера не зазнає атаки, клієнт виявить, що сервер проходить автентифікацію, і майбутні підключення можна захистити. Цей підхід був ефективним, оскільки клієнти SSH зазвичай підключаються до серверів, з якими вони раніше спілкувалися.

Особливості SSH.

- **Шифрування:** між сервером і клієнтом відбувається обмін зашифрованими даними, що забезпечує конфіденційність і запобігає несанкціонованим атакам на систему.
- **Автентифікація:** для автентифікації SSH використовує пари відкритих і закритих ключів, які забезпечують більшу безпеку, ніж традиційна автентифікація за допомогою пароля.
- **Цілісність даних:** SSH забезпечує цілісність даних повідомлення, яким обмінюються під час спілкування.
- **Тунелювання:** за допомогою SSH ми можемо створювати безпечні тунелі для пересилання мережевих з'єднань по зашифрованих каналах.

Функції SSH.

Існує кілька функцій, які виконує функція SSH, ось деякі функції:

1. SSH забезпечує високий рівень безпеки, оскільки шифрує всі повідомлення зв'язку між клієнтом і сервером.
2. SSH забезпечує конфіденційність.
3. SSH дозволяє здійснювати віддалений вхід, отже, є кращою альтернативою TELNET.
4. SSH забезпечує безпечний протокол передачі файлів, що означає, що ми можемо безпечно передавати файли через Інтернет.
5. SSH підтримує тунелювання, що забезпечує більш безпечне з'єднання, обмін даними.

Методи, що використовуються в SSH.

В SSH використовуються три основні методи:

1. **Симетрична криптографія:** використовується стандарт шифрування даних AES .
2. **Асиметрична криптографія: :** використовується алгоритм RSA (Rivest–Shamir–Adleman) та алгоритм цифрового підпису.
3. **Хешування**: це процедура в криптографії, яка перетворює рядок змінної довжини на рядок фіксованої довжини, це фіксоване значення довжини називається хеш-значенням, яке генерується хеш-функцією.

Різниця між SSL і SSH.

SSL	SSH
SSL означає «рівень захищених сокетів».	SSH означає «безпечна оболонка».
SSL – це протокол безпеки.	SSH — мережевий криптографічний протокол.
Працює через порт 443.	Працює через порт 22.
Використовується в основному для встановлення безпечних з'єднань між веб-серверами та клієнтами (веб-браузерами).	Зазвичай використовується для безпечного зв'язку з віддаленим комп'ютером.
Автентифікація здійснюється за допомогою цифрового сертифіката X.509 (сертифікат SSL/TLS).	Автентифікація виконується за 3 кроки: *перевірка сервера, генерація ключа сеансу та автентифікація клієнта*.
SSL працює на основі сертифікатів SSL/TLS.	SSH працює на основі мережевих тунелів.
В основному використовується для захисту від атак "людина посередині" (MiTM) і крадіжки особистих даних.	Захищає від підробки DNS, IP-маршрутизації джерела, маніпулювання даними, перехоплення даних під час передачі, підробки IP-адрес тощо.

3.4.6 Протокол SNMP(Simple Network Management Protocol - простий протокол управління мережею) – це протокол прикладного рівня, який збирає інформацію про керування з таких пристроїв, як комп'ютери, маршрутизатори, комутатори, брандмауери та принтери. SNMP є надзвичайно надійним, простим, і досить потужним. Він розроблений для мереж, які використовують TCP/IP. SNMP є найпопулярнішим протоколом управління мережею, що використовується. Простий протокол управління мережею SNMP найчастіше використовується для збору статистичної та конфігураційної інформації про мережеві пристрої для моніторингу продуктивності та стану пристроїв у мережі в режимі реального часу. Протокол SNMP використовує станцію керування та агентів керування, які спілкуються з цією станцією. Станція розташована на вузлі, на якому запущена програма керування мережею. Агенти SNMP відстежують потрібні об'єкти у своєму середовищі, упаковують цю інформацію відповідним чином і відправляють її на станцію керування негайно або за запитом. Протокол працює з менеджером SNMP або програмним клієнтом, який надсилає SNMP get-запити на пристрої з підтримкою SNMP. Кожен пристрій із підтримкою SNMP має локального агента SNMP, який збирає дані про продуктивність пристрою та пересилає цю інформацію диспетчеру SNMP, щоб адміністратор міг отримати подання продуктивності та стану зверху вниз. Статистична інформація включає в себе число відправлених або отриманих пакетів або кадрів в секунду, число помилок в секунду, тощо. Інформація про конфігурацію включає IP-адресу інтерфейсу на пристрої, версію операційної системи, запущеної на пристрої, тощо. Системи керування використовуються для моніторингу працездатності мережі, виявлення помилок, виконання діагностики та формування звітів.

Що таке сервіс SNMP?

До особливостей сервісу SNMP можна віднести наступне:

1. Підтримка інтерфейсу прикладного програмування (API) Windows Sockets
2. Розширюваність (можна додавати інформаційні бази керування за допомогою сторонніх бібліотек динамічних посилань.)

3. Використання стандартного порту 161 протоколу користувацьких дейтаграм UDP.

Додаткова служба SNMP Trap Service дає змогу Windows перехоплювати події SNMP, наприклад умови виникнення помилок.

Як працює SNMP?

Мережа з SNMP, по суті, складається з двох компонентів, які працюють разом:

1. **Агенти SNMP** – це програми, які виконуються на мережевих керованих пристроях і збирають інформацію про конфігурацію та статистичні дані про роботу цих пристроїв, пов'язані з TCP/IP. Зібрана агентом інформація міститься в локальній базі даних - базі керуючих відомостей (MIB - Management Information Base). Бази даних MIB мають ієрархічну структуру та містять керовані об'єкти, які мають унікальні призначені ідентифікатори, видані Міжнародною організацією зі стандартизації (ISO). Змінні SNMP – це конкретні екземпляри керованих об'єктів у MIB. Агенти, що працюють на керованих пристроях, відстежують певні набори змінних SNMP і тимчасово зберігають цю інформацію до тих пір, поки агент не буде опитаний системою управління, після чого агент повідомляє значення збереженої інформації системі управління. Більшість мережевих пристроїв мають вбудоване програмне забезпечення агента SNMP і пов'язані з ним MIB.
2. **Програмне забезпечення SNMP (**Management Systems або Network Management System - NMS**)**, яке працює на робочій станції адміністратора і може відображати дані (топологію та компоненти мережі, трафік), зібрані з керованих пристроїв (комп'ютерів, маршрутизаторів, концентраторів і комутаторів), у зручній для користувача формі з використанням графічного інтерфейсу користувача (GUI). Програмне забезпечення SNMP (Management Systems) може сповіщати адміністратора про виникнення наприклад, помилок. Системи управління SNMP регулярно опитують керовані пристрої за допомогою повідомлень SNMP для отримання статистичної та конфігураційної інформації, а потім зберігають цю інформацію в центральній базі даних.

Повідомлення SNMP містять заголовок і корисне навантаження, які називаються одиницею даних протоколу (PDU). Заголовок містить інформацію про спільноту, на яку посилаються. **Спільнота** – це підмножина агентів, яка контролюється певною системою управління та встановлює примітивний рівень безпеки. Ім'я спільноти використовується для автентифікації, а зв'язок SNMP виконується за допомогою порту 161 протоколу користувацьких дейтаграм (UDP). Агент може мати декілька спільнот, станції можуть входити до кількох спільнот для одного агента, а станція може бути частиною спільнот, пов’язаних з різними агентами.

SNMP-повідомлення бувають 4-х типів (три видають станції управління і одне виконують агенти):

1. **Get message - Отримати повідомлення:** Видається системою управління агенту на керованому пристрої для зчитування значення певної змінної на пристрої.
2. **Getnext message - Отримати наступне повідомлення:** Видається системою управління для визначення того, які змінні SNMP підтримуються агентом, що працює на керованому пристрої, і для обходу ряду змінних для послідовного зчитування їх значень.
3. **Set message - Встановити повідомлення:** Видається системою управління агенту на керованому пристрої для запису значення певної змінної на пристрої.
4. **Trap message -Повідомлення про пастку:** Видається агентом, запущеним на керованому пристрої, у разі виникнення помилки або стану попередження. На додаток до пакетів для обробки запитів і переміщення пакетів у вузол і з нього, SNMP надсилає пастки (trap). Пастка — це спеціальний пакет, який надсилається від агента до станції керування, щоб повідомити адміністраторів про те, що сталося щось незвичайне.

Станція керування регулярно надсилає повідомлення getn, getn ext і set агенту SNMP на керованому пристрої, фактично періодично опитуючи агента щодо стану пристрою. Агент перевіряє ім'я спільноти в повідомленні, перевіряє IP-адресу або ім'я хоста системи управління SNMP, обробляє запит і відправляє результати в систему управління.(рис. 5)

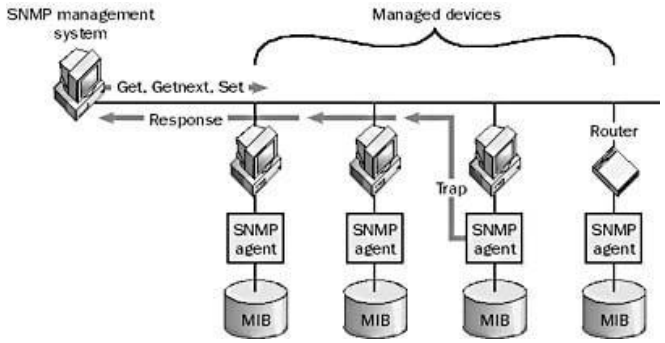


Рис.5 Простий протокол управління мережею (SNMP) [Simple Network Management Protocol (SNMP)]

SNMP версії 2 додає додаткові функції безпеки, може застосовуватися до мережевих архітектур, відмінних від TCP/IP, і підтримує додаткові типи даних.

SNMP 2 також визначає два додаткові типи повідомлень:

1. **Getbulk message -Повідомлення Getbulk:** Схожий на *getnext,* але дозволяє отримувати більші обсяги інформації в одному блоці даних
2. **Inform message -Інформ-повідомлення:** Дозволяє системам керування надсилати інформацію іншим системам керування за допомогою повідомлення, схожого на пастку

Впровадження TCP/IP корпорацією Майкрософт у Microsoft Windows включає агентів та MIB для збору інформації про умови TCP/IP та статистику.

4.Протоколи транспортного рівня (TCP, UDP).

Як відомо, протокол TCP/IP (Transmission Control Protocol/Internet Protocol - Протокол керування передачею /Інтернет-протокол) - є основним набором протоколів Інтернету, що забезпечує надійний зв'язок. Протокол TCP гарантує, що дані надсилаються надійно та в порядку. Протокол IP маршрутизує пакети даних до місця призначення на основі IP-адрес. TCP дбає про сегментацію даних, забезпечення надійної доставки та підтримання порядку сегментів. IP займається маршрутизацією та пересиланням пакетів, доставляючи їх від пристрою-відправника до пристрою-одержувача.

Транспортний рівень має 2 протоколи:

1. *Протокол керування передачею*

4.1 TCP (Transmission Control Protocol - протокол керування передачею транспортного рівня) - забезпечує надійність зв'язку між окремими хостами і *забезпечує надійні передачі з підтвердженням успішної доставки*. TCP використовує IP-протокол для доставки пакетів. TCP надає послугу, орієнтовану на з'єднання. TCP частіше використовується для більшості інтернет-сервісів в порівнянні з UDP. Хоча TCP гарантує надійну доставку та перевірку помилок, ця доставка має додаткові накладні витрати та затримку.

Розглянемо протокол TCP більш детально.

Протокол управління передачею TCP, служить основою для передачі даних через комп'ютерні мережі по всьому світу. TCP спеціалізується на забезпеченні надійної, впорядкованої та перевіреної безпомилкової доставки даних. TCP гарантує, що коли ви надсилаєте електронний лист, завантажуєте файл або транслюєте фільм, кожен пакет даних надходить у цілості послідовно та без помилок.

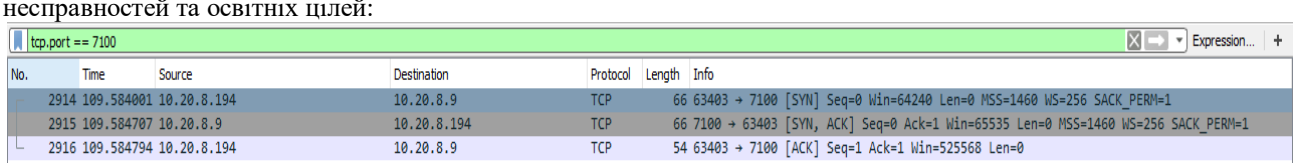
Ключові особливості TCP:

1. **Надійна передача**: Однією з визначальних рис TCP є його надійність. Коли надсилається пакет даних, відправник очікує підтвердження (ACK) від одержувача. Якщо АКК не приймається протягом заданого часу, пакет передається повторно.
2. **Замовлена доставка**: TCP нумерує кожен пакет даних, який він надсилає. Така нумерація дозволяє одержувачу повторно зібрати дані в правильному порядку, навіть якщо пакети надходять не послідовно.
3. **Перевірка помилок**: перед надсиланням і після отримання пакета TCP виконує перевірку помилок за допомогою контрольної суми. Якщо пакет приходить з помилкою, він відкидається, і спрацьовує автоматичний запит на повторну передачу.
4. **Контроль перевантажень**: TCP динамічно регулює швидкість передачі у відповідь на перевантаження мережі.

TCP працює шляхом встановлення надійного каналу між двома кінцевими точками мережі. Цей канал діє як двосторонній канал зв'язку, який забезпечує надійну передачу та прийом даних.

Тристороннє рукостискання TCP має першорядне значення для забезпечення надійного встановлення з'єднання між двома вузлами мережі. Синхронізуючи відправника та одержувача, воно гарантує, що обидві сторони готові обмінюватися даними, виділили достатньо ресурсів та узгодили параметри сесії. Цей протокол не тільки готує мережу до передачі даних, але й захищає від втрати та дублювання даних, гарантуючи, що кожен пакет досягає місця призначення в правильному порядку. Встановлення надійного з'єднання, що забезпечується рукостисканням, є основою надійної роботи мереж TCP/IP, що лежить в основі надійності та ефективності інтернет-комунікацій.

Тристороннє рукостискання TCP можливо спостерігати в режимі реального часу з допомогою інструменту мережевого аналізу Wireshark, який може захоплювати та відображати тристороннє рукостискання TCP у режимі реального часу, дозволяючи користувачам аналізувати мережевий трафік для усунення несправностей та освітніх цілей:



Алгоритм роботи протоколу TCP

Крок 1. Тристороннє рукостискання TCP. Перш ніж обмінюватися будь-якими даними, як відправник, так і одержувач повинні домовитися про спілкування. Цей процес відомий як «рукостискання TCP». Тристороннє рукостискання TCP встановлює з'єднання шляхом обміну повідомленнями SYN, SYN-ACK і ACK (рис. 6):

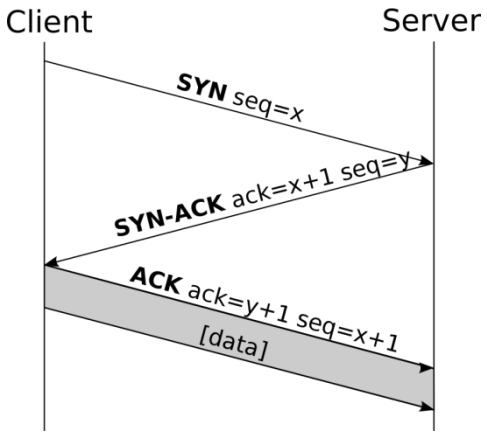


Рис. 6. Тристороннє рукостискання TCP.

1. **SYN:** Ініціюючий сокет надсилає пакет із встановленим прапорцем SYN (Synchronize Sequence Number) на приймальний сокет на першому кроці тристороннього рукостискання TCP. Прапорець SYN позначає початок нового запиту на з'єднання шляхом вказівки порядкового номера, який відправник буде використовувати при передачі пакетів даних.
2. **SYN-ACK**: Отримавши пакет SYN, приймальний сокет відповідає пакетом, у якому встановлено прапорці SYN і ACK (підтвердження).
3. **ACK**: Нарешті, ініціюючий сокет надсилає пакет ACK назад до приймального сокета, щоб підтвердити встановлене з'єднання. На цьому етапі з'єднання TCP вважається встановленим, і дані можуть почати протікати між двома сокетами.

   Якщо повідомлення ACK не надійшло, що вказує на те, що третій крок рукостискання не вдався, з'єднання не встановлюється. Хост-ініціатор (зазвичай це клієнт) може спробувати повторно надіслати пакет SYN, повторюючи процес рукостискання у разі тимчасових проблем із мережею або затримок.

Крок 2. Передача даних

Під час передачі даних кожен пакет даних послідовно нумерується, що допомагає приймальному сокету знову зібрати дані в правильному порядку. Пакети також визнаються одержувачем, що дає відправнику впевненість у тому, що дані досягли призначеного пункту призначення.

Крок 3.Управління потоком

TCP включає механізми керування потоком, такі як масштабування вікон, щоб запобігти перевантаженню швидких відправників повільними приймачами.

Крок 4. Завершення з'єднання

Як тільки передача даних завершується, з'єднання розривається за допомогою аналогічного процесу рукостискання з прапорцями FIN (Finish).

5. Структура сегмента TCP

TCP-сегмент — це стандартизований пакет даних, який передається між відправником і одержувачем під час сеансу зв'язку TCP. Розуміння його структури допомагає зрозуміти, як TCP досягає своїх цілей щодо надійності, впорядкування та цілісності даних. Сегмент TCP складається із заголовка та розділу даних.

Поля заголовка (рис. 7)

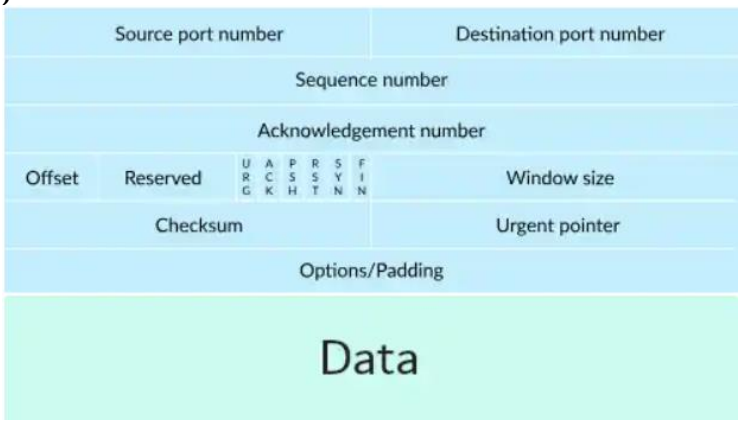


Рис. 7. Структура заголовка TCP.

Заголовок TCP містить різні поля, які керують різними процесами комунікації:

1. **Порт джерела та порт призначення (**source port number, destination port number**)**: ці 16-бітові поля ідентифікують кінцеві точки з'єднання.
2. **Порядковий номер (**sequence number**):** 32-бітне поле, яке визначає порядок надсилання даних.
3. **Номер підтвердження (**acknowledgement number**)**: якщо встановлено прапорець ACK, це 32-бітне поле містить значення наступного порядкового номера, який очікує відправник.

4. **Зміщення даних (**offset**)**: вказує довжину заголовка TCP, яка необхідна, оскільки заголовок може відрізнятися за довжиною.
5. **Прапорці**: набір прапорців, таких як SYN, ACK, PSH, RST, URG, FIN та інші, які вказують на призначення сегмента. Якщо пакет має ознаку PSH (PUSH), то це команда для отримувача передати дані з буфера мережевої карти чи ОС в застосунок;
6. **Розмір вікна (**window size**)**: визначає розмір вікна прийому, яке використовується для керування потоком.
7. **Контрольна сума (**checksum): 16-бітне поле для перевірки помилок заголовка та даних.
8. **Термінова вказівка (**urgent pointer**)**: якщо встановлено прапорець URG, це 16-бітне поле вказує зміщення, де закінчуються термінові дані.
9. **Розділ даних (data)**: містить фактичні дані, що передаються. Його розмір може варіюватися від 0 байт до максимуму, який визначається максимальним розміром сегмента (MSS), який обмовляється під час налаштування з'єднання.

Практичні поради для адміністраторів мережі

- **Регулярний моніторинг мережевого трафіку:** Використовуйте інструменти моніторингу, щоб виявити ранні ознаки перевантаження або несправності обладнання.
- **Налаштування брандмауерів та параметрів безпеки:** Переконайтеся, що конфігурації безпеки дозволяють рукостискання TCP, захищаючи їх від зловмисних дій.
- **Підтримка систем в актуальному стані:** Регулярні оновлення можуть вирішити відомі помилки, які можуть вплинути на зв'язок TCP.
- **Навчання користувачів:** Інформуйте користувачів про важливість безпечних мережевих методів для запобігання перевантаженню та потенційним DDoS-атакам.

Додатково:

RFC 793 – "Протокол керування передачею"
RFC 1185 – "Розширення TCP для високошвидкісних доріжок"

2. *Протокол без перевірки передачі*

4.2 UDP (User Datagram Protocol - протокол дейтаграм користувача транспортного рівня) *протокол без перевірки передачі*. UDP дозволяє надсилати пакети від одного хоста до іншого хоста в мережі TCP/IP. Однак UDP не підтверджує успішну передачу датаграми. Тобто це протокол транспортного рівня без встановлення з'єднання з використанням IP-протоколу міжмережного рівня для доставки пакетів. UDP використовується хостами для обміну невеликими обсягами даних.

Розглянемо протокол UDP більш детально.

Протокол UDP.

Перевага віддається протоколу UDP у ситуаціях, коли продуктивність у реальному часі має вирішальне значення, наприклад, під час онлайн-ігор, відео- або голосових дзвінків VoIP, а також під час проведення конференцій у реальному часі. Це тому, що UDP забезпечує швидшу передачу даних, навіть якщо є скидання деяких пакетів, а не затримку всього процесу. Крім того, оскільки перевірка помилок не здійснюється, UDP заощаджує пропускну здатність. Розглянемо структуру заголовка UDP.

Структура заголовка UDP (рис. 8).

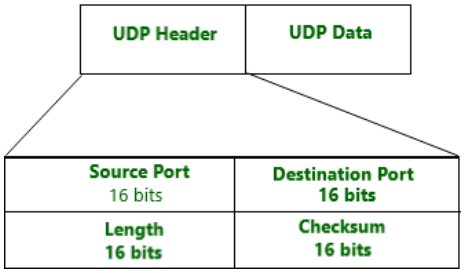


Рис. 8. Заголовок UDP. **[Приклади заголовка UDP]**

UDP має 8-байтовий фіксований і простий заголовок. Перші 8 Байт містять всю необхідну інформацію заголовка, а решта частина складається з даних. Поля з номерами портів UDP мають довжину 16 біт, тому діапазон номерів портів визначається від 0 до 65535, а номер порту 0 зарезервований. Номери портів допомагають розрізняти різні запити користувачів або процеси.

Заголовок UDP *складається з чотирьох ключових полів:*

1. **Порт джерела (16 біт):** Ідентифікує номер порту на стороні відправника. Допомагає одержувачу зрозуміти, в яку програму відправляти дані.
2. **Порт призначення (16 біт):** Ідентифікує номер порту на стороні приймача. Використовується для спрямування вхідних даних у відповідну програму.
3. **Довжина (16 біт):** Визначає загальну довжину заголовка UDP і даних. Мінімальна довжина становить 8 байт (розмір заголовка UDP), а максимальна – 65 535 байт.

4. **Контрольна сума (16 біт):** Забезпечує перевірку помилок для заголовка UDP та даних. Забезпечує цілісність даних під час передачі.

Коли пакет UDP надсилається з джерела до місця призначення:

2. **Порт джерела** та **порт призначення** гарантують, що пакет досягає потрібного місця призначення.
3. **Поле length(**довжина**)** вказує на розмір всього пакета, щоб приймальна система могла визначити, скільки даних потрібно зчитати.
4. **Контрольна сума** гарантує, що дані пакета не були пошкоджені під час передачі, що дозволяє виявляти помилки.

UDP використовується для таких мережних служб і функцій Microsoft Windows:

1. Розпізнавання імен хостів системи доменних імен (DNS) за допомогою пакетів UDP, надісланих серверам імен
2. Служби тривіального протоколу передачі файлів (TFTP)
3. Розпізнавання імен NetBIOS за допомогою трансляцій UDP підмережі, надісланих усім хостам у підмережі, або за допомогою одноадресних UDP-пакетів, надісланих безпосередньо на сервер Windows Internet Name Service.

Приклади використання UDP-заголовка для мережевих інженерів

1. **Програми в реальному часі:** UDP використовується в таких програмах, як VoIP і потокове відео в реальному часі, де низька затримка є критичною, а випадкова втрата пакетів прийнятна.
2. **DNS-запити:** DNS (система доменних імен) часто використовує UDP для швидкого перетворення доменних імен на IP-адреси. Невеликий розмір заголовка UDP добре підходить для таких типів запитів.
3. **Ігри та мультимедіа:** Онлайн-ігри та потокове мультимедіа часто використовує UDP для швидшої передачі даних, оскільки безз'єднувальний характер протоколу зменшує накладні витрати.

Примітка: заголовок UDP також містить дані корисного навантаження, які мають змінну довжину. Використання UDP – це протокол тунелювання, коли кінцева точка тунелю інкапсулює пакети іншого протоколу всередині датаграм UDP і передає їх іншій кінцевій точці тунелю, яка декапсулює датаграми UDP і пересилає оригінальні пакети, що містяться в корисному навантаженні.

*Додатково про UDP:*RFC 768 – User Datagram Protocol, 28 August 1980

5.Порти транспортного рівня.

IP-адреса призначається кожному кінцевому пристрою протоколом мережевого рівня. Але обмін даними відбувається не лише між кінцевими пристроями, а і між встановленими на них додатками. Лише IP-адреси недостатньо для отримання доступу до конкретного мережевого додатку. **Порти** також використовуються для ідентифікації додатків. Комбінація IP-адреси та порту називається сокетом. Крім власних протоколів, програми на рівні додатків часто мають фіксований номер порту для доступу до мережі. Орган присвоєння номерів в Інтернеті (**IANA** - Internet Assigned Numbers Authority), який розподіляє діапазони IP-адрес, також призначає порти мережевим програмам. Номери портів вказані в RFC 1700. Порт завжди записується після IP-адреси і відокремлюється від неї двокрапкою. Наприклад, 192.168.0.2:443.

Протокол TCP/IP	**Номер порта**
DNS	порт 53 DNS-сервера
HTTP	порт 80, порт 8080
HTTPS	порт 443
SMTP	порт 25
POP3	порт 110
Telnet	порт 23
SSH	порт 22
SNMP	порт 161
BGP	порт 179
IMAP	порт 143
BOOTP сервер для отримання запитів клієнта	порт 67
BOOTP клієнт для отримання відповідей сервера	порт 68
FTP для керування	порт 21
FTP для даних	порт 20
TFTP	порт 69

Приклад 1. DNS дозволяє не запам'ятовувати IP - комп'ютер сам відправляє запит на 53 порт DNS-сервера, отриманий від інтернет-провайдера. DNS-сервер дає комп'ютеру відповідь. Потім комп'ютер встановлює з'єднання з веб-сервером отриманого IP, який слухає на порту 80 для протоколу HTTP і на порту 443 для HTTPS. Порт не відображається в адресному рядку, але використовується за замовчуванням.

Приклад 2. Поштові додатки, які обмінюються даними по протоколу SMTP, використовують порт 25, при використанні протоколу POP3 - порт 110.

6. Протоколи мережевого рівня (IP, DHCP, ICMP, ARP, RARP).

До протоколів мережевого рівня належать: Інтернет-протокол (**IP**), протокол визначення адрес (**ARP**), Internet Control Message Protocol **ICMP**(для сповіщення про помилки), Internet Group Management Protocol (**IGMP**)(керує груповою (multicast) передачею даних в мережах), Open Shortest Path First (**OSPF**)(протокол динамічної маршрутизації для знаходження найкращого шляху в мережі) та інші.

Розглянемо детальніше протоколи мережевого рівня.

6.1 Протокол IP (Інтернет-протокол): обробляє адресацію та маршрутизацію пакетів даних. Інтернет-протокол— це протокол міжмережевого рівня TCP/IP для адресації та маршрутизації пакетів даних між вузлами в мережі TCP/IP. Інтернет-протокол– це безз'єднувальний протокол, який забезпечує доставку за допомогою служб комутації пакетів. Протокол IP не гарантує доставку даних. Гарантує доставку даних та надсилання підтверджень протокол вищого транспортного рівня - TCP.

Зауваження. Структура IP-пакета описана в лекції №3.

Найважливіші поля заголовка включають:

1. **IP-адреса джерела:** IP-адреса хоста, що передає пакет.
2. **IP-адреса призначення:** IP-адреса хоста, на який надсилається пакет, адреса багатоадресної групи або широкомовна IP-адреса 255.255.255.
3. **Контрольна сума заголовка:** Математичне обчислення для перевірки того, що пакет був отриманий неушкодженим.
4. **Час життя (TTL) (від 2 до 20 хвилин) :** Кількість переходів маршрутизатора, яку пакет може зробити перед видаленням.
5. **Зміщення фрагмента:** Позиція фрагмента, якщо вихідний IP-пакет був фрагментований (наприклад, маршрутизатором). Ця інформація дозволяє реконструювати вихідний пакет.

IP-пакети маршрутизуються наступним чином:

1. Якщо IP визначає, що IP-адреса призначення є локальною, він передає пакет безпосередньо хосту призначення.
2. Якщо IP визначає, що IP-адреса призначення є віддаленою адресою, він перевіряє локальну таблицю маршрутизації для маршруту до хоста призначення. Якщо маршрут знайдений, він використовується; якщо маршрут не знайдено, IP пересилає пакет на шлюз за замовчуванням. У будь-якому випадку пакет, призначений для віддаленої адреси, зазвичай відправляється на маршрутизатор.
3. На маршрутизаторі TTL зменшується на 1 або більше (залежно від перевантаження мережі), і пакет може бути роздроблений на менші пакети, якщо це необхідно. Потім маршрутизатор визначає, чи пересилати пакет на один з локальних мережевих інтерфейсів маршрутизатора або на інший маршрутизатор. Цей процес повторюється до тих пір, поки пакет не прибуде на хост призначення або його TTL не зменшиться до 0 (нуль) і не буде відкинутий маршрутизатором.

Інтернет-протокол працює з пртоколами транспортного рівня TCP і UDP.

Додаткова інформація про інтернет-протокол: RFC 791 – Internet Protocol (DARPA Internet Program), September 1981

6.2 Протокол DHCP (**Dynamic Host Configuration Protocol** - протокол динамічної настройки хоста) - це мережевий протокол, використовуваний для динамічного призначення пристроям у мережі IP-адрес і інших параметрів конфігурації мережі. Тобто DHCP автоматично призначає IP-адреси пристроям у мережі і зменшує конфігурацію вручну та конфлікти IP-адрес.

Налаштувати конфігурацію IP на пристрій можна двома способами: вручну і динамічно. При ручному способі ми вручну призначаємо пристроям конфігурацію IP. У динамічному методі налаштовуємо сервіс DHCP. Сервіс DHCP автоматично забезпечує конфігурацію IP для пристроїв. Якщо розмір мережі невеликий, ви можете використовувати ручний метод для налаштування та керування IP-адресами на всіх пристроях. Але, якщо розмір вашої мережі великий, ручний метод обтяжливий. Протокол DHCP широко використовується в сучасних IP-мережах, включаючи локальні мережі (LAN), глобальні мережі (WAN) та Інтернет.

Протокол DHCP працює за моделлю клієнт-сервер, тобто клієнти DHCP запитують IP-адреси й інші параметри конфігурації у DHCP-сервера, який відповідає на запитувану інформацію. DHCP-сервер управляє пулом IP-адрес, і коли клієнт запитує адресу, сервер орендує доступну адресу з пулу адрес на визначений період часу. Це дозволяє ефективно управляти розподілом IP-адрес і допомагає запобігти конфліктам IP-адрес.

Отже, DHCP – це простий спосіб керування конфігурацією IP. Це дозволяє пристроям динамічно отримувати конфігурацію IP. Він визначений у RFC 2131 і 2939. DHCP працює в моделі сервер/клієнт. DHCP -сервер пропонує та доставляє конфігурації IP. DHCP-клієнти запитують і отримують свої конфігурації IP. DHCP працює на всіх типах мереж. DHCP можна використовувати в домашній мережі або в офісі чи бізнес-мережі. З допомогою сервісу DHCP наступні параметри мережі налаштовуються автоматично:

1. IP адреса
2. Маска підмережі
3. Шлюз за замовчуванням

Приклад.

Нехай компанія, що має мережу з понад 2000 пристроїв придбала нове інтернет-з'єднання і призначила адміністратора мережі налаштувати всі пристрої для використання цього нового інтернет-з'єднання. Якщо адміністратор мережі вручну налаштовував IP-адреси на всіх пристроях, то він повинен налаштувати всі

пристрої заново. Однак, якщо використовувати DHCP, то адміністратор мережі змінить конфігурацію тільки на DHCP-сервері. Коли клієнти будуть перезавантажуватися або оновлювати свою конфігурацію IP, то вони автоматично отримають нову конфігурацію IP від DHCP-сервера.

До протоколів конфігурації хоста належать:

1. **DHCPv4**: Dynamic Host Configuration Protocol version 4 - протокол динамічної конфігурації хосту для IPv4. Сервер DHCPv4 динамічно призначає інформацію про адреси IPv4 клієнтам DHCPv4 під час запуску та дозволяє повторно використовувати адреси, коли вони більше не потрібні.
2. **DHCPv6**: протокол динамічної конфігурації хоста для IPv6. DHCPv6 схожий на DHCPv4. Сервер DHCPv6 динамічно призначає інформацію про адресу IPv6 клієнтам DHCPv6 під час запуску.
3. **SLAAC**: Stateless address autoconfiguration - автоконфігурація адреси без збереження стану. SLAAC дозволяє пристрою отримувати інформацію про адресу IPv6 без використання сервера DHCPv6.

Типи розподілу DHCP:

Статичний розподіл

У цьому способі адміністратор налаштовує таблицю розподілу на DHCP-сервері. У цій таблиці адміністратор додає MAC-адреси всіх клієнтів і призначає кожному клієнту конфігурацію IP.

Сервер DHCP використовує цю таблицю для надання конфігурацій IP. Коли клієнт запитує конфігурацію IP, DHCP-сервер знаходить MAC-адресу клієнта в таблиці. Якщо він знаходить запис для клієнта, він надає клієнту конфігурацію IP.

Динамічний розподіл

У цьому методі адміністратор налаштовує діапазон IP-адрес на DHCP-сервері. DHCP-сервер призначає конфігурацію IP з налаштованого діапазону кожному клієнту, який запитує конфігурацію IP.

У цьому методі DHCP пропонує конфігурацію IP лише на певний час, відомий як *оренда*. Конфігурація IP залишається чинною до закінчення терміну оренди. Після закінчення терміну оренди клієнт повинен знову отримати нову конфігурацію IP від сервера.

Автоматичний розподіл

Цей метод схожий на динамічний. У цьому методі адміністратор налаштовує діапазон IP-адрес на DHCP-сервері, а DHCP-сервер призначає конфігурацію IP з налаштованого діапазону кожному клієнту, який запитує конфігурацію IP.

У цьому методі DHCP-сервер призначає конфігурацію IP на постійній основі. Для цього DHCP-сервер встановлює тривалість оренди на нескінченність. В результаті, як тільки DHCP-сервер вибирає конфігурацію IP з пулу і призначає конфігурацію IP клієнту, конфігурація IP залишається з цим же клієнтом на невизначений термін.

Як працює DHCP?

Нехай є мережа підключених пристроїв і DHCP-сервер, який керує IP-адресами. Підключений до мережі новий пристрій ще не має IP-адреси.

1) новий пристрій виконує пошук IP-адреси. Він надсилає широкомовне (broadcast) сповіщення на всі пристрої мережі (з IP-адресою отримувача, яка складається з 32 одиниць в двійковій системі числення та MAC-адресою отримувача, яка складається з 48=12x4 одиниць в двійковій системі числення). Цей запит надійде на всі пристрої, і DHCP-сервер також його отримає.

2) DHCP-сервер отримує це сповіщення, вибирає з діапазону IP-адресу, яка ще не використовується і надсилає цю IP-адресу на щойно підключений пристрій.

3) IP-адреса надходить на новий пристрій. Пристрій приймає її і надсилає на DHCP-сервер запит на її використання.

4) DHCP-сервер отримує запит, що приймається з нового пристрою. DHCP-сервер надає IP-адресу цьому новому пристрою, маску під мережі, IP-адресу шлюзу за замовчуванням та список IP-адрес DNS-серверів. DHCP-сервер створює запис з даними про новий підключений пристрій, який зазвичай включає MAC-адресу нового підключеного пристрою, призначену IP-адресу та дату закінчення терміну дії цієї IP-адреси. DHCP здає IP-адресу в оренду лише на обмежений час. Після закінчення терміну IP-адреса повертається до пулу IP-адрес доступних IP-адрес і може бути знову призначена іншому новому пристрою. Порт UDP для зв'язку зазвичай є портом 68 для клієнтів і портом 67 для серверів.

В роботі DHCP-сервера можуть бути деякі відмінності, в залежності від постачальників мережевого обладнання.

Примітка. Якщо новий пристрій в мережі визначає, що надіслана йому DHCP-сервером IP-адреса не унікальна, він надсилає в мережу широкомовне сповіщення відмови від щойно наданої йому IP-адреси і далі повторюються кроки 1)-4) отримання нової унікальної IP-адреси.

Як працює DHCP: пояснення на прикладі

Коли хосту (DHCP-клієнту) потрібна конфігурація IP, він підключається до DHCP-сервера та запитує конфігурацію IP. DHCP-сервер містить кілька попередньо налаштованих конфігурацій IP. Коли він отримує DHCP-запит від DHCP-клієнта, він надає клієнту конфігурацію IP з усіх доступних конфігурацій IP.

Цей процес складається з чотирьох етапів: «Виявлення», «Пропозиція», «Запит» і «Підтвердження».

На наступному зображенні показані всі чотири кроки зв'язку DHCP.

Виявлення DHCP (DHCPDISCOVER): Це широкомовне повідомлення, яке надсилається DHCP-клієнт для виявлення доступних DHCP-серверів локальної мережі. Він відправляється у вигляді широкомовного кадру Ethernet рівня 2 з використанням MAC-адреси призначення FF:FF:FF:FF:FF:FF.
Пропозиція DHCP (DHCPOFFER): Це одноадресное повідомлення, відправлене DHCP-сервером у відповідь на повідомлення DHCP Discover. Він надає DHCP-клієнту IP-адресу й інші параметри конфігурації. Він відправляється у вигляді одноадресного кадру Ethernet рівня 2, адресованого MAC-адресою DHCP-клієнта.
Запит DHCP (DHCPREQUEST): Це широкомовне або одноадресное повідомлення, яке надсилається DHCP-клієнтом для офіційного запиту пропонованого IP-адреси з визначеного DHCP-сервера. Він може бути відправлений у вигляді широкомовного кадру Ethernet рівня 2 або одноадресного кадру, в залежності від ситуації.
Підтвердження DHCP (DHCPACK): Це одноадресное повідомлення, надіслане сервером DHCP для підтвердження запиту клієнта і надання IP-адреси та інших параметрів конфігурації. Він відправляється у вигляді одноадресного кадру Ethernet рівня 2, адресованого MAC-адресою DHCP-клієнта.

У наведеній нижче таблиці описано повідомлення, які використовуються на кожному етапі.

Discover	Клієнт DHCP передає це повідомлення для пошуку DHCP-сервера.
Offer	Сервер DHCP транслює це повідомлення для передачі конфігурації IP в оренду клієнту DHCP.
Request	Клієнт DHCP використовує це повідомлення, щоб повідомити DHCP-сервер, чи приймає він запропоновану конфігурацію IP чи ні.
Acknowledgment	Сервер DHCP використовує це повідомлення, щоб підтвердити клієнту DHCP, що він може використовувати запропоновану конфігурацію IP.

Кожна буква написання DORA відповідає першій букві кожного кроку: D (Discover), O (Offer), R (Request) і A (Acknowledgment - підтвердження).

Переваги і недоліки DHCP.

Переваги DHCP	Недоліки DHCP
1. *Точна конфігурація IP*: параметри конфігурації IP-адреси мають бути точними (адреса типу «192.168.XXX.XXX» повинна бути визначена як помилка). Крім того, помилки при ручному введенні адрес часто важко усунути, і цей ризик можна мінімізувати, використовуючи DHCP-сервер. 2. *Зменшує конфлікти IP-адрес*: кожен підключений пристрій повинен мати IP-адресу. Однак кожну адресу можна використовувати лише один раз, а дублювання адрес веде до конфліктів, коли один або обидва пристрої не можуть бути підключені. Це може статися, коли адреси призначаються вручну, особливо якщо є велика кількість кінцевих точок, які регулярно підключаються, наприклад, мобільні пристрої. Використання DHCP гарантує, що кожна адреса використовується лише один раз. 3. *Автоматизує керування IP-адресами*: без DHCP адміністраторам мережі доводилося б вручну призначати та відкликати адреси. Відстежувати, який пристрій має яку адресу, може бути марним, тому що майже неможливо зрозуміти, коли пристрою потрібен доступ до мережі, а коли він повинен вийти з неї. DHCP дозволяє автоматизувати та централізувати його, тому мережеві професіонали можуть керувати всіма локаціями з одного місця. 4. *Ефективно керує змінами адрес*: Використання DHCP спрощує зміну адрес, областей або кінцевих точок. Наприклад, організація може	1. *DHCP створює більше широкомовного трафіку в мережі,* оскільки кожен клієнтський пристрій надсилає запит на IP-адресу. Це може вплинути на загальну продуктивність мережі, особливо в невеликих мережах. 2. *DHCP менш безпечний, ніж статична IP-адреса,* оскільки зловмисникам легше підробити запити DHCP і отримати IP-адресу в мережі. 3. *DHCP менш надійний, ніж статична IP-адресація,* оскільки це залежить від доступності та правильного функціонування сервера DHCP, який призначає IP-адреси клієнтам.

захотіти змінити схему IP-адресації з однієї області на іншу. Сервер DHCP налаштовано з новою інформацією, яка поширюватиметься на нову кінцеву точку. Аналогічно, якщо ви оновлюєте та замінюєте мережеве обладнання, вам не потрібна конфігурація мережі.	

6.3 Протокол ICMP (Internet Control Message Protocol) — це протокол мережевого рівня TCP/IP, який використовується маршрутизаторами та хостами TCP/IP для побудови та підтримки таблиць маршрутизації, регулювання швидкості потоку даних, а також повідомлення про помилки та керуючих повідомлень для мережевого зв'язку TCP/IP. Протокол ICMP описано в RFC 792.
ICMP використовує дейтаграми безз'єднуваного Інтернет-протоколу (IP) різних типів для передачі керуючих повідомлень між хостами та маршрутизаторами в мережі TCP/IP.
Найбільш поширені ICMP-пакети:

1. **Echo Reply (**ICMP type 0 - Відповідь Echo (ICMP тип 0**)) :** Команда **ping** використовує цей тип пакета для перевірки з'єднання TCP/IP.
2. **Destination Unreachable (**ICMP type 3 - Пункт призначення недосяжний (ICMP тип 3**):** Вказує на те, що неможливо зв'язатися з мережею призначення, хостом або портом.
3. **Source Quench (**ICMP type 4 - Гасіння джерела (ICMP тип 4**):** Маршрутизатори надсилають цей тип пакета, коли вони не можуть обробляти IP-трафік так швидко, як він надсилається. Повідомлення Source Quench по суті означає: «Slow down! - Сповільніться!». Хост Microsoft Windows може відповісти на повідомлення Source Quench, сповільнюючи швидкість передачі даних.
4. **Redirect Message (**ICMP type 5 - Перенаправлення повідомлення (ICMP тип 5**):** Використовується для перенаправлення хоста на інший мережевий шлях. Це повідомлення, по суті, повідомляє маршрутизатору перевизначити запис у його внутрішній таблиці маршрутизації для цього пакета.
5. **Echo Request (**ICMP type 8 - Exo-запит (ICMP тип 8**):** Команда ping використовує цей тип пакета для перевірки з'єднання TCP/IP.
6. **Time Exceeded (**ICMP type 11 - Перевищено час (ICMP тип 11**):** Вказує на те, що час життя (TTL) пакету було перевищено через занадто велику кількість стрибків (hops). Команда **tracert** використовує це повідомлення для перевірки серії маршрутизаторів між локальним і віддаленим хостами.

Типи ICMP

ICMP Type	ICMP Code	Description
0	0	Echo Reply (used by ping)
3	0	Destination Network Unreachable
3	1	Destination Host Unreachable
3	3	Destination Port Unreachable
8	0	Echo Request (used by ping)
11	0	TTL Expired (used by traceroute)

Джерело повідомлень Quench
Коли комп'ютер під керуванням Windows використовується як маршрутизатор, він не надсилає повідомлення Source Quench хостам, що передають, якщо дані надходять надто швидко. Замість цього він просто відкидає пакети, які не можуть бути буферизовані та оброблені.
ICMP та атака типу «відмова в обслуговуванні»
Перенаправлення ICMP можуть змінювати таблицю маршрутизації маршрутизатора, тому іноді хакери намагаються підірвати маршрутизатори, видаючи підроблені перенаправлення ICMP, щоб виконати атаку типу «відмова в обслуговуванні».
ICMP редиректи зазвичай надсилаються роутерами лише за наявності всіх наступних умов:

1. Роутер налаштований на генерацію ICMP перенаправлень.
2. Інтерфейс вхідного маршрутизатора для пакета такий самий, як інтерфейс вихідного маршрутизатора.
3. Підмережа IP-адреси джерела ідентична IP-адресі наступного стрибка.
4. Дейтаграма IP не маршрутизується за допомогою джерела.

Більш детальні відомості про ICMP:
RFC 792 – Internet Control Message Protocol

6.4 Протоколи визначення адрес ARP і RARP.

6.4.1 Протокол ARP(Address Resolution Protocol - протокол визначення адрес) має вирішальне значення: він ставить у відповідність IP-адреси фізичним MAC-адресам, гарантуючи, що пакети даних знайдуть шлях до правильного місця призначення (забезпечує правильну маршрутизацію пакетів даних і доставку їх адресатам по локальній мережі). У пакеті протоколів TCP/IP це протокол для зіставлення (4-байтовим) IP-адресам (6-байтових) адрес каналів даних. IP-адреси є мережевими; адреси каналів передачі даних є апаратними, фізичними та пов’язані з комп’ютером. Це відображення є життєво важливим для транспортування пакетів даних від пристрою-джерела до пристрою призначення в межах тієї ж мережі. Без ARP пристрої не зможуть ефективно обмінюватися даними, що дасть збій в роботі мережі. *Команда arp доступна, лише якщо на ПК встановлено TCP/IP.*

ARP працює по-різному залежно від того, який протокол рівня доступу до мережі працює в даній мережі - протокол локальної мережі (Ethernet, Token Ring, FDDI) з можливістю широкомовного доступу одночасно до всіх вузлів мережі, або ж протокол глобальної мережі (X.25, Frame Relay) без широкомовного доступу. Вузол, якому потрібно визначити за IP-адресою локальну MAC- адресу, формує ARP-запит, вкладає його в кадр протоколу рівня доступу до мережі, вказуючи в ньому відому IP-адресу, і розсилає запит широкомовно. Всі вузли локальної мережі отримують ARP-запит і порівнюють вказану там IP-адресу з власною IP-адресою. У разі їх співпадання вузол-отримувач ARP-запиту формує ARP-відповідь, в якій вказує свою IP-адресу і свою локальну MAC- адресу і відправляє ARP-відповідь на отриману IP-адресу вузла-відправника ARP-запиту. ARP-запити і ARP-відповіді мають однаковий формат пакета.

ARP — це телекомунікаційний протокол, який використовується для перетворення адрес міжмережевого рівня(рівень 2) на адреси рівня доступу до мережі (рівня 1) в моделі TCP/IP, що є критично важливим етапом у процесі доставки пакетів в локальних мережах LAN. Тобто протокол ARP працює між рівнем 2 і рівнем 1.

ARP працює в основному в рамках протоколу IPv4, який сьогодні є основою для більшості інтернет-комунікацій. Коли пристрій отримує команду зв'язатися з іншим пристроєм у тій самій мережі, він спочатку перевіряє свій кеш ARP, збережений список відповідностей IP-адрес MAC-адресам. Якщо потрібної IP-адреси немає в кеші, пристрій надсилає ARP-запит на всі пристрої в мережі (broadcast), запитуючи MAC-адресу, пов'язану з відповідною IP-адресою. Пристрій із відповідною IP-адресою надсилає у відповідь свою MAC-адресу. Отже, ARP є критично важливою утилітою мережевих комунікацій, яка усуває розрив між логічним мережевим рівнем і фізичним канальним рівнем в моделі OSI.

Протокол ARP має важливе значення для безперебійної роботи локальних мереж, полегшуючи перетворення логічних адрес на фізичні. Механізм кешу ARP мінімізує частоту запитів, зменшуючи накладні витрати та прискорюючи зв'язок між пристроями.

Чому команда arp важлива?

Розуміння кешу ARP і керування ним має вирішальне значення для усунення несправностей мережі та оптимізації. Неправильний або застарілий кеш ARP може призвести до багатьох проблем, таких як недоступність мережевих пристроїв, повторювані IP-адреси або низька продуктивність мережі. Команда arp надає механізм для прямої взаємодії з кеш-пам'яттю ARP, що дозволяє швидко діагностувати та виправляти помилки. Відображає та змінює записи в кеші протоколу розпізнавання адрес (ARP). *Кеш ARP містить одну або кілька таблиць, які використовуються для зберігання IP-адрес і їх розпізнаних фізичних адрес Ethernet або Token Ring. Існує окрема таблиця для кожного мережевого адаптера Ethernet або Token Ring, встановленого на вашому комп'ютері.* Якщо використовувати без параметрів, arp відображає довідкову інформацію. Команда **arp** дозволяє додавати, видаляти або переглядати записи кешу ARP безпосередньо з командного рядка.

Динамічні та статичні записи

У таблиці кешу **arp** є два типи записів - динамічні та статичні.

Динамічний запис — це пара IP- адреса – MAC- адреса, яку ваш комп'ютер отримав під час останнього зв'язку з цим пристроєм.

Статичний запис– це запис, який було введено вручну (або операційною системою) до кешу. Статичні записи залишаються в кеші необмежений час, поки їх не буде видалено. Динамічні записи залишатимуться в кеші, якщо вони не використовувалися останнім часом і час очікування кешу ARP минув.

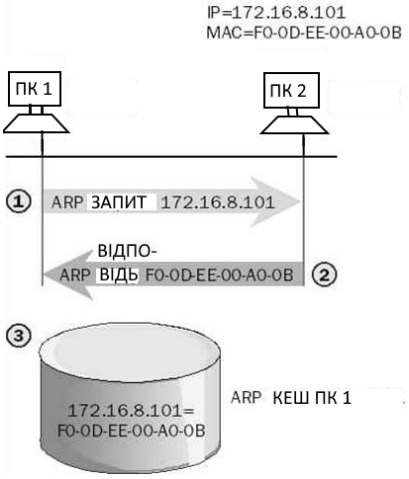


Рис. 9. Схема роботи протоколу ARP (ПК – персональний комп'ютер).

Робота протоколу ARP (рис. 9):

1. **ARP-запит**: Коли пристрій 1 повинен зв'язатися з іншим пристроєм 2, і не має MAC-адреси пристрою 2 у своєму ARP-кеші, він генерує ARP-запит. Цей запит у виді повідомлення широко розсилається на всі пристрої локальної мережі із запитанням: «Хто має IP-адресу X? Надішліть мені свою MAC-адресу."
2. **Відповідь ARP**: Пристрій 2, розпізнавши свою IP-адресу в запиті ARP, надсилає відповідь ARP. Ця відповідь, що містить MAC-адресу пристрою 2, надсилається безпосередньо на пристрій 1. Цей прямий зв'язок встановлює точку контакту між двома пристроями.

3. **Оновлення кешу ARP**: отримавши відповідь ARP, пристрій 1 оновлює свій кеш ARP новим відображенням IP-MAC. Цей кеш є важливим компонентом процесу ARP, що дозволяє пристроям зберігати та швидко отримувати відомі відображення адрес, зменшуючи потребу в майбутніх запитах ARP.
4. **Передача даних**: Оскільки MAC-адреса пристрою 2 тепер відома, пристрій 1 може перейти до інкапсуляції пакета даних з правильною MAC-адресою призначення. Потім пакет передається через мережу, керуючись фізичною адресацією канального рівня, щоб дістатися до пристрою 2.

Динамічний характер

Протокол ARP легко адаптується до змін у мережі. Пристрої можуть приєднуватися або виходити без порушення загального процесу зв'язку, а кеші ARP періодично оновлюються, щоб відображати поточний стан мережі. Ця гнучкість є свідченням надійності ARP та його вирішальної ролі в підтримці неперервної роботи локальних мереж. Використання широкомовних повідомлень для ARP-запитів гарантує, що всі пристрої в локальній мережі можуть слухати і відповідати в разі необхідності, а прямі відповіді запобігають непотрібному перевантаженню мережі. Таким чином, протокол ARP є ефективним, масштабованим та адаптивним. Його здатність динамічно вирішувати відображення адрес лежить в основі безперебійного зв'язку в мережах, постійно підтверджуючи важливість ARP у сфері комп'ютерних мереж.

Практичні приклади використання ARP .

1. *Підключення до Інтернету*

Одним із найпоширеніших випадків використання ARP є спроба пристрою підключитися до Інтернету. Перш ніж надсилати дані у зовнішню мережу, пристрій має пройти через маршрутизатор-шлюз локальної мережі. ARP використовується для перетворення IP-адреси шлюзового маршрутизатора на його MAC-адресу, забезпечуючи правильне спрямування пакетів даних через локальну мережу для доступу до Інтернету.

2. *Обмін даними між пристроями в локальних мережах*

У локальній мережі пристроям часто потрібно обмінюватися даними один з одним, будь то обмін файлами, мережеві ігри або програмне забезпечення для спільної роботи. ARP сприяє цьому, дозволяючи пристроям виявляти MAC-адреси один одного, забезпечуючи прямий обмін пакетами в мережі без необхідності маршрутизації через зовнішню мережу.

Синтаксис arp:

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

Команда arp **/?** відображає список доступних параметрів:

Параметри:

-a Відображає поточні записи ARP шляхом опитування даних поточного протоколу. Якщо вказано inet_addr, IP і відображаються фізичні адреси лише для вказаного комп'ютера. Якщо більше однієї мережі інтерфейс використовує ARP, відображаються записи для кожної таблиці ARP.
-g Те саме, що -a.
-v Відображає поточні записи ARP у докладному режимі.
inet_addr Вказує інтернет-адресу.
-N if_addr Відображає записи ARP для мережевого інтерфейсу, визначеного в if_addr.
-d Видаляє хост, указаний inet_addr. inet_addr може бути зі знаком *, щоб видалити всі хости
-s Додає хост і пов'язує адресу Інтернету inet_addr з фізичною адресою eth_addr. Фізична адреса подається як 6 шістнадцяткових байтів, розділених дефісами. Вхід є постійним.
eth_addr Вказує фізичну адресу.
if_addr Якщо є, це вказує Інтернет-адресу інтерфейсу, в якому слід змінити таблицю трансляції адрес. Якщо його немає, використовуватиметься перший застосовний інтерфейс.

Приклад:

> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Додає статичний запис.
> arp -a .... Відображає таблицю arp.

ARP не є безпечним і не захищений від атак підробки.

ARP легко інтегрується з сучасними мережевими практиками та технологіями. Віртуальні локальні мережі (VLAN), віртуальні приватні мережі (VPN) і середовища хмарних обчислень покладаються на ARP для ефективного мережевого зв'язку. NDP (Neighbor Discovery Protocol) – аналог ARP для IPv6.

Постійна актуальність ARP

Хоч впровадження IPv6 зростає, ARP залишається актуальним у мережах з IPv4. Двостековий підхід, коли пристрої підтримують як IPv4, так і IPv6, гарантує використання ARP у майбутньому. Принципи, що лежать в основі ARP, залишаються критично важливими для проектування та експлуатації як поточних, так і майбутніх мережевих архітектур.

6.4.2 Протокол RARP (Reverse Address Resolution Protocol - реверсивний ARP**) –** протокол визначення IP-адреси за відомою локальною MAC-адресою. Він використовується під час входу бездискових робочих станцій у мережу, яким не відома в початковий момент їх IP-адреса, але відома адреса свого мережного адаптера (мережевий адаптер - це пристрій, що забезпечує інтерфейс комп'ютера та інших пристроїв для з'єднання з мережею). RARP описано в RFC903. Не всі реалізації TCP/IP підтримують RARP (BOOTP – RFC 1542 – надає IP натомість).

7.Безпека мережних протоколів. Вразливості найпоширеніших протоколів стеку TCP/IP мереж.

Проте, дані технології мають низку вразливостей, зумовлених насамперед алгоритмами протоколів різного рівня взаємодії. Великий клас мережевих атак включав уразливості протоколу TCP/IP - вони були спрямовані на недоліки протоколів TCP/IP або помилки в реалізації TCP/IP, а не на вразливості безпеки в окремих службах TCP/IP. Більшість із них призвели до проблем з відмовою в обслуговуванні; вразливий хост або втрачає здатність спілкуватися в мережі, або в гіршому випадку зависає або виходить з ладу. Однак деякі (зокрема, спуфінг TCP) можуть мати серйозні реалізації зламу безпеки. Рівень забезпечення інформаційної безпеки є результатом ефективності методів управління трафіком мережі.

Вразливості протоколів мережевої взаємодії пов'язані з особливостями їхньої програмної реалізації та обумовлені обмеженнями на розміри застосовуваного буфера, недоліками процедури аутентифікації, відсутністю перевірок правильності службової інформації та ін..

Вразливості деяких протоколів TCP/IP (табл. 3).

У цих протоколів є й інші вразливості, які обумовлені помилками та недоробками у їх реалізації, потенційної наявності в них «закладок», шкідливих програм тощо. Є протоколи з підтримкою шифрування, процедури авторизації (наприклад, SFTP, SSH), але і їх не можна назвати повністю безпечними. Для систематизації опису множини вразливостей програмних мережевих протоколів та ПЗ використовується єдина база даних (БД) уразливостей **CVE** (Common Vulnerabilities and Exposures), у розробці якої брали участь фахівці багатьох відомих компаній та організацій, таких як MITRE, ISS, Cisco, BindView, Axent, NFR, L-3, CyberSafe, CERT, Carnegie Mellon University, Інститут SANS тощо. Ця БД постійно поповнюється і використовується при розробці численних програмних засобів аналізу захищеності та, насамперед, засобів моніторингу мереж.

Таблиця 3. Вразливості окремих протоколів стеку протоколів TCP/IP

Назва протоколу	Характеристика вразливості	Зміст порушення безпеки
IP (Internet Protocol) – протокол міжмережевої взаємодії	Криптографічно захищене з'єднання. Але вибір криптографічного набору сервером не зашифрований.	Атака man-in-the-middle (людина посередині) може фактично замінити хороший криптографічний вибір поганим (зламним) вибором, а потім розшифрувати та видавати себе за сервер.
TCP – протокол управління передачею	Відсутність механізму перевірки коректності заповнення службових заголовків пакета	Істотне зниження швидкості обміну та навіть повний розрив довільних з'єднань за протоколом TCP
FTP (File Transfer Protocol) – протокол передачі файлів через мережу	1. Аутентифікація на базі відкритого тексту (паролі пересилаються в незашифрованому вигляді). 2. Доступ за замовчуванням. 3. Наявність двох відкритих портів	Можливість перехоплення даних облікового запису (імен зареєстрованих користувачів, паролів). Отримання віддаленого доступу до хостів
Telnet – протокол керування віддаленим терміналом	Аутентифікація на базі відкритого тексту (паролі пересилаються в незашифрованому вигляді).	Можливість перехоплення даних облікового запису користувача. Отримання віддаленого доступу до хостів
UDP – протокол передачі даних без встановлення з'єднання	Відсутність механізму запобігання перевантаженню буфера. Відсутність перевірки доставки пакетів адресату	Можливість реалізації UDP-шторму. В результаті обміну пакетами відбувається суттєве зниження продуктивності сервера. Імовірність втрати інформації у процесі передачі.
ARP (Address Resolution Protocol) та (зворотний **RARP**) – протокол перетворення IP-адреси на фізичну адресу	Аутентифікація на базі відкритого тексту (інформація пересилається в незашифрованому вигляді)	Можливість перехоплення трафіку зловмисником
RIP2 (Routing Information Protocol v2) – протокол маршрутної інформації другої версії	Відсутність автентифікації керуючих повідомлень про зміну маршруту	Можливість перенаправлення трафіку через хост зловмисника
DNS – протокол встановлення відповідності менімонічних імен та мережевих адрес	Відсутність засобів перевірки аутентифікації отриманих даних від джерела	Фальсифікація відповіді DNS-сервера
ICMP (Internet Control Message Protocol) - протокол міжмережевих керуючих повідомлень, що використовується для повідомлення про помилки	Відсутність автентифікації повідомлень про зміну параметрів маршруту	Можливість підробки маршруту. Приводить до зупинки операційних систем Windows

SMTP – протокол забезпечення сервісу доставки повідомлень електронною поштою	Відсутність підтримки автентифікації заголовків повідомлень	Можливість підроблення повідомлень електронної пошти, а також адреси відправника повідомлення
SNMP – протокол управління маршрутизаторами в мережах	Відсутність підтримки аутентифікації заголовків повідомлень	Можливість досягнення максимальної пропускної спроможності мережі
OSPF (Open Shortest Path First) - протокол динамічної маршрутизації, заснований на алгоритмах відстеження стану каналу, задіяно алгоритм Дейкстри.	Відсутність автентифікації керуючих повідомлень про зміну маршруту. (В мережу з динамічною маршрутизацією вводяться шахрайські маршрути, що може призвести до того, що комунікації можуть бути захоплені, що вплине на потік трафіку в мережі).	Можливість припинення роботи основних частин програми. У гіршому випадку маршрутизатор вводиться в мережу зі зловмисних причин, щоб вкрасти дані програми або ввести шахрайські команди в систему управління.

Висновки

Модель TCP/IP стала стандартом для інтернет-зв'язку і використовується майже в кожному пристрої, підключеному до Інтернету. Цей протокол вибирають для всього, від невеликих локальних мереж до величезної, розгалуженої архітектури www.
Стек TCP/IP протоколів регулює взаємодію різних рівнів. Ключовим поняттям тут є протоколи, які утворюють стек, спираючись один на одного для передачі даних. TCP/IP модель має спрощену архітектуру в порівнянні з OSI. Сама TCP/IP модель залишається незмінною, а стандарти протоколів можуть бути оновлені, що робить роботу з TCP/IP ще простішою. Стек TCP/IP отримав широке поширення і використовувався спочатку як основа для створення глобальної мережі, а пізніше і для створення стабільного інтернету.
TCP – це основний протокол, наріжний камінь цифрової комунікації більшості сучасних інтернет-комунікацій, адаптивний і стійкий. Розуміння його основних функцій, структури сегментів, міркувань безпеки та додаткових функцій дає змогу зазирнути в складну інженерію, яка гарантує безперебійну та надійну цифрову взаємодію.

Контрольні питання

1. Які рівні має модель TCP/IP?
2. Чи безпечний FTP для передачі файлів?
3. Для чого використовуються мережеві протоколи?
4. Чим заголовок UDP відрізняється від заголовка TCP?
5. Які поля містить заголовок UDP?
6. Яке призначення протоколу DNS?
7. Яке призначення протоколів HTTP та HTTPS?
8. Яке призначення протоколу FTP?
9. Яке призначення протоколу SMTP?
10. Яке призначення протоколу POP3?
11. Яке призначення протоколу IMAP4?
12. Яке призначення протоколів Telnet та SSH?
13. Яке призначення протоколу SNMP?
14. Яке призначення протоколу TCP?
15. Яке призначення протоколу UDP?
16. Яке призначення протоколу IP?
17. Яке призначення протоколу DHCP?
18. Яке призначення протоколу ICMP?
19. Яке призначення протоколів ARP та RARP?
20. Яке призначення протоколу UDP?
21. Які вразливості найпоширеніших протоколів стеку TCP/IP мереж?

Додатково (необов'язково)

Список команд FTP (командний рядок Windows)

Команда FTP	**Опис команди**
!	Ця команда перемикається між операційною системою та ftp. Після повернення в операційну систему введення ex поверне вас назад до командного рядка FTP.
?	Відкриває екран довідки.

Команда FTP	Опис команди
append	Додайте текст до локального файлу.
ascii	Перейдіть у режим передачі ASCII.
bell	Вмикає або вимикає режим дзвінка.
binary	Переходить у режим двійкового переказу.
bye	Виходи з FTP.
cd	Каталог змін.
close	Виходи з FTP.
delete	Видаляє файл.
debug	Вмикає або вимикає налагодження.
dir	Відображає список файлів, якщо вони підключені. dir -C = Виводить файли у широкому форматі. dir -1 = Перелічує файли в звичайному форматі в алфавітному порядку. dir -r = Перелічує каталог у зворотному алфавітному порядку. dir -R = Виводить список усіх файлів у поточному каталозі та підкаталогах. dir -S = Перелічує файли в звичайному форматі в алфавітному порядку.
disconnect	Виходи з FTP.
get	Отримайте файл з віддаленого комп'ютера.
glob	Вмикає або вимикає поглинання. При вимкненні ім'я файлу в командах **put** і **get** сприймається буквально, і символи підстановки не враховуються.
hash	Вмикає або вимикає друк хеш-міток. При включенні на кожні 1024 байти отриманих даних виводиться хеш-марка (#).
help	Відкриває екран довідки та відображає інформацію про команду, якщо команду введено після довідки.
lcd	Відображає локальний каталог, якщо введено окремо, або якщо шлях, введений після РК-дисплея, змінить локальний каталог.
literal	Надсилає буквальну команду на підключений комп'ютер з очікуваною однорядковою відповіддю.
ls	Відображає файли віддалено підключеного комп'ютера.
mdelete	Багаторазове видалення.
mdir	Відображає вміст кількох віддалених каталогів.
mget	Отримайте кілька файлів.
mkdir	Створити довідник.
mls	Відображає вміст кількох віддалених каталогів.
mput	Надішліть кілька файлів.

Команда FTP	Опис команди
open	Відкриває адресу.
prompt	Вмикає або вимикає запит.
put	Надішліть один файл.
pwd	Друк робочої директорії.
quit	Виходи з FTP.
quote	Те саме, що і буквальна команда.
recv	Отримати файл.
remotehelp	Отримайте допомогу з віддаленого сервера.
rename	Перейменовує файл.
rmdir	Видаляє каталог на віддаленому комп'ютері.
send	Надішліть один файл.
status	Показує стан поточних увімкнених і вимкнених опцій.
trace	Перемикає відстеження пакетів.
type	Встановіть тип передачі файлів.
user	Надішліть нову інформацію про користувача.
verbose	Вмикає або вимикає докладність.