**КОМП’ЮТЕРНІ МЕРЕЖІ. ВСТУП**

**Вступ до курсу**

Ласкаво просимо до нашого другого курсу  **"Біти й байти комп’ютерних мереж".** Можливо, ви пам’ятаєте мої відео з першого курсу  або вирішили його пропустити – в такому випадку ми зустрічаємося вперше.

Мене звати Віктор Ескобедо, я інженер з корпоративних операцій Google.  Я захопився сферою ІТ ще у віці 9 років,  коли тато приніс додому наш перший комп’ютер.  Він був інженером-механіком і почав  використовувати його для CAD-проєктів.  Це був мій перший досвід з ПК.  Пізніше я зрозумів, що можу додавати інше ПЗ, зокрема ігри!

Невдовзі я та комп’ютер  почали його нервувати. Я все більше й більше цікавився тим, як він працює, врешті-решт почав заглядати всередину.  Знайшов елементи, які можна видалити, навіть ті, які не слід було видаляти.  Я рухався шляхом спроб і помилок.  Я не міг пояснити, що саме це було,  я просто знаходив механіку цього всього неймовірно захоплюючою.

Сьогодні я розумію – саме цей досвід обумовив мій вибір професії.  Але знаєте, там де я виріс,  про навчання в коледжі й продовження кар’єри не йшлося.  Я є мексикано-американцем першого покоління,  і я знав обмаль людей з технологічною кар’єрою.  Друзі та родина переважно турбувалися  про закінчення середньої школи та стабільну роботу,  не думаючи про довгострокову кар’єру.

У моїй школі кількість технічних класів була дуже обмеженою, і хоча мій батько й працював у машинобудуванні,  комп’ютер був для нього інструментом, як фреза, лінійка або молоток.  Батьки заохочували мене працювати й далі вивчати комп’ютери, але вони  не могли порадити коледж або щось інше для кар’єри в техніці.  Не маючи в тому своєї провини,  вони просто не мали необхідного досвіду.  Коли я вирішив піти до коледжу,  то захотів спробувати свої сили в інформатиці, аби задовільнити  своє бажання розуміти роботу ПК на більш фундаментальному рівні.  Я розумів, що наявність цих знань дозволить мені  розуміти концепції вищого рівня, що є критичним для кар’єри в ІТ.  Отже, ще перебуваючи в школі, я влаштувався на першу роботу в ІТ в невеличкій місцевій компанії. На сьогодні я працюю в ІТ вже 12 років, з них останні сім років – у Google. Зараз я працюю над реалізацією внутрішніх ІТ-проєктів для компанії, застосовуючи знання, отримані за роки праці у відділі підтримки, аби зрозуміти мій вплив на користувачів та різні команди підтримки. У ролі інженера з корпоративних операцій я відповідаю за розуміння впливу змін на корпоративну інфраструктуру. Через це навички роботи в мережі є критично важливими. Мені потрібно розуміти не тільки як працюють програми в одній системі, але й як вони взаємодіють з іншими системами компанії та системами зовні.

Тепер, коли ви трохи дізналися про мене, розглянемо наші "біти та байти". Комп’ютери спілкуються між собою подібно до людей. Гарний приклад – вербальне спілкування. Двоє людей мусять використовувати  ту саму мову та чути один одного, аби спілкуватися ефективно.  У шумному середовищі  одному з них, можливо, доведеться просити іншого повторити. Якщо одна людина тільки частково розуміє про що йдеться, то вона може попросити роз’яснень. Одна особа може звертатися лише до однієї особи чи розмовляти з групою.  І зазвичай існують як привітання, так і спосіб завершення розмови.  При спілкуванні люди дотримуються низки правил.  Комп’ютери поводяться аналогічно. Визначений набір стандартів, яких комп’ютери повинні  дотримуватися при спілкуванні, називається протоколом.  Комп’ютерні мережі – так ми називаємо  сукупність взаємодій, коли комп’ютери спілкуються між собою.  Комп’ютери мусять чути один одного, аби взаємодіяти в мережі,  використовувати протоколи, зрозумілі іншім, повторно надсилати повідомлення, які не повністю надійшли, зважати на інші речі,  як і в людському спілкуванні. Існує низка моделей, що описують різні рівні взаємодій у комп’ютерних мережах. І для цього курсу ми вибрали п’ятирівневу модель TCPIP. Ми також торкнемося іншої моделі первинної мережі,  моделі OSI, що має сім рівнів.  Якщо ви не знаєте, що це за моделі або як вони працюють, не хвилюйтеся.  Ми детально вивчатимемо ці теми протягом курсу. Знати ці багаторівневі моделі надважливо,  аби розуміти комп’ютерні мережі – ці багаторівневі системи. Протоколи кожного рівня приводять у дію протоколи над ними, переміщуючи дані з одного місця в інше. Протокол для переміщення даних між кінцями мережевого кабелю повністю відрізняється від протоколу для передачі даних з одного боку планети на інший. Але обидва ці протоколи повинні працювати одночасно, щоб такі речі, як Інтернет та бізнес-мережі працювали так, як вони працюють. Іноді виникають проблеми, коли комп’ютери в Інтернеті чи бізнес-мережах спілкуються між собою. Вирішують ці проблеми фахівці з ІТ-підтримки. Ось чому розуміння мереж є настільки важливим. До кінця цього курсу  ви розумітимете п’ять рівнів нашої моделі й не тільки це. Ви зможете описати, як комп’ютери визначають, куди надсилати повідомлення, як працюють мережеві служби DNS і DHCP і як використовувати потужні інструменти для вирішення мережевих проблем. Готові? Тож поїхали.

# Модель п’ятирівневої мережі TCP/IP

Щоб по-справжньому зрозуміти мережі, нам потрібно розуміти функції всіх компонентів. Починаючи від кабелів, що з’єднують пристрої між собою, і закінчуючи протоколами, які використовуються для зв’язку. Існує безліч моделей, що пояснюють, як мережеві пристрої спілкуються, але в цьому курсі ми зосередимось на п’ятирівневій моделі. До кінця цього уроку ви зможете ідентифікувати та описати кожен рівень і мету, якій він служить. **Почнемо з нижньої частини нашого стеку, де маємо наш фізичний рівень**. Фізичний рівень відповідає своїй назві. **Це фізичні пристрої, які з’єднують комп’ютери.**Це включає специфікації мережевих кабелів та роз’єми, які з’єднують пристрої між собою разом зі специфікаціями, що описують, як саме через ці з’єднання надсилаються сигнали. **Другий рівень нашої моделі відомий як канальний рівень. Його ще називають рівнем мережевого інтерфейсу чи рівнем доступу до мережі.**На цьому рівні ми представляємо наші перші протоколи. У той час як фізичний рівень – це кабелі, роз’єми та надсилання сигналів, канальний рівень відповідає за визначення загального способу інтерпретації цих сигналів, аби мережеві пристрої могли комунікувати. Існує низка протоколів канального рівня найбільш поширеним з яких є Ethernet, хоча бездротові технології й далі набирають обертів. Окрім визначення атрибутів фізичного рівня, стандарти Ethernet також визначають протокол, що відповідає за передачу даних між вузлами однієї мережі або каналу. **Третій рівень – рівень мережі, також називають інтернет-рівнем**. Саме цей рівень дозволяє різним мережам комунікувати через пристрої, відомі як маршрутизатори. Сукупність мереж, з’єднаних маршрутизаторами, формують інтермережі, найвідомішою з яких є Інтернет. Гадаю, ви чули про це. У той час як канальний рівень відповідає за передачу даних по каналу, мережевий рівень відповідає за передачу даних по сукупності мереж. Згадайте, як пристрій домашньої мережі з’єднується із сервером в Інтернеті. Саме мережевий рівень допомагає передавати дані між цими двома місцями. Найпоширеніший протокол цього рівня відомий як IP або інтернет-протокол. IP є серцем Інтернету та більшості невеликих мереж по всьому світу. Мережеве ПЗ поділяється на клієнтську та серверну категорії. Клієнтська програма ініціює запит даних, а серверне програмне забезпечення відповідає на мережевий запит. На одному вузлі може працювати низка клієнтських або серверних програм. Ви можете запустити програму електронної пошти та браузер (обидві клієнтські програми) на ПК одночасно, – і пошта, і браузер можуть працювати на одному сервері. Незважаючи на це, електронні листи потраплятимуть до поштового агента, а вебсторінки відкриватимуться в браузері. **Це відбувається завдяки наступному, транспортному рівню.**У той час як мережевий рівень передає дані між двома окремими вузлами, транспортний рівень з’ясовує, які саме клієнтські та серверні програми повинні отримати ці дані. Коли ви почули про IP-протокол мережевого рівня, можливо, ви згадали TCP IP, що є досить поширеною абревіатурою. Протокол, який найчастіше використовується на четвертому, транспортному рівні, відомий як TCP (протокол керування передаванням). Незважаючи на те, що TCP IP часто використовується разом, аби розуміти та ефективно вирішувати мережеві проблеми, важливо усвідомлювати, що це абсолютно різні протоколи, що виконують різні задачі. Інші протоколи передачі також використовують IP, зокрема протокол, відомий як UDP (протокол користувацьких дейтаграм). **Важлива різниця між ними в тому, що TCP забезпечує механізми надійної доставки даних, а UDP цього не робить.**Увага, спойлер! Ми детальніше розглянемо відмінності між протоколами TCP і UDP пізніше. Наразі важливо зрозуміти, що мережевий рівень, у нашому випадку – IP, відповідає за передачу даних між вузлами. Також пам’ятайте, що **транспортний рівень, переважно протоколи TCP та UDP, несе відповідальність за те, щоб дані потрапляли до потрібних додатків, що працюють на відповідних вузлах**. І останнє, але не менш важливе. **П’ятий рівень, або прикладний рівень. На цьому рівні працює безліч різних протоколів і, як ви могли здогадатися з назви, вони відносяться до програм.**Протоколи перегляду вебсторінок або роботи з поштою є найпоширенішими. Протоколи прикладного рівня вам найбільш знайомі, оскільки саме з ними ви безпосередньо взаємодіяли раніше, навіть того не усвідомлюючи. Ви краще зрозумієте рівні за аналогією посилки, що доставляється. Фізичний рівень – це вантажівка доставки та дороги. Канальний рівень – рух вантажівок від перехрестя до перехрестя. Мережевий рівень визначає, якими саме дорогами треба рухатись, щоб дістатися з адреси А до адреси Б. Транспортний рівень гарантує, що водій вантажівки знає, як саме треба постукати в двері й повідомити отримувача, що пакунок прибув. А прикладний рівень – це вміст самого пакунка.

# Алекс: чому саме мережі

Кожен з обчислювальних пристроїв, якими ми користуємося щодня, уже є мережевим пристроєм, чи не так? Комп’ютери втратили автономність. Чи то наш телефон, чи то планшет, ноутбук чи ПК. Усі вони, так чи інакше, є в мережі, у взаємодії з іншими пристроями. Для багатьох мережі – це чорна магія, і тільки вузьке коло спеціалістів розуміє, що насправді відбувається. Але з мого досвіду, IТ-спеціаліст із розумінням мереж на фундаментальному рівні здатен виконувати кожну зі своїх функцій набагато ефективніше. Є безліч курсів з мережевих технологій. Власне, цей предмет розпочали викладати в такий спосіб ще в 90-х роках. Цей курс є особливим, він зосереджений на низці практичних кейсів, фокусується на речах, важливих для персоналу ІТ-підтримки, а не тільки для мережевих інженерів. Ми приділимо достатньо часу DNS, розглянемо низку технік з усунення несправностей і поговоримо про базові речі, котрі мусить знати кожен ІТ-спеціаліст.

# Основні відомості про мережеві пристрої

# Кабелі

Різноманітні кабелі та мережеві пристрої використовуються для забезпечення належного зв’язку між комп’ютерами. Проглянувши це відео, ви будете в змозі самостійно визначити та описати їх різновиди. Комп’ютерні мережі – значна частка повсякденної роботи ІТ-спеціалістів, тому розуміння класифікації різних мережевих елементів є важливим.

Почнемо з найпростішої складової дротової мережі – кабелю. Кабель з’єднує пристрої між собою, дозволяючи їм отримувати або передавати дані. Сучасні мережеві кабелі найчастіше мідні або оптоволоконні. Мідний кабель є найбільш вживаним і складається з кількох пар мідних проводів у пластиковому ізоляторі. Відомо, що комп’ютери спілкуються у двійковий формі, представленій одиницями та нулями. Надсилаючий пристрій передає двійкові дані крізь мідні дроти, змінюючи напругу між двома діапазонами, а приймаючий пристрій інтерпретує зміни напруги як одиниці й нулі, та переводить в різноманітні формати даних. Найпоширенішими типами мідного кабелю, що використовується в мережах, є Cat5, Cat5e і Cat6 з мідної витої пари. Тобто кабелі п’ятої та шостої категорії. Відмінності полягають у характеристиках, як-от кількість скруток, різна робоча довжина та швидкість передачі даних. Категорія Cat5 є старішою, зараз більше використовуються Cat5e і Cat6. Зовні вони майже однакові та й зсередини дуже схожі. Відмінності полягають у тому, як виті пари розташовані всередині, що й є головним фактором, від якого залежить швидкість передачі даних та стійкість сигналу до зовнішніх перешкод. Кабелі Cat5e витіснили Cat5 завдяки внутрішнім компонентам, які зменшують взаємні перешкоди. Взаємна перешкода – це коли електричний імпульс одного з дротів помилково захоплюється іншим. Приймаючий вузол не розпізнає даних, що й викликає помилку мережі. Протоколи вищого рівня мають можливість виявити відсутні дані та виконати повторний запит. Це, звісно, потребує додаткового часу. Вища якість специфікацій кабелю Cat5e зменшує ймовірність потреби повторної передачі даних. У результаті за той самий проміжок часу буде оброблено більший обсяг даних. Cat6 має ще суворіші специфікації задля уникнення перехресних перешкод, що робить ці кабелі дорожчими. Кабель Cat6 передає дані швидше та є надійнішим, ніж кабелі Cat5e, але через його внутрішню структуру він має меншу робочу дистанцію на високих швидкостях. Другий різновид мережевого кабелю відомий як Fiber, що є скороченням для "волокняно-оптичний". Волокняно-оптичні кабелі містять окремі оптичні волокна, крихітні трубки зі скла завтовшки з людське волосся, що можуть транспортувати промені світла. На відміну від міді та електричної напруги, для передачі одиниць і нулів даних тут використовуються імпульси світла. Волокно використовується в середовищах з електромагнітними перешкодами від зовнішніх джерел, які можуть впливати на дані, що надсилаються по мідних дротах.

Волоконні кабелі передають дані значно швидше ніж мідні, але вони набагато дорожчі та крихкіші. Волокно може транспортувати дані на значно більші відстані, ніж мідь, та практично не зазнає втрат даних. Тепер ви знаєте більше про плюси й мінуси волоконних кабелів, але ви зустрічатимете їх частіше у центрах обробки даних, ніж в офісах або приватних будинках.

# Концентратори та комутатори

У цих двох відео ми пропонуємо вашій увазі огляд мережевих пристроїв.Оскільки кожному ІТ-спеціалісту доведеться взаємодіяти з ними на регулярній основі. Кабелі дозволяють утворювати мережеві з’єднання типу "точка–точка" – мережі, де на кожному кінці з’єднання лише один пристрій. Ми не будемо розглядати недоліки цих з’єднань, зауважимо лише, що вони не надто корисні у світі з мільярдами комп’ютерів. На щастя, існують мережеві пристрої, що дозволяють багатьом комп’ютерам спілкуватися між собою одночасно. Найпростіший із них – концентратор.

Концентратор – це пристрій фізичного рівня, що дозволяє з’єднувати декілька комп’ютерів одночасно. Усі пристрої, підключені до концентратора, можуть розмовляти з усіма іншими пристроями одночасно. Кожна підключена до концентратора система визначає, чи вхідні дані призначалися для неї, або ігнорує їх, якщо це не так. Це генерує багато шуму в мережі та створює домени колізій. Домен колізій – це сегмент мережі, де лише один пристрій може сповіщати одночасно. Коли декілька систем намагаються надсилати дані одночасно, електричні імпульси, що проходять через кабель, заважають один одному. Простої та багаторазові спроби систем надіслати дані знову, суттєво уповільнюють мережеві комунікації. Саме тому хаби зустрічаються досить рідко. Здебільшого вони вже стали історичними артефактами. Актуальніший спосіб з’єднання багатьох комп’ютерів потребує використання мережевого комутатора, відомого як концентратор-перемикач. Перемикач схожий на концентратор й дозволяє підключити декілька пристроїв, надаючи їм можливість спілкуватися. Різниця полягає в тому, що концентратор – це пристрій першого або фізичного рівня, а комутатор – другого рівня передачі даних. Це означає, що комутатор може перевірити контент кадрів Ethernet, що надсилаються по мережі, визначити, для якої із систем ці дані призначені й тільки тоді пересилати їх адресату. Це зменшує або навіть повністю усуває домени колізій у мережі. Завдяки цьому зменшується кількість повторних передач і підвищується пропускна здатність мережі.

# Маршрутизатори

Концентратори та перемикачі – головні пристрої для підключення комп’ютерів до мережі, яку називають локальною мережею (LAN). Але нам також треба надсилати та отримувати дані з інших мереж. Для цього нам знадобляться маршрутизатори. Маршрутизатор – це пристрій для пересилання даних між мережами. У той час як концентратор – пристрій першого рівня, а комутатор – другого, маршрутизатор – пристрій третього мережевого рівня. Як комутатор перевіряє дані Ethernet, визначаючи, куди надсилати дані, маршрутизатор перевіряє дані IP, щоб визначити, куди надіслати інформацію. Маршрутизатори зберігають таблиці з інформацією про маршрутизацію трафіку між безліччю різних мереж по всьому світу. Найпоширеніший тип маршрутизатора – це пристрої для домашньої мережі або невеличких офісів. Ці пристрої, як правило, не мають деталізованих таблиць маршрутизації. Призначення цих маршрутизаторів – приймати трафік, що надходить з будинку або локальної офісної мережі, і пересилати його інтернет-провайдеру. У провайдера трафік приймають значно складніші базові маршрутизатори. Це основні маршрутизатори, що складають інтернет-магістралі, і відповідають за надсилання та отримання даних з Інтернету щодня. Маршрутизатори провайдерів не лише обробляють набагато більше трафіку, ніж невеличкі офісні маршрутизатори. Вони мають справу зі складнішими завданнями, приймаючи рішення про маршрутизацію трафіку. Базовий маршрутизатор зазвичай має масу підключень до інших маршрутизаторів. Маршрутизатори діляться даними між собою через протокол, відомий як BGP, або "Протокол прикордонного шлюзу". Це дозволяє їм дізнатися про найбільш оптимальні шляхи пересилання трафіку. Коли ви завантажуєте вебсторінку, трафік між комп’ютерами та вебсерверами може подорожувати десятками різних маршрутизаторів. Інтернет є неймовірно великим і складним, а маршрутизатори є глобальними провідниками трафіку до потрібних нам місць.

# Сервери та клієнти

Усі мережеві пристрої, про які ви дізналися, існують, щоб комп’ютери могли спілкуватися один з одним, як знаходячись в одній кімнаті, так і в тисячі миль один від одного. Ми називаємо ці пристрої вузлами і вживатимемо цей термін надалі. Але також важливо розуміти терміни "сервер" і "клієнт". Щоб зрозуміти ці поняття, уявіть, що сервер — це пристрій, що надає дані іншим пристроям, які їх запитують. Пристрій, що отримує дані, – це клієнт. Хоча ми часто згадуємо вузли, які є серверами або клієнтами, причина, з якої ми вживаємо таке визначення, полягає в тому, що серверами або клієнтами можуть бути не тільки вузли. Окремі комп’ютерні програми, які працюють на одному вузлі, можуть бути серверами-клієнтами одна для іншої. Так само й більшість пристроїв не є лише сервером або клієнтом. Майже всі вузли бувають і тим, і іншим, оскільки ці пристрої є багатозадачними. У більшості мережевих топографій кожен вузол у першу чергу є або сервером, або клієнтом. Називаючи сервер електронної пошти сервером електронної пошти, ми не зважаємо на те, що сам він є клієнтом DNS-сервера. Чому? Тому що його головна функція – надавати дані клієнтам. Так само, коли настільний ПК діє як сервер, передаючи дані на інший ПК, його головна функція – це отримання даних із серверів, щоб користувач ПК міг виконувати свою роботу. Підсумовуючи, сервер – це все, що може надавати дані клієнтам, але ми також використовуємо цей термін, посилаючись на основну функцію різних вузлів у мережі. Усе зрозуміло? Супер! Тоді вже час для невеличкого тесту без оцінювання для перевірки ваших знань про мережі.

# Серджіо: робота мережевого інженера

Мене звати Серхіо Ла Торре, і я мережевий інженер Google. Формально, я працюю над продуктом YouTube TV та є частиною "Команди операторів лінійного ТБ". Ми забезпечуємо безперервну доступність Live TV для глядачів. Мережевий інженер – це той, хто "проєктує дороги" в Інтернеті. Мережевий інженер не має "звичайних днів". Це може бути що завгодно, від усунення несправностей до нарад, або колективна праця над складними проблемами чи проєктами. Стосовно ж мене та більшості мережевих інженерів, то це інцидент-менеджмент. Оперативна команда – це як пожежники для мережі. Коли мережа виходить з ладу, бізнес зупиняється. Так і в моєму випадку. Якщо наша мережа виходить з ладу, глядачі не можуть насолоджуватися ТБ. Нещодавнім інцидентом для нашої команди був ураган "Ірма". Ми мали багато телесервісів у Флориді, де люди намагалися використовувати телефони для перегляду новин та не мали такої можливості. Ми зреагували й розробили план, коли прийшов ураган, і ця послуга доступна й зараз. Я вважаю, що мережі дійсно важливі, вони відкривають шлях в Інтернет, який потрібен нашим пристроям щодня. Без мереж не було б можливості користуватись мобільними додатками, відвідувати сайти. Без мережевих інженерів, які створюють ці віртуальні шляхи, ви б не змогли насолоджуватися Snapchat або пошуком Google.

# **Фізичний р**івень

# Переміщення бітів за допомогою дротів

У певному сенсі фізичний рівень нашої мережі – стек-модель є найскладнішим з усіх. Його задача – переміщення одиниць і нулів з одного кінця каналу на інший. Дуже складна математика, фізика та електротехнічні рішення використовуються для передачі величезних обсягів даних по крихітних дротах на неймовірних швидкостях. На щастя для нас, більшість із цього відноситься до іншої сфери. Те що ви, як фахівець з ІТ-підтримки, мусите знати про фізичний рівень є набагато доступнішим. У кінці цього уроку ви повинні мати чітке розуміння різних аспектів фізичного рівня, що дозволить вам коректно усувати несправності в мережах і створювати нові мережі. Тож давайте розглянемо цю тему докладніше. Фізичний рівень складається з пристроїв та засобів передачі бітів у комп’ютерних мережах. Біт – це найменше представлення даних, які комп’ютер здатний зрозуміти. Це або одиниця, або нуль. Саме ці одиниці й нулі, що надсилаються мережею на найнижчому рівні, складають кадри й пакети даних, про які ми дізнаємося ще більше, коли говоритимемо про інші шари. Найголовніше полягає в тому, що незалежно від того, транслюєте ви свою улюблену пісню, надсилаєте електронний лист босу чи використовуєте банкомат, насправді ви лише надсилаєте одиниці та нулі на фізичному рівні низки різних мереж між вами та сервером, з яким ви взаємодієте. Стандартний мідний мережевий кабель, по підключенню пристроїв з обох сторін, нестиме постійний електричний заряд. Одиниці та нулі надсилаються таким мережевим кабелем з використанням процесу, що називається модуляцією. Модуляція – це спосіб зміни напруги заряду, що рухається кабелем. При використанні в комп’ютерних мережах цей вид модуляції більш відомий як лінійне кодування. Воно дозволяє обом пристроям на лінії зрозуміти, що електричний заряд в певному стані є нулем, а в іншому – одиницею. Завдяки цій відносно простій техніці сучасні мережі здатні пересилати 10 мільярдів одиниць і нулів через один мережевий кабель щосекунди.

# Кабелі з крученою парою і дуплексний зв’язок

Найпоширеніший тип кабелів, що використовуються для підключення обчислювальних пристроїв, відомий як вита пара. Він називається кабелем витої пари, оскільки має пари мідних проводів, скручених разом. Ці пари діють як єдиний провідник для інформації – вита конструкція сприяє захисту від електромагнітних та перехресних перешкод від сусідніх пар. Стандартний кабель Cat6 має вісім дротів, зібраних у чотири витих пари всередині однієї оболонки. Достеменна кількість пар залежить від технології передачі даних, яка буде використовуватись. Однак говорячи про сучасні форми мережевих взаємодій, важливо розуміти, що такий кабель забезпечує дуплексний зв’язок. Дуплексний зв’язок – це концепція, згідно з якою інформація може рухатися кабелем в обох напрямках. Також існує симплексний зв’язок, котрий є однонапрямним. Згадайте дитячий монітор, радіо-няню, де передача даних йде тільки в одному напрямку, що робить його симплексним зв’язком. Телефонний дзвінок є дуплексним – дві сторони можуть слухати та говорити. Спосіб, за допомогою якого мережеві кабелі забезпечують дуплекс, – це резервування однієї або двох пар для спілкування в одному напрямку й використання інших (однієї чи двох) пар для спілкування в іншому. Отже, пристрої з обох боків мережевого з’єднання можуть спілкуватися один з одним одночасно, що відомо як повний дуплекс. Якщо щось не так зі з’єднанням, зв’язок погіршується й можливе повідомлення про роботу "напівдуплекс". Напівдуплекс означає, що зв’язок, можливий в кожному напрямку, але тільки один пристрій може надавати інформацію одночасно.

# Мережеві порти і патч-панелі

Заключні етапи роботи фізичного рівня відбуваються в кінцевих точках мережевих з’єднань. На кінцях мережевих кабелів витої пари є роз’єм, що оголює окремі внутрішні дроти. Найпоширенішим є роз’єм RJ-45 або Registered Jack 45. Це одна з багатьох специфікацій кабельних роз’ємів і, безумовно, найбільш поширена в комп’ютерних мережах. Мережевий кабель з RJ-45 підключається до мережевого порту RJ-45. Як правило, мережеві порти безпосередньо приєднані до пристроїв, що складають комп’ютерну мережу. Комутатори мають чимало мережевих портів для підключення багатьох пристроїв, а сервери й настільні комп’ютери зазвичай лише один або два. На ноутбуці, планшеті чи телефоні, ймовірно, немає жодного. Але до бездротових мереж ми перейдемо пізніше. Більшість мережевих портів мають два невеличких світлодіоди. Один з них – індикатор з’єднання, інший – індикатор активності. Індикатор з’єднання світиться, коли кабель належним чином підключено до двох пристроїв, обидва з яких ввімкнені. Індикатор активності блиматиме, коли дані активно передаються по кабелю. Колись давно миготіння індикатора активності безпосередньо відповідало одиницям і нулям, що передавалися. Мережі сьогодні настільки швидкі, що індикатор активності повідомляє тільки про наявність трафіку загалом. На комутаторах може бути 1 світлодіод для стану посилання й активності. Він може вказувати й на інші параметри, наприклад швидкість передачі даних. Я рекомендую вивчати інструкції пристроїв, з якими працюєте, бо індикатори портів відображають інформацію про несправності. Іноді мережевий порт може підключатися до пристрою опосередковано. Натомість мережеві порти можуть бути вмонтовані в стінах або під столом. Ці порти підключаються до мережі за допомогою кабелів, що пролягають через стіни й закінчуються на комутаційній панелі. Комутаційна панель – пристрій з низкою портів без жодних інших функцій. Це просто контейнер для кінцевих точок купи кабелів. Додаткові кабелі йдуть від комутаційних панелей до комутаторів або маршрутизаторів, надаючи доступ до комп’ютерів на іншому кінці ліній.

# Канальний рівень

# Ethernet і MAC-адреси

Бездротовий та стільниковий доступ до Інтернету швидко стали поширеними способами підключення обчислювальних пристроїв до мереж. Гадаю, ви зараз теж піключились аналогічним чином. Ви можете здивуватися, почувши, що традиційні кабельні мережі все ще є найпоширенішим варіантом підключення на робочому місці. Найчастіше для передачі даних використовується протокол Ethernet. Ethernet канального рівня надає засоби для надсилання та отримання даних для ПЗ на вищих рівнях стека. Однією з основних цілей цього рівня є мінімізація потреби інших рівнів звертати увагу на фізичний рівень та апаратне забезпечення. З переміщенням такої відповідальності на канальний рівень Інтернет, транспортний та прикладний рівні можуть працювати без перешкод, незалежно від того, як саме підключено певний пристрій. Наприклад, вашому браузеру не потрібно знати, чи він працює на пристрої через виту пару чи через бездротове з’єднання. Аби тільки нижчі рівні надсилали та отримували для нього дані. Під кінець цього уроку ви зрозумієте, що таке MAC-адреси та як вони використовуються для ідентифікації комп’ютерів. Знатимете, як описати всі компоненти кадру Ethernet. Зможете розрізнити одноадресні, багатоадресні та трансляційні адреси. І, нарешті, зможете пояснити, як циклічні перевірки надмірності забезпечують цілісність даних, що надсилаються через Ethernet. Розуміння цих концептуальних речей допоможе вам усувати найрізноманітніші проблеми в якості фахівця з ІТ-підтримки. Попередження: попереду урок історії з традиційних технологій! Тож почнемо. Ethernet є досить старою технологією. Вона з’явилася в 1980 році й пройшла свою першу, повністю відшліфовану стандартизацію в 1983 році. З того часу було внесено декілька змін, насамперед для задоволення постійно зростаючих потреб пропускної здатності. Здебільшого, Ethernet, що використовується сьогодні, співставний зі стандартами, що були вперше опубліковані роки тому. У 1983 році комп’ютерні мережі були геть іншими, аніж сьогодні. Однією з помітних відмінностей топології була відсутність комутаторів та мережевих концентраторів. Це означало, що досить часто більшість або всі пристрої в мережі мали спільний домен колізій. Пам’ятаєте з нашої дискусії про концентратори й перемикачі, домен колізій – це сегмент мережі, де одночасно говорить лише один пристрій. Це тому, що всі дані в домені колізій надсилаються на всі підключені вузли. Коли два комп’ютери одночасно відправляють дані по каналу, це призводить до буквальних зіткнень електричного струму, що представляє наші одиниці та нулі, залишаючи кінцевий результат нерозбірливим. Ethernet, як протокол, вирішив цю проблему, використовуючи метод, відомий як множинний доступ з контролем несучої та виявленням колізій. Не те щоб легко злітало з вуст. Як правило, ми скорочуємо це до CSMA/CD. CSMA/CD використовується для визначення того, чи канали зв’язку є вільними, аби пристрій міг вільно передавати дані. Спосіб роботи CSMA/CD досить простий. Якщо наразі даних, що передаються в сегменті мережі, немає, то вузол буде вільно надсилати дані. Якщо ж два або більше комп’ютерів намагатимуться надіслати дані одночасно, вони виявляють зіткнення і припиняють надсилання даних. Після чого обидва пристрої очікують випадковий інтервал часу, перш ніж поновити спробу надіслати дані. Цей випадковий інтервал допомагає запобігти новим зіткненням комп’ютерів при наступній спробі надсилання, коли вони намагатимуться щось передати наступного разу. Якщо сегмент мережі є доменом колізії, це означає, що всі пристрої в сегменті будуть отримувати всі повідомлення, що транслюються по сегменту. Тобто потрібен спосіб визначити, якому саме вузлу адресована передача. Саме для цього й використовується адреса контролю доступу до носія або ж MAC-адреса. MAC-адреса – це унікальний глобальний ідентифікатор, приєднаний до мережевого інтерфейсу. Це 48-бітне число, представлене 6 групами з двох шістнадцяткових чисел. Як двійковий є способом представлення чисел лише двома цифрами, шістнадцяткове число є способом представлення чисел із 16 символів. Оскільки ми не маємо цифри більшої за дев’ять, шістнадцяткові числа використовують літери A, B, C, D, E та F для представлення чисел 10, 11, 12, 13, 14 та 15. Інший спосіб посилання на групу чисел у MAC-адресі – октет. Октет в комп’ютерних мережах є будь-яким числом, що може бути представлене 8 бітами. А дві шістнадцяткові цифри можуть представляти ті ж числа, що й 8 біт. Оскільки MAC-адреси мусять бути "глобально-унікальними", то, можливо, ви замислилися, як це взагалі можливо. Коротка відповідь полягає в тому, що 48-бітне число є неймовірно великим. Загальна кількість можливих MAC-адрес, становить 2 піднесену до 48 степеня, або 281 474 976 710 656 можливих унікальних адрес. А це досить багато. MAC-адреса складається з двох розділів. Перші три октети MAC-адреси відомі як організаційно унікальний ідентифікатор (OUI). Вони призначаються виробникам апаратного забезпечення, Інститутом інжинірингу електротехніки та електроніки (iEEE). Це корисна інформація, яку варто знати, оскільки завдяки їй можна ідентифікувати виробника інтерфейсу суто за його MAC-адресою. Останні три октети MAC-адреси можуть призначатися будь-яким способом, за бажанням виробника. Єдина умова – вони не можуть бути змінені й призначаються тільки раз, щоб зберегти всі MAC-адреси унікальними. Протокол Ethernet використовує MAC-адреси, щоб гарантувати, що дані містять як адресу відправника, так і адресу отримувача. Таким чином, навіть у сегменті мережі, що містить домен колізії, кожному вузлу відомо, коли трафік призначений для нього.

# Unicast (одностороннє передавання даних), Multicast (групове передавання даних) і трансляція

Досі ми обговорювали шляхи передачі даних з одного пристрою на інший, тобто одноадресну передачу. Одноадресна передача завжди призначена лише для однієї адреси отримувача. На рівні Ethernet це реалізується шляхом перегляду спеціального біта MAC-адреси отримувача. Якщо найменш значущий біт у першому октеті адреси призначення є нулем, це означає, що кадр Ethernet прямує тільки за адресою призначення. Тобто, що його буде надіслано всім пристроям в домені колізії, але буде отримано та оброблено тільки за адресою призначення. Якщо найменш значущий біт у першому октеті адреси є одиницею, це означає, що ви маєте справу з кадром багатоадресної розсилки. Багатоадресний кадр також надсилається на всі пристрої локальної мережі. Відмінність у тому, що його буде прийнято або відкинуто кожним пристроєм, залежно від критеріїв, крім власної MAC-адреси апаратного забезпечення. Мережеві інтерфейси може бути налаштовано приймати списки адрес багатоадресної розсилки для такого роду комунікацій. Третій тип Ethernet передачі відомий як широкомовна трансляція. Таке передавання Ethernet надсилається на кожний пристрій у локальній мережі. Це здійснюється за допомогою спеціальної широкомовної адреси. Широкомовне передавання Ethernet – це всі F. Таке передавання дозволяє пристроям дізнатися більше один про одного. Не хвилюйтеся, ви дізнаєтеся більше про широкомовні трансляції та технологію ARP, відому як протокол розв’язання адрес, пізніше. Наразі перейдімо до аналізу кадру Ethernet.

# Аналіз кадру Ethernet

Для завершення наших уроків про основи організації мереж, проаналізуймо кадр Ethernet. Розуміння основ мережі – це перший крок у побудові потужної основи мережевих знань, життєво необхідних в ІТ-підтримці. Пакет даних – це всеохоплюючий термін, представлений окремою множиною двійкових даних у мережевому каналі. Термін "пакет даних" не прив’язаний до певного рівня чи технології, він являє собою лише концепцію. А саме: один набір даних надсилається з точки А в точку Б. Пакети даних на рівні Ethernet відомі як кадри Ethernet. Кадр Ethernet є високо структурованою сукупністю даних в особливому порядку. Мережеві інтерфейси фізичного рівня можуть перетворити бітові комбінації в каналі на значущі дані або навпаки. Майже всі поля кадру Ethernet є обов’язковими, більшість з них мають фіксований розмір. Перша частина кадру Ethernet відома як преамбула. Преамбула має довжину 8 байт (64 біти) і може містити дві секції. Перші сім байт – серія з поперемінних одиниць і нулів. Ці частини діють як буфер між кадрами й використовуються мережевими інтерфейсами як синхроімпульси для регулювання швидкості передачі даних. Цей останній байт у преамбулі відомий як роздільник кадрів (SFD). Він сигналізує приймальному пристрою, що преамбула закінчена й далі йтиме безпосередній зміст кадру. Відразу після SFD йде MAC-адреса отримувача, апаратна адреса безпосереднього отримувача. Після якої – вихідна MAC-адреса, адреса надсилання кадру. Не забувайте, що кожна MAC-адреса має довжину 48 біт або 6 байт. Наступна частина кадру Ethernet називається полем EtherType. Воно має довжину 16 біт й описує протокол змісту кадру. Ми розглянемо ці протоколи детальніше трохи пізніше. Варто згадати, що замість поля EtherType, ви можете зустріти те, що відомо як заголовок VLAN. Це вказує на те, що сам кадр є VLAN кадром. Якщо присутній заголовок VLAN, поле EtherType слідує за ним. VLAN означає "віртуальна локальна мережа". Ця техніка дозволяє мати декілька логічних локальних мереж на одному й тому ж фізичному обладнанні. Будь-який кадр з тегом VLAN буде доставлено лише інтерфейсом комутатора, налаштованим транслювати цей тег. Таким чином, одна фізична мережа може працювати наче кілька локальних мереж. Віртуальні мережі використовуються для розділення різних форм трафіку. Наприклад, IP-телефонія компанії може працювати на одній VLAN, а стаціонарні комп’ютери на іншій. Далі ви знайдете корисне навантаження кадру Ethernet. Корисне навантаження в сенсі мережі – це фактичні дані, що транспортуються, тобто все те, що не є заголовком. Довжина корисного навантаження стандартного кадру Ethernet може бути від 46 до 1500 байт. Тут містяться всі дані з вищих рівнів, як-от IP, тип транспорту й рівень додатків, що фактично передаються. Слідом за цими даними йде послідовність перевірки кадрів. Це 4-байтове (32-бітове) число, що являє собою контрольну суму кадру. Значення контрольної суми розраховується шляхом виконання циклічної перевірки кадру надмірним кодом. Циклічна перевірка надмірності (CRC) – важливе поняття цілісності даних, що використовується при всіх обчисленнях, не тільки при передачі даних. CRC – це математичне перетворення, що використовує поліноміальний поділ для створення числа, яке являє собою більший набір даних. У результаті виконання CRC по відношенню до набору даних, результатом має бути ідентичне число контрольної суми. Причина включення CRC до кадру Ethernet – надати інтерфейсу приймаючої мережі можливість визначення пошкодженості даних. Коли пристрій готується надіслати кадр, він збирає всю інформацію, що ми розглянули, як-от MAC-адреси відправника й отримувача даних, корисне навантаження тощо. Він виконує CRC цих даних і приєднує отриманий номер контрольної суми як контрольну послідовність кадру в кінці кадру. Ці дані надсилаються мережевим каналом і отримуються на іншому кінці, де всі поля кадру Ethernet збираються і тепер уже приймаюча сторона виконує CRC цих даних. Якщо обчислена контрольна сума не збігається з контрольною сумою в полі послідовності перевірки кадрів, дані відкидаються, оскільки певну кількість інформації було втрачено або пошкоджено під час передачі. Після цього протокол вищого рівня вирішує, чи ці дані потрібно передати ще раз. Ethernet звітує виключно про цілісність даних, не виконуючи їх відновлення. Ми добре попрацювали над принципами функціонуванням мереж. Далі буде тест, з яким, я впевнений, ви впораєтесь. Якщо ж ні – перегляньте матеріал ще раз, аби почуватися впевненіше.

# Мережевий рівень. Вступ

Комп’ютери здатні взаємодіяти на величезних відстанях із майже миттєвою швидкістю. Це технічне досягнення вражає, адже мільярди людей користуються Інтернетом щодня. Раніше в цьому курсі ми дізналися, як комп’ютери взаємодіють між собою на коротких відстанях або в одному сегменті мережі чи локальній мережі. На наступних уроках ми зосередимося на технологіях, які дозволяють даним перетинати багато мереж, що полегшує зв’язок на великих відстанях.У кінці цього модуля ви зможете описати схему IP-адреси й те, як формуються підмережі. Ви навчитеся виконувати базові математичні дії у двійковій системі, щоб описувати підмережі. Ви також зможете продемонструвати, як працює інкапсуляція та як протоколи, як-от ARP, дозволяють взаємодіяти різним рівням мережі. І, нарешті, ви будете розуміти основи маршрутизації, протоколи маршрутизації та те, як працює Інтернет. А зараз "маршрутизуйтесь" до наступного відео, і розпочнімо.

# Мережевий рівень

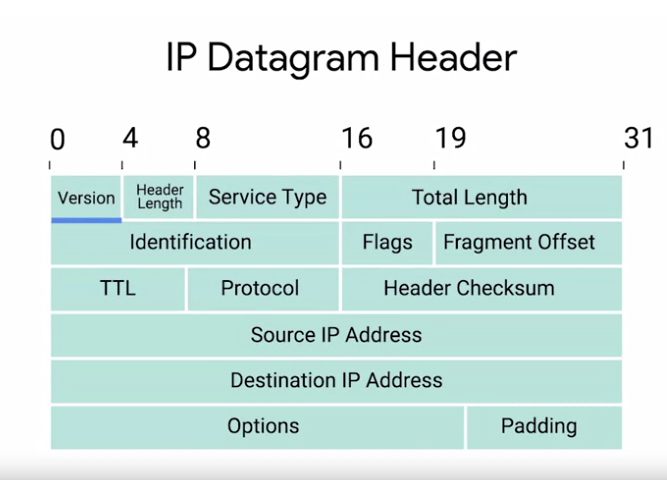
У локальній мережі (або LAN) вузли можуть обмінюватися даними через свої фізичні MAC-адреси. Це добре працює в невеликих масштабах, адже комутатори можуть швидко дізнаватися MAC-адреси, з’єднані між собою портами, щоб належно передавати дані. Але схема використання MAC-адрес погано масштабується. Кожен окремий мережевий інтерфейс на планеті має свою унікальну MAC-адресу, і вони не впорядковані систематично. Неможливо дізнатися, у якій саме точці планети знаходиться певна MAC-адреса в даний момент, тому це не ідеальний варіант для обміну даними на відстані. Пізніше в цьому уроці ми дізнаємося про ARP (протокол визначення адрес), і ви побачите, що спосіб, яким вузли дізнаються фізичний адрес один одного, не перетворюється ні на що інше, крім одного сегмента мережі. Очевидно, що нам потрібне інше рішення. І це – мережевий рівень та інтернет-протокол (або IP) в IP-адресах, що пов’язані з ним. Після цього уроку ви вмітимете ідентифікувати IP-адресу, описувати, як інкапсулюються IP-дейтаграми в корисних даних кадру Ethernet, і правильно визначати й опасувати поля заголовка IP-дейтаграми.

# IP-адреси

P-адреси – це 32-розрядні числа, що складаються з 4 октетів. Кожен октет зазвичай виражається десятковим числом. 8 бітів даних, або один октет, може представляти всі десяткові числа від 0 до 255. Наприклад, 12.34.56.78 – це дійсна IP-адреса. Але 123.456.789.100 було б недійсною адресою, тому що містить числа більші, ніж можна представити 8 бітами. Цей формат відомий як десяткове представлення з крапками. Ми глибше вивчимо, як це працює, на майбутньому уроці про підмережі. Наразі важливо знати, що IP-адреси великими групами розподіляються між різними організаціями й компаніями, а не визначаються постачальниками апаратного забезпечення. Це означає, що IP-адреси мають чіткішу ієрархію й зручніші для зберігання даних про них, ніж фізичні адреси. Подумайте про IBM, що володіє кожною IP-адресою з цифрою 9 у першому октеті. На дуже високому рівні це означає: якщо маршрутизатору потрібно визначити, куди надіслати пакет даних, призначений для IP-адреси 9.0.0.1, то все, що йому потрібно знати, – пакет потрібно доставити на один із маршрутизаторів IBM. А вже той маршрутизатор подбає про решту процесу доставлення. Важливо зазначити, що IP-адреси належать до мереж, а не до пристроїв, підключених до цих мереж. Тому ваш ноутбук завжди матиме однакову MAC-адресу, де б ви його не використовували. Але IP-адреса, призначена йому в інтернет-кафе, відрізнятиметься від тієї, яку він має, коли ви вдома. Локальні мережі в інтернет-кафе й у вас удома будуть кожна окремо відповідати за призначення IP-адреси ноутбуку, коли ви ввімкнете його там. Щоденний процес отримання IP-адреси зазвичай майже непомітний. На одному з наступних уроків ви більше дізнаєтеся про деякі з цих технологій. Поки що запам’ятайте, що до багатьох сучасних мереж можна підключити новий пристрій, і йому автоматично буде призначено IP-адресу за допомогою технології, відомої як протокол динамічного налаштування вузла. Призначену таким чином IP-адресу називають динамічною IP-адресою. Протилежністю цього поняття є статична IP-адреса, яку потрібно налаштувати на вузлі вручну. У більшості випадків статичні IP-адреси зарезервовані для серверів і мережевих пристроїв, а динамічні — для клієнтів. Але, звісно, є ситуації, коли це може бути не так.

# IP-дейтаграми та інкапсуляція

Так само як усі пакети даних на рівні Ethernet мають свою назву – кадри Ethernet, свою назву мають і пакети на рівні мережі. У протоколі IP такий пакет зазвичай називають IP-дейтаграмою. Як і будь-який кадр Ethernet, IP-дейтаграма – це чітко структурована серія строго визначених полів. Два основні розділи IP-дейтаграми – заголовок і корисна інформація. Як ви могли помітити, заголовок IP-дейтаграми містить набагато більше даних, ніж заголовок кадру Ethernet. Найперше поле складається з 4 бітів і вказує, яка версія інтернет-протоколу використовується. Найпоширенішою версією IP є версія 4, або IPv4. Версія 6, або IPv6, стрімко набирає широкої популярності, тому ми розглянемо її в одному з наступних модулів. Після поля версії йде поле довжини заголовка. Воно теж складається з 4 бітів і визначає довжину всього заголовка. У випадку IPv4 ця довжина майже завжди становить 20 байтів. Фактично, 20 байтів – це мінімальна довжина IP-заголовка. Ви б не змогли вмістити всі дані для IP-заголовка належного формату в меншому просторі. Далі йде поле типу сервісу. Ці 8 бітів можуть використовуватися для зазначення деталей про якість сервісу або технології QoS. Щодо QoS важливо знати, що існують сервіси, які дозволяють маршрутизаторам вирішувати, яка IP-дейтаграма важливіша за інші. Наступне поле складається з 16 бітів і називається "Загальна довжина". Його призначення відповідає назві: позначати загальну довжину відповідної IP-дейтаграми. Поле ідентифікації – це 16-бітне число, яке використовується для групування повідомлень. IP-дейтаграми мають максимальний розмір, і ви вже можете зрозуміти, який він. Оскільки поле "Загальна довжина" містить 16 бітів і воно вказує розмір окремої дейтаграми, максимальний розмір однієї дейтаграми – це найбільше число, яке може представляти у вигляді 16 бітів, а саме: 65 535. Якщо загальний обсяг даних, які потрібно надіслати, більший, ніж можна вмістити в одну дейтаграму, рівень IP має розділити ці дані на багато окремих пакетів. У такому разі поле ідентифікації використовується так, що одержувач розуміє, що кожен пакет з однаковим значенням у цьому полі є частиною однієї й тієї ж передачі. Далі йдуть два тісно пов’язані між собою поля: "Прапорці" і "Зміщення фрагментації". Поле "Прапорці" вказує на те, чи має дейтаграма дозвіл на фрагментацію або чи її вже фрагментовано. Фрагментація – це процес, коли ми беремо одну IP-дейтаграму й розбиваємо її на декілька менших дейтаграм. Хоча більшість мереж працюють зі схожими налаштуваннями щодо дозволеного розміру IP-дейтаграм, іноді цей параметр можна налаштувати інакше. Якщо дейтаграма має переходити з мережі, де дозволено більший розмір дейтаграм, у мережу з меншим розміром дейтаграм, така дейтаграма має бути фрагментована на дрібніші. Поле зміщення фрагментації містить значення, які одержувач використовує, щоб прийняти всі частини фрагментованого пакета й знову скласти їх у правильному порядку. Перейдімо до поля "Термін дії", або TTL. Це 8-бітове поле, що вказує, скільки маршрутизаторів може пройти дейтаграма, перш ніж її буде видалено. Щоразу, коли дейтаграма надходить на новий маршрутизатор, він зменшу значення в полі TTL на 1. Коли це значення досягає нуля, маршрутизатор знає, що він не має пересилати дейтаграму далі. Основне завдання цього поля – забезпечувати, щоб, коли маршрутизація налаштована неправильно й це спричиняє нескінченний цикл, дейтаграми не намагалися б без кінця досягти місця призначення. Нескінченний цикл може виникати, якщо маршрутизатор А вважає маршрутизатор В наступним пунктом призначення, а маршрутизатор B вважає, що ним є маршрутизатор A. Увага: спойлер! У наступному модулі ви дізнаєтеся, що поле TTL дуже корисне для усунення несправностей, але подібні секрети відкриються лише тим, хто продовжить навчання. Після поля TTL йде поле "Протокол". Це ще одне 8-бітне поле, що містить дані про те, який протокол транспортного рівня використовується. Найпоширенішими протоколами транспортного рівня є TCP і UDP. Ми докладно розглянемо їх на наступних уроках. Отже, наступне поле – "Контрольна сума заголовка". Це поле – контрольна сума вмісту всього заголовка IP-дейтаграми. Це працює дуже схоже на поле контрольної суми Ethernet, яке ми розглядали в попередньому модулі. Оскільки поле TTL має обчислюватися повторно на кожному маршрутизаторі, куди надходить дейтаграма, значення в полі контрольної суми теж неодмінно змінюється.  І ось нарешті ми дістаємося до двох дуже важливих полів: вихідної та цільової IP-адрес. Як ви пам’ятаєте, IP-адреса – це 32-бітне число, тому не дивно, що кожне з цих полів має довжину 32 біти. Далі йде поле параметрів IP. Це необов’язкове поле, яке встановлює особливі характеристики для дейтаграм, що використовуються переважно з метою тестування. За полем параметрів IP зазвичай слідує поле незначущої інформації. Оскільки поле параметрів IP необов’язкове й має змінну довжину, поле незначущої інформації – це просто серія нулів, що використовуються для забезпечення правильного розміру заголовка. Тепер, коли ви знаєте про всі частини IP-дейтаграми, ви можете запитати, як це пов’язано з тим, що ми вивчили досі. Ви, мабуть, пам’ятаєте: коли ми розбирали структуру фрейма Ethernet, ми згадали про розділ, який описали як розділ корисних даних. Ось саме він і є IP-дейтаграмою, а цей процес називається інкапсуляція. Весь вміст IP-дейтаграми інкапсулюється як корисна інформація кадру Ethernet. Можливо, ви помітили, що наша IP-дейтаграма також містить розділ корисної інформації. Вмістом цієї корисної інформації є вся сукупність пакета TCP або UDP, який ми розглянемо пізніше. Сподіваюся, це допоможе вам краще зрозуміти, чому ми говоримо про мережі з точки зору рівнів. Кожен рівень потрібен для рівня над ним.



# Класи IP-адрес

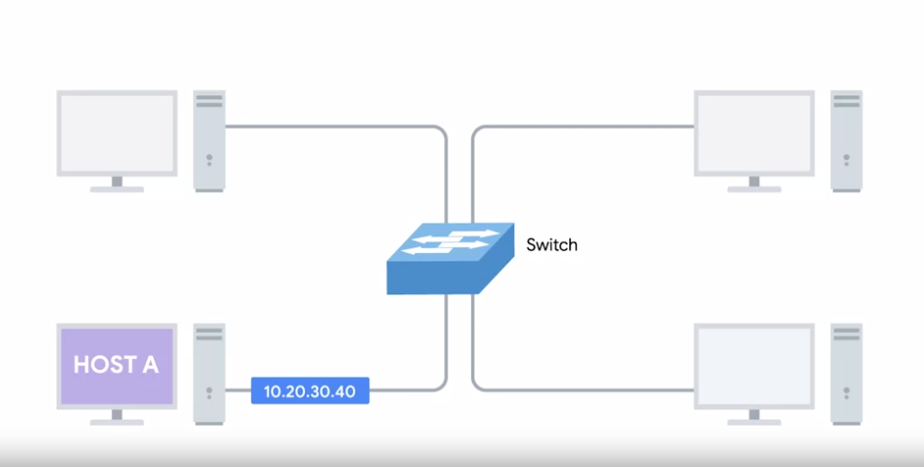
IP-адреси складаються з двох розділів: ідентифікатор мережі й ідентифікатор вузла. Раніше ми згадували, що IBM володіє всіма IP-адресами, тому перший октет в IP-адресі має значення 9. Якщо взяти як приклад IP-адресу 9.100.100.100, ідентифікатором мережі буде перший октет, а ідентифікатором вузла – другий, третій і четвертий октети. Система класів адрес – це спосіб визначення того, як поділено глобальний простір IP-адрес. Існує три основні типи класів адрес: клас А, клас В і клас С. Адреси класу А – це ті, де перший октет є ідентифікатором мережі, а останні три є ідентифікатором вузла. Адреси класу B – це ті, де перші два октети є ідентифікатором мережі, а два інші – ідентифікатором вузла. Як ви могли здогадатися, адреси класу С – це ті, де перші три октети є ідентифікатором мережі й лише кінцевий октет є ідентифікатором вузла. Усі класи адрес представляють мережі дуже різного розміру. Наприклад, оскільки мережа класу А має 24 біти для ідентифікатора вузла, це означає два у двадцять четвертій степені, або 16 777 216 індивідуальних адрес. Порівняйте це з мережею класу С, що має лише 8 бітів місця для ідентифікатора вузла. Для мережі класу С це дає два у восьмій степені, або 256 адрес. Ви також можете точно сказати, до якого класу належить IP-адреса, просто поглянувши на неї. Якщо найперший біт IP-адреси дорівнює 0, вона належить до мережі класу А. Якщо перші біти дорівнюють 1 і 0, вона належить до мережі класу В. Нарешті, якщо перші біти дорівнюють 110, вона належить до мережі класу С.Оскільки люди не дуже добре рахують у двійковій системі, корисно знати, що ці значення чудово перетворюються на представлення цих адрес у десяткові з крапками. Як ви пам’ятаєте, кожен октет в IP-адресі – це вісім бітів. Це означає, що кожен октет може приймати значення від 0 до 255. Якщо перший біт повинен бути нулем, тому що це, скажімо, адреса класу А, можливі значення для першого октету складають від 0 до 127 включно. Це означає, що будь-яка IP-адреса, де перший октет має одне з цих значень, є адресою класу А. Аналогічно, до адрес класу B належать лише ті, що починаються зі значення першого октета від 128 до 191 включно, а адреси класу С починаються зі значення першого октета від 192 до 223. Ви можете помітити, що ми не охоплюємо всі можливі IP-адреси. Все дуже просто: ще два класи IP-адрес, але вони не настільки важливі для розуміння. Адреси класу D завжди починаються з бітів 1110 і використовуються для багатоадресної розсилки пакетів. Саме так можна одну IP-дейтаграму можна надіслати всій мережі одночасно. Ці адреси починаються з десяткових значень від 224 до 239. Нарешті, до класу E належать усі решта IP-адрес, але вони не призначені й використовуються лише для тестування. З практичних міркувань цю систему класів здебільшого було замінено на систему, відому як CIDR (безкласова міждоменна маршрутизація). Але система класів адрес досі багато де застосовується, і її важливо розуміти будь-кому, хто хоче отримати універсальну освіту у сфері мереж. І на цьому поки все. Не хвилюйтеся, систему CIDR ми розглянемо на одному з наступних уроків.

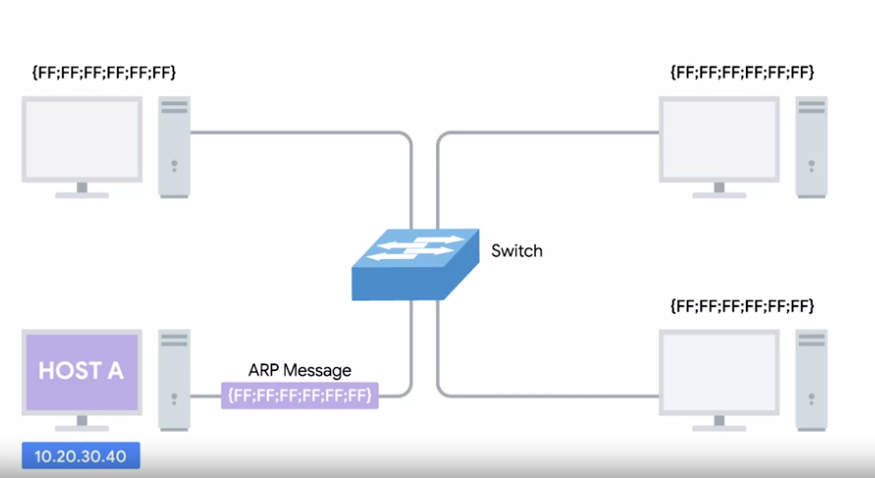


<https://www.geeksforgeeks.org/what-is-network-id-and-host-id-in-ip-addresses/>

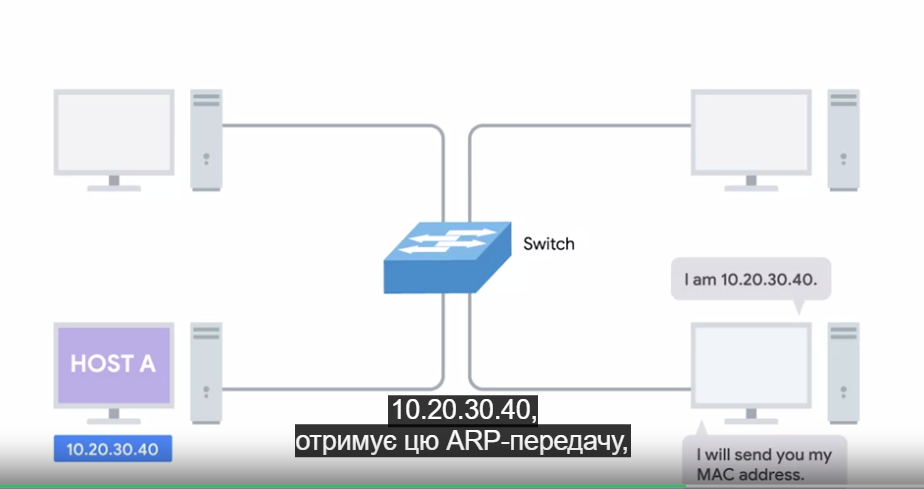
# Протокол перетворення адрес

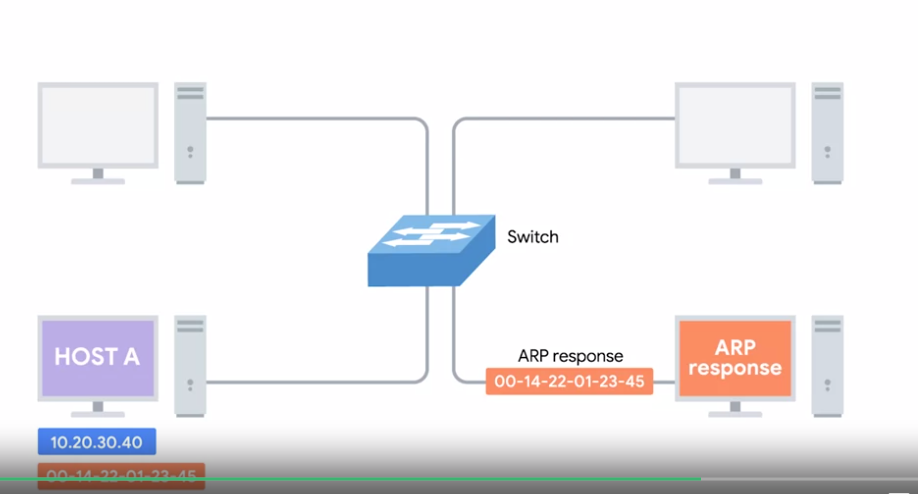
Вітаю! Тепер ви розумієте, як використовуються MAC-адреси на канальному рівні та як IP-адреси використовуються на мережевому. Тепер нам треба обговорити, як ці 2 типи адрес пов’язані між собою. Тут вступає в гру протокол визначення адрес або ARP. ARP − це протокол, який використовується для виявлення фізичної адреси вузла з певною IP-адресою. Після повного формування IP-дейтаграми вона має бути інкапсульована всередині кадру Ethernet. Це означає, що передавачу потрібна MAC-адреса призначення для завершення заголовка кадру Ethernet. Майже всі підключені до мережі пристрої зберігають локальну ARP-таблицю. В ARP-таблиці наведено лише список IP-адрес і пов’язаних з ними MAC-адрес. Скажімо, ми хочемо надіслати певні дані на IP-адресу 10.20.30.40. Можливо, ця адреса не записана в ARP-таблиці. Коли це трапляється, вузол, який хоче надіслати дані, надсилає широкомовне ARP-повідомлення на широкомовну MAC-адресу, що позначена літерами F. Ці види широкомовних ARP-повідомлень доставляються на всі комп’ютери локальної мережі. Коли мережевий інтерфейс, якому було призначено IP-адресу, 10.20.30.40, отримує цю ARP-передачу, він надсилає назад те, що називається ARP-відповіддю. Це повідомлення-відповідь міститиме MAC-адресу для вищезгаданого мережевого інтерфейсу. Тепер комп’ютер-передавач знає, яку MAC-адресу вставляти в поле фізичної адреси призначення, і кадр Ethernet готовий до доставлення. Він, імовірно, збереже цю IP-адресу у своїй локальній ARP-таблиці, щоб йому не довелося знову надсилати ARP-передавання, коли йому треба буде зʼєднатися з цією IP. Зручно! Як правило, записи з ARP-таблиці зникають через короткий проміжок часу для того, щоб відбувалося врахування змін у мережі. Тому не розраховуйте на їх присутність так, як я розраховую на вашу на наступному тесті без оцінювання.











# Пошук даних IP-адрес

**Ваша IP-адреса не є секретом.**

Багатьох користувачів турбує, що за допомогою IP-адреси можна дізнатися їхнє ім’я, домашню адресу, вік, переглянуті вебсторінки тощо. Це не зовсім так. Звісно, певну інформацію можна з’ясувати, але нічого важливого.

Дізнаймося, які відомості доступні, вказавши реальну IP-адресу на сайті для пошуку даних IP-адрес, як-от

<http://whatismyipaddress.com/>

Існує безліч причин, чому люди використовують IP Lookup, навіть з усіма його обмеженнями:

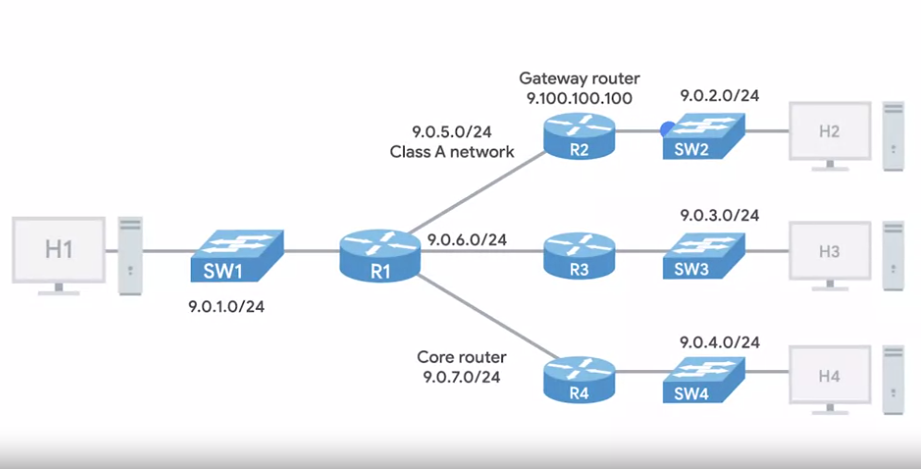
* Правоохоронці та слідчі відділу шахрайств використовують онлайн-інструменти, щоб визначити, у якого інтернет-провайдера розміщений спамер.
* Бази даних "чорних списків" використовують їх для пошуку спамерів або інших порушників і для блокування їхнього доступу до серверів електронної пошти.
* Магазини часто використовують IP Lookup, щоб переконатися, що покупець, який витрачає тисячі доларів, перебуває за поштовою адресою, пов’язаною з банківською карткою, а не користується викраденою кредиткою за кордоном.
* Ці сайти також можна використовувати для того, щоб перевірити, чи відправник електронного листа, який зазначає своє місцезнаходження поблизу, насправді не перебуває на закинутому складі в іншій країні.

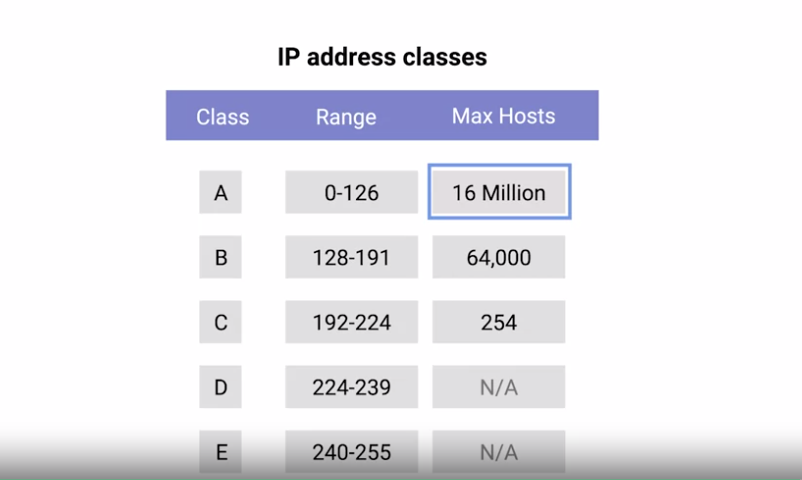
# Кар’єрний шлях Серджіо

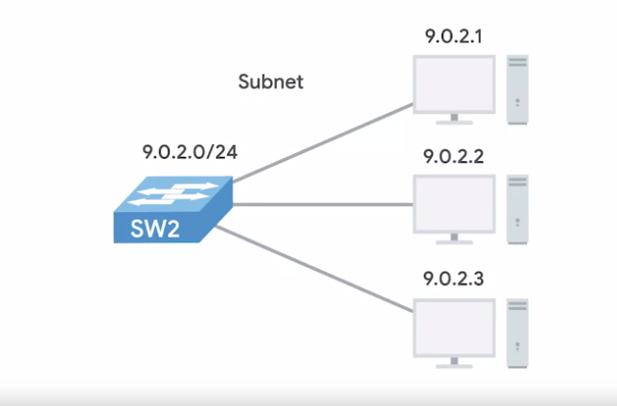
Мене звати Серджіо Ла Торре, і я інженер мереж у Google. Ще в дитинстві мені подобались аналогові інструменти, як-от викрутки та свердла, і я любив розбирати речі на частини. В підлітковому віці я зрозумів, що можу розбирати комп’ютери, і це мене дуже зацікавило, тому що, коли відкриваєш його, там є різні вентилятори, лампочки, схеми тощо. Я їх не розумів, це було як китайська грамота, але я знав, що, зібравши усе правильно, я зможу розважатися й грати у відеоігри. Так усе й почалося. Коли я підріс і закінчив школу, то зрозумів, що це може стати моїм хобі і я можу дізнатися, як працюють ці технології. Думаю, моя улюблена частина – усунення несправностей. Мені подобається знати, як дані передаються з пристрою на пристрій, спілкуватися з клієнтами та вирішувати їх проблеми. Я опанував стільки різних інструментів і навичок, що тепер у роботі почуваюсь як швейцарський армійський ніж. У мене так багато різного досвіду, який я можу використати для вирішення актуальних проблем. На моїй першій посаді спеціаліста з внутрішніх мереж був проєкт модернізації брандмауера для департаменту поліції. Брандмауер зазвичай використовується як засіб безпеки на вході в мережу. Ми думали, що все пройшло успішно, поки мені не зателефонували дуже пізно вночі й не сказали, що все погано. І ми зрозуміли, що електронні листи й нові виклики поліції не проходили й що модернізація не вдалася. Тоді я усвідомив важливість речей, над якими я працюю, і що це може впливати на життя людей. Ми можемо приносити користь або завдавати шкоди. Робота над тим проєктом справді стала для мене гарним уроком.

# Підмережі

У базовому розумінні створення підмереж – це процес, коли велика мережа розділяється на багато окремих підмереж меншого розміру. Прослухавши цей урок, ви зможете пояснити, навіщо потрібні підмережі, описати, як маски підмережі розширюють можливості порівняно лише з ідентифікаторами мережі та вузла. Ви також зможете пояснити, як технологія під назвою CIDR забезпечує ще більшу гнучкість, ніж звичайна підмережа. Нарешті, ви зможете застосувати базові методи двійкової математики, щоб краще зрозуміти, як усе це працює. Неправильне налаштування підмереж – поширена проблема, з якою ви можете зіткнутися як фахівець з ІТ-підтримки, тому важливо добре розумітися на підмережах. Це чимало матеріалу, тож давайте розбиратися. Як ви, мабуть, пам’ятаєте з останнього уроку, класи адрес дають нам спосіб розбити весь глобальний IP-простір на дискретні мережі. Якщо ви хочете зв’язатися з IP-адресою 9.100.100.100, магістральні маршрутизатори в Інтернеті знають, що ця IP-адреса належить до мережі 9.0.0.0 класу А. Вони направляють повідомлення на маршрутизатор шлюзу, який відповідає за мережу, перевіряючи ідентифікатор мережі. Зокрема, маршрутизатор шлюзу є точкою входу та виходу певної мережі. Цим він відрізняється від магістральних маршрутизаторів, які спілкуються лише з іншими магістральними маршрутизаторами. Щойно ваш пакет потрапляє на маршрутизатор шлюзу для мережі 9.0.0.0 класу A, цей маршрутизатор тепер відповідає за доставку таких даних у потрібну систему згідно з ідентифікатором вузла. Усе це має сенс, поки ви не згадаєте, що одна лише мережа класу А містить 16 777 216 окремих IP-адрес. Це дуже багато пристроїв для підключення до одного маршрутизатора. Ось тут і допомагають підмережі. За їх допомогою ви можете розділити свою велику мережу на багато менших. Усі ці окремі підмережі матимуть власні маршрутизатори шлюзу, що слугуватимуть точкою входу та виходу для кожної підмережі.





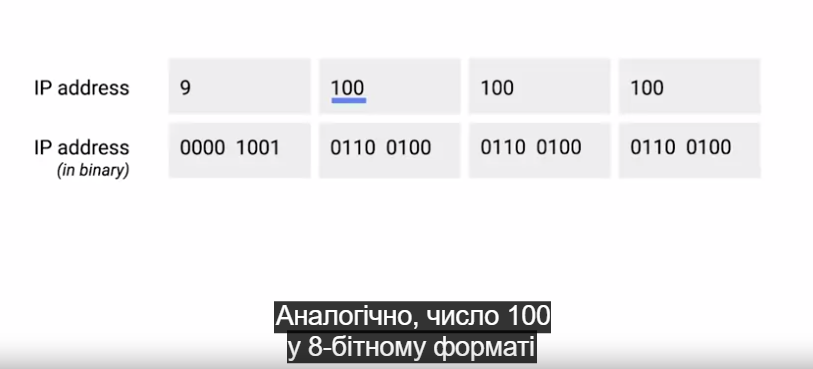


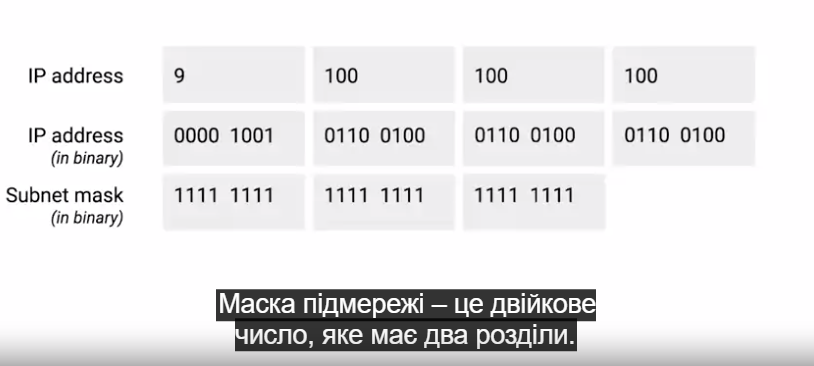
# Маски підмережі

Наразі ми вже дізналися про ідентифікатори мережі, які використовуються для ідентифікації мереж, та ідентифікатори вузлів, що ідентифікують окремі вузли. Якщо ми хочемо ще більшої деталізації (а ми хочемо), нам потрібно ввести третє поняття – ідентифікатор підмережі. Можливо, ви пам’ятаєте, що IP-адреса – це лише 32-бітне число. У світі без підмереж певна кількість цих бітів використовується для ідентифікатора мережі і певна кількість – для ідентифікатора вузла. У світі з підмережами деякі біти, які в іншому разі містили б ідентифікатор вузла, використовуються для ідентифікатора підмережі. З усіма цими трьома ідентифікаторами, які можуть бути представлені однією IP-адресою, ми отримуємо єдине 32-бітне число, яке може точно доставлятися в багатьох різних мережах. На рівні Інтернету магістральним маршрутизаторам потрібен лише ідентифікатор мережі, за допомогою якого вони надсилають дейтаграму на відповідний маршрутизатор шлюзу в цій мережі. Цей маршрутизатор шлюзу має деяку додаткову інформацію, за допомогою якої він надсилає цю дейтаграму на цільовий комп’ютер або наступний маршрутизатор на шляху до нього. Нарешті, останній маршрутизатор використовує ідентифікатор вузла, щоб доставити дейтаграму на вказаний комп’ютер одержувача. Ідентифікатори підмереж обчислюються за допомогою так званої маски підмережі. Як і IP-адреса, маски підмережі – це 32-бітні числа, які сьогодні зазвичай записуються як чотири октети десяткових чисел. Найпростіший спосіб зрозуміти, як працюють маски підмережі, – порівняти їх з IP-адресою. Увага! Зараз буде трохи складно. Зараз ми розглянемо досить важкий матеріал, але він надважливий, щоб належно зрозуміти, як працюють маски підмереж, бо їх дуже часто розуміють неправильно. Маски підмереж часто вважають просто магічними числами. Люди просто запам’ятовують деякі найпоширеніші з них, повністю не розуміючи, що за ними стоїть. Але ж у цьому курсі ми прагнемо, щоб ви отримали універсальну підготовку у сфері мереж. Тому, навіть якщо спершу маски підмереж можуть здаватися складними, проявіть трохи терпіння – і ви швидко їх зрозумієте. Просто знайте, що в наступному відео ми розглянемо деякі додаткові основи двійкової математики. Пройшовши весь матеріал, перегляньте це відео вдруге або втретє. Рухайтеся у власному темпі, і ви все опануєте за оптимальний для вас час. Давайте знову попрацюємо з IP-адресою 9.100.100.100. Ви, мабуть, пам’ятаєте, що кожна частина IP-адреси є октетом, що означає, що вона складається з восьми бітів. Число 9 у двійковому форматі записується як 1001. Але оскільки кожному октету потрібно 8 бітів, ми маємо додати спереду кілька нулів. Якщо йдеться про IP-адресу, то число 9 у першому октеті фактично представляється як 0000 1001. Аналогічно, число 100 у 8-бітному форматі представляється як 0110 0100. Отже, повне двійкове представлення IP-адреси, а саме 9.100.100.100, – це багато одиниць і нулів. Маска підмережі – це двійкове число, яке має два розділи. Перша частина, яка є власне маскою, – це рядок одиниць, після якого стоять нулі. Маска підмережі, тобто частина числа лише з одиниць, говорить нам, що ми можемо ігнорувати під час обчислення ідентифікатора вузла. Частина лише з нулів говорить нам, що слід залишити. Скористаймося поширеною маскою підмережі 255.255.255.0. Її можна представити як 24 одиниці, за якими слідують 8 нулів. Мета маски або тієї частини, що складається з одиниць, – повідомити маршрутизатору, яка частина IP-адреси є ідентифікатором підмережі. Згадайте: ми вже знаємо, як отримати ідентифікатор мережі для IP-адреси. Для мережі 9.100.100.100 класу А ми знаємо, що це просто перший октет. У нас залишаються три останні октети. Візьмімо ці октети й уявімо, що вони стоять після маски підмережі у двійковому форматі. Числа в решті октетів, яким відповідає одиниця в масці підмережі, будуть ідентифікатором підмережі. А числа в решті октетів, яким відповідає нуль, будуть ідентифікатором вузла. Розмір підмережі повністю визначається її маскою підмережі. Так, наприклад, з маскою підмережі 255.255.255.0 ми знаємо, що лише останній октет доступний для ідентифікаторів вузлів, незалежно від розміру ідентифікаторів мережі та підмережі. Одне 8-бітне число може представляти 256 різних чисел або, точніше, числа від 0 до 255. Саме час зазначити, що загалом підмережа зазвичай може містити на 2 ідентифікатори вузлів менше, ніж їхня загальна кількість. Знову ж таки, використовуючи маску підмережі 255.255.255.0, ми знаємо, що октет, доступний для ідентифікаторів вузлів, може містити числа від 0 до 255, але нуль зазвичай не використовується, а 255 зазвичай зарезервовано як широкомовну адресу для підмережі. Це означає, що насправді для призначення вузла доступні лише числа від 1 до 254. Хоча цей підхід "загальна кількість – 2" майже завжди відповідає дійсності, загалом ви будете називати повну кількість вузлів, доступних у підмережі. Отже, навіть якщо зрозуміло, що 2 адреси недоступні для призначення, ви все одно скажете, що 8 бітів місця для ідентифікаторів вузлів дають 256 адрес, а не 254. Це тому, що інші IP-адреси – це також IP-адреси, навіть якщо вони не призначаються вузлу в цій підмережі. Тепер давайте розглянемо маску підмережі, яка не займає цілий октет, тобто 8 бітів адреси. Маска підмережі 255.255.255.224 представляється як 27 одиниць, за якими слідують 5 нулів. Це означає, що у нас є 5 бітів місця для ідентифікаторів вузлів, або загалом 32 адреси. Це звужує діапазон для написання масок підмережі. Скажімо, ми маємо справу з нашим давнім другом – 9.100.100.100 з маскою підмережі 255.255.255.224. Оскільки ця маска підмережі представлена як 27 одиниць і 5 нулів, швидший спосіб визначити її – представити як 27. Усю IP-адресу й маску підмережі тепер можна записати як 9.100.100.100/27. Кожне з цих представлень не є обов’язково поширенішим, ніж інше, тому важливо розуміти обидва. Ми пройшли багато матеріалу. За потреби поверніться й перегляньте це відео знову. Якщо ж усе зрозуміло, переходьте до наступного відео про основи двійкової математики. Побачимося!

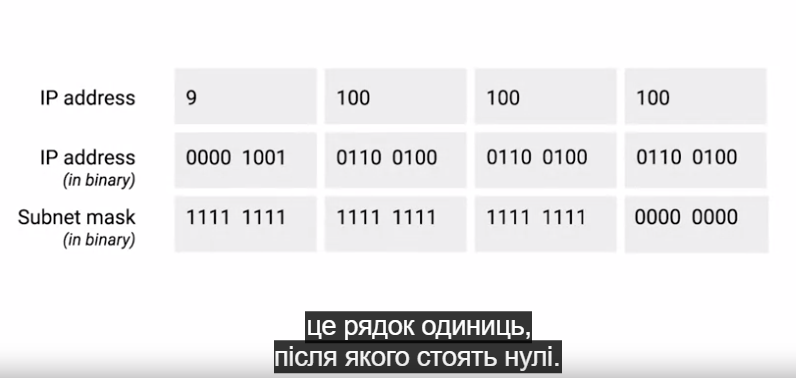








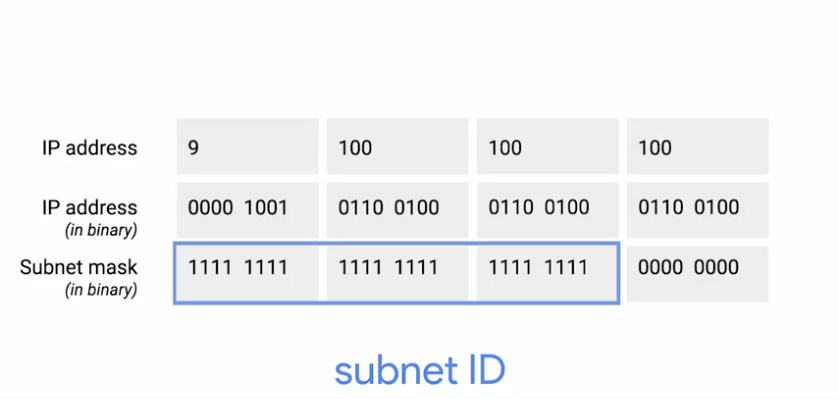


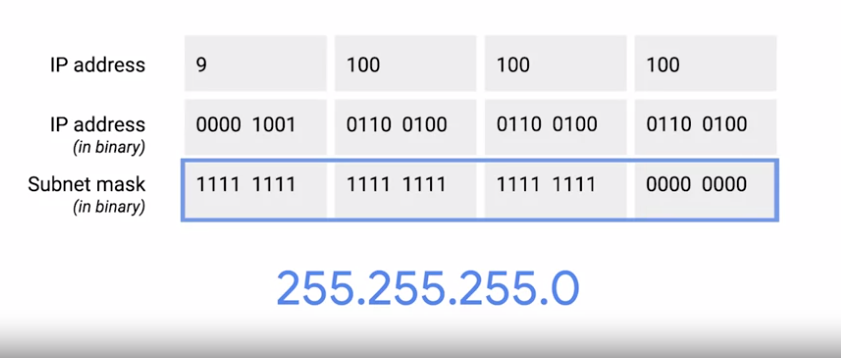


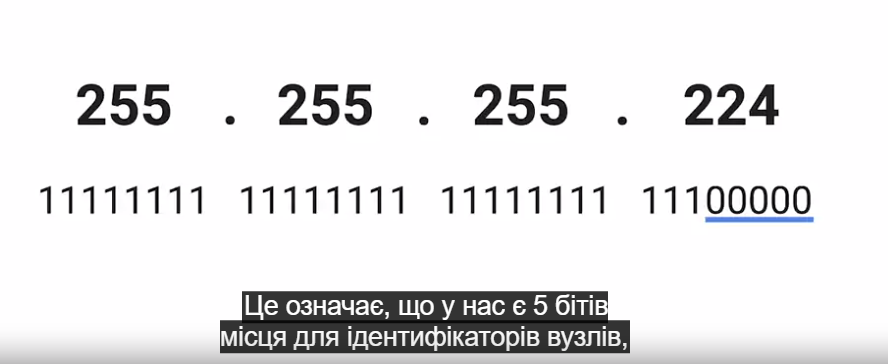


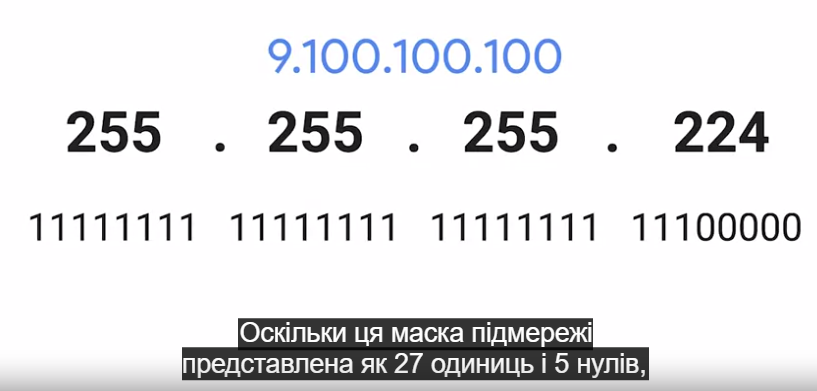


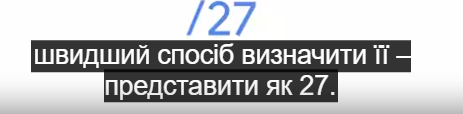




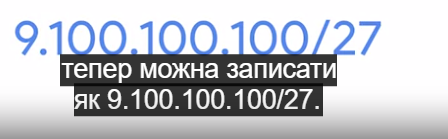












# Базові двійкові обчислення

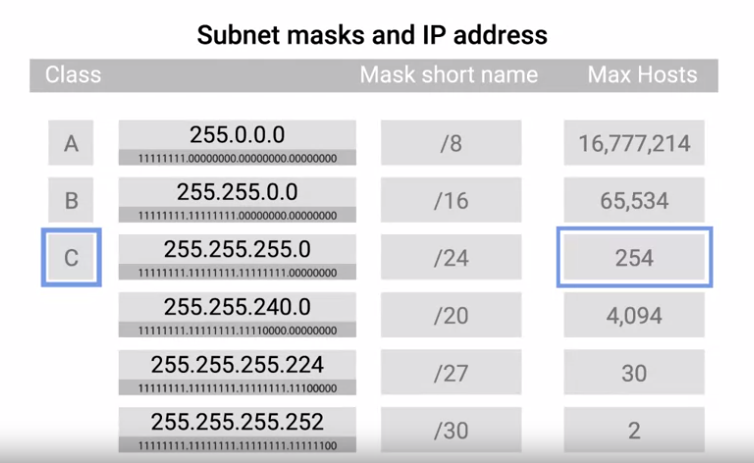
Двійкові числа спочатку можуть вас злякати, адже вони так відрізняються від десяткових чисел. Проте якщо ми говоримо про основні математичні операції, як-от лічба, додавання або віднімання двійкових чисел, то тут усе точно так само, як і з десятковими. Важливо зазначити, що не існує різних типів чисел. Числа універсальні. Є лише різні визначення їхніх назв. Люди, ймовірно тому, що більшість з нас мають десять пальців, вирішили використовувати систему з десяти цифр, з яких складаються всі числа.Цифри 0, 1, 2, 3, 4, 5, 6, 7, 8 і 9 можуть комбінуватись будь-яким чином і складати будь-яке ціле число. Оскільки в десятковій системі використовується 10 цифр, то її ще також називають основою 10. Через обмеження, спричинені специфікою роботи логічних затворів усередині процесора, комп’ютерам набагато простіше оперувати лише нулями та одиницями. Ця система відома як бінарна, або двійкова. Ви можете подавати всі цілі числа однаково як у двійковій, так і в десятковій системі. Просто виглядають вони трохи по-різному. Коли ви рахуєте в десятковій системі, ви рухаєтесь по цифрах за наростаючою. А коли закінчуєте, додаєте другий стовпчик із більшим значенням. Почнімо рахувати з 0, доки не дійдемо до 9. Щойно ми дійдемо до дев’яти, ми просто почнемо заново. Ми додаємо 1 у новий стовпчик і знову починаємо рахунок з 0 в початковому стовпці. Ми повторюємо цей процес знову й знову, щоб підрахувати всі цілі числа. Підрахунок у двійковій системі точно такий самий, просто у вас є лише дві цифри. Ви починаєте з 0, який дорівнює 0 у десятковій системі. Потім ви додаєте 1. Тепер у вас 1. Така ж як і в десятковій системі. Оскільки в нас уже закінчилися цифри, треба додати новий стовпець. Отже, тепер ми маємо число 1 0 Що те ж саме, що й 2 в десятковій системі, 1 1 – це 3, 1 0 0 – це 4, 1 0 1 – це 5, 1 1 0 – це 6, 1 1 1 – це сім тощо. Це те ж саме, що ми робимо з десятковим числом, просто ми маємо менше цифр. При роботі з різною обчислювальною технікою ви часто стикатиметеся з поняттям бітів або одиниць і нулів. Є досить простий прийом, щоб з’ясувати, скільки десяткових чисел можна виразити певною кількістю бітів.Почати відтворення відео о :2:37 і дотримуватися тексту2:37Якщо у вас є 8-бітове число, ви можете просто піднести 2 до восьмого степеня. Ви отримаєте 256, а це означає, що 8-бітне число може являти собою 256 десяткових чисел, або, інакше кажучи, числа від 0 до 255. 4-бітове число дорівнює 2 у четвертому степені або 16 чисел. 16-бітове число дорівнює 2 у 16 степені або 65536 чисел. Щоб прив’язати це до того, що ви вже, можливо, знаєте, цей трюк працює не тільки для двійкової, але й для будь-якої іншої системи чисел. Змінюється лише система числення. Ви, можливо, пам’ятаєте, що двійкова система має за основу 2, і десяткова − 10. Усе, що потрібно зробити, це замінити основу на кількість стовпців. Наприклад, візьмемо число основи 10 із двома стовпцями цифр. Це означає 10 у другому степені. 10 у другому степені дорівнює 100, і саме стільки чисел можна представити двома стовпцями десяткових чисел або числами від 0 до 99. Аналогічно, 10 у третьому степені дорівнює 1000, і саме стільки чисел можна виразити 3-ма стовпцями десяткових чисел або числами від 0 до 999. У різних системах не тільки лічба однакова, а й прості арифметичні дії, як-от додавання. Насправді двійкове додавання навіть простіше, ніж у будь-якій іншій системі, оскільки ми маємо 4 можливі сценарії. 0+0=0 як і в десятковій системі, 0+1=1, і 1+0=1 − це вам теж знайомо. 1+1=10 − тут трохи по-іншому, але й у цьому є своя логіка. У десятковій системі ви переносите цифру до наступного стовпця, як тільки досягаєте значення 10. У двійковій ви переносите цифру до наступного стовпця, як тільки досягаєте 2. Додавання − це оператор, і є багато операторів, які комп’ютери використовують для виконання обчислень. Два з найважливіших операторів − АБО та І. У комп’ютерній логіці 1 виражає істину, а 0 − хибність.Почати відтворення відео о :4:57 і дотримуватися тексту4:57Робота оператора АБО полягає в тому, що ви дивитеся на кожну цифру, і якщо одна з них приймає значення "Істина", результат приймає таке саме значення. Базове рівняння − X АБО Y = Z, що також можна прочитати як "якщо X або Y є істинним, то й Z є істинним, в іншому випадку − це значення хибне". Тому 1 АБО 0 = 1, але 0 АБО 0 = 0. Оператор І, як зрозуміло з його назви, повертає значення "Істина" тільки коли обидва значення істинні. Тому 1 І 1 = 1, але 1 І 0 = 0, а 0 І 0 = 0 тощо. Вам може бути цікаво, для чого це нам. Ні, ми не намагаємось вас заплутати. Усе це дійсно допомагає більше пояснити маски підмережі. Маска підмережі − це спосіб використання комп’ютером операторів І, щоб визначити, чи IP-адреса існує в тій самій мережі. Це означає, що частина ID хосту також відома, а отже нічого не залишиться поза увагою. Використаймо двійкову систему нашої улюбленої IP-адреси 9.100.100.100 і нашу улюблену маску підмережі 255.255.255.0. Після того, як ви помістите одну над іншою й виконаєте двійковий оператор І на кожному стовпці, ви помітите, що результатом є ID мережі й частина ID підмережі нашої IP-адреси або 9.100.100. Комп’ютер, що виконав цю операцію, тепер може порівнювати результати з власним ID мережі, щоб визначити, чи перебуває адреса теж в тій самій мережі чи в іншій. Б’юсь об заклад, що ви ніколи не думали, що у вас буде улюблена IP-адреса чи підмережа. Але трапляється в чудовому світі базової двійкової математики.

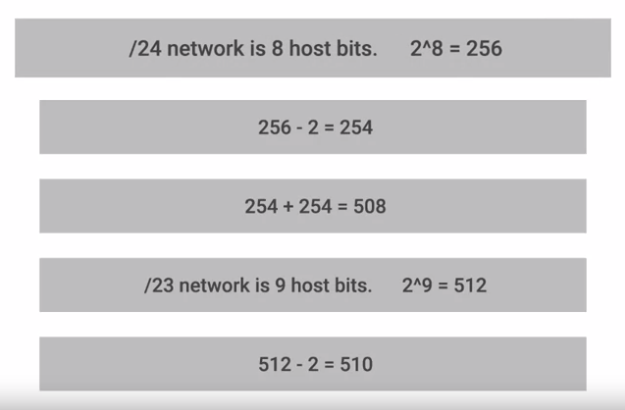


Вітаю, ви правильно розумієте, що 2^8 = 256. Однак, лише 254 адреси в мережі класу C доступні для використання.  
Перша адреса в мережі класу C зарезервована для **маршрутизатора**, який зв'язує мережу з іншими мережами. Друга адреса також зарезервована для використання в якості **мережевої маски**.

# CIDR

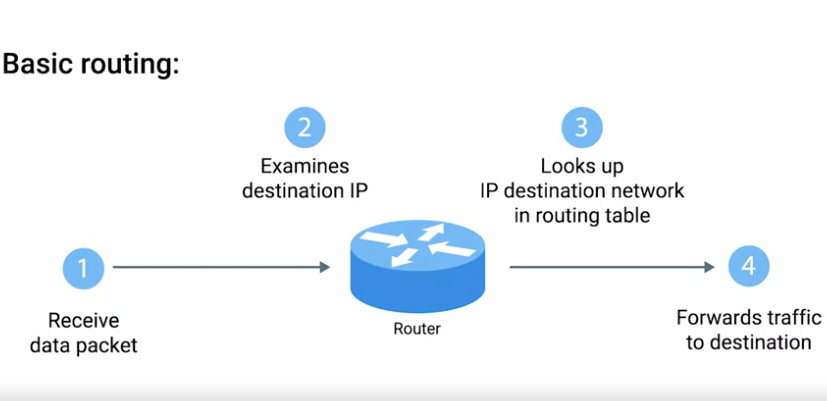
Класи адрес були першою спробою поділу глобального інтернет-простору IP-адрес. Поділ на підмережі запровадили, коли стало зрозуміло, що класи адрес не були ефективним способом упорядкування. Інтернет продовжував розширюватись, а традиційний поділ на підмережі не встигав за його ростом. За традиційним поділом на підмережі та класами адрес ідентифікатор мережі завжди є 8-бітовим для мереж класу А, 16-бітовий − для мереж класу B, або 24-бітовий для мереж класу C. Це означає, що може існувати лише 254 хости, що також говорить про існування 2970152 потенційних мереж класу С. Це багато компонентів у таблиці маршрутизації. До того ж розміри цих мереж не завжди відповідають потребам більшості підприємств. 254 хостів у мережі класу С замалі для багатьох випадків використання, але 65534 хости, доступні для використання в мережі класу B, часто занадто великі. Багато компаній зупинилися на варіанті декількох суміжних мереж класу С для задоволення своїх потреб. Це означало, що в таблиці маршрутизації опинялася купа записів для купи мереж класу С, які, по суті, переадресовувалися в однакове місце. Тут вступає в гру CIDR або безкласова міждоменна маршрутизація. CIDR − це ще гнучкіший підхід для опису блоків IP-адрес. Він розширює концепцію поділу підмереж, використовуючи маски підмереж для демаркації мереж. Демаркувати щось означає розмежувати. Коли йдеться про комп’ютерні мережі, ви будете часто чути термін "точка розмежування" − місце, де закінчується одна мережа або система й починається інша. У нашій попередній моделі ми покладалися на ідентифікатор мережі, ідентифікатор підмережі та ідентифікатор хосту, щоб доставити IP-дейтаграму в правильне місце. При CIDR ідентифікатор мережі та ідентифікатор підмережі, об’єднані в одне ціле. CIDR − це там, де ми використовуємо скорочену слеш-нотацію, яку ми обговорювали раніше у відео про поділ на підмережі. Ця слеш-нотація також відома як нотація CIDR. CIDR в основному повністю відмовляється від концепції класів адрес, дозволяючи визначати адресу лише за двома індивідуальними ідентифікаторами. Візьмімо 9.100.100.100 з мережевою маскою 255.255.255.0. Пам’ятайте, що це також можна записати як 9.100.100.100/24. Тепер, коли нам більше не важливий клас цієї IP-адреси, все, що нам потрібно це те, що нам каже мережева маска − визначити ідентифікатор мережі. У цьому випадку це буде 9.100.100, ідентифікатор хосту залишається тим самим. Така практика не тільки спрощує те, як маршрутизатори й інші мережеві пристрої повинні сприймати частини IP-адреси, але він також допускає більш довільні розміри мережі. До цього розміри мереж були статичними. Тільки клас А, клас В або клас С і тільки підмережі могли бути різного розміру. CIDR дозволяє самим мережам бути різного розміру. До цього, якщо компанії треба було більше адрес, ніж міг надати один клас С, їй потрібен був цілий другий клас C. Із CIDR вони можуть об’єднати цей адресний простір в один суцільний шматок із мережевою маскою /23 або 255.255.254.0. Це означає, що тепер маршрутизаторам треба знати не два, а лиш один компонент у своїй таблиці маршрутизації, щоб доставити трафік на ці адреси. Також важливо зазначити, що ви отримуєте додаткові доступні ідентифікатори хостів під час такої практики. Пам’ятайте, що ви завжди втрачаєте 2 ідентифікатори хостів у кожній мережі. Отже, якщо в мережі /24 є від двох до восьми або 256 потенційних хостів, насправді у вас тільки 256 − 2, або 254 доступних для призначення IP-адрес. Якщо вам потрібні дві мережі такого розміру, у вас загалом є 254 + 254 або 508 хостів. Єдина /23 мережа, з іншого боку, це 2 в девʼятому степені, або 512. 512 − 2 = 510 хостів. Зупиніться на мить і зафіксуйте це в пам’яті. А коли ви будете готові, ми пройдемо короткий тест без оцінювання перед тим, як перейти до маршрутизації на наступному уроці.



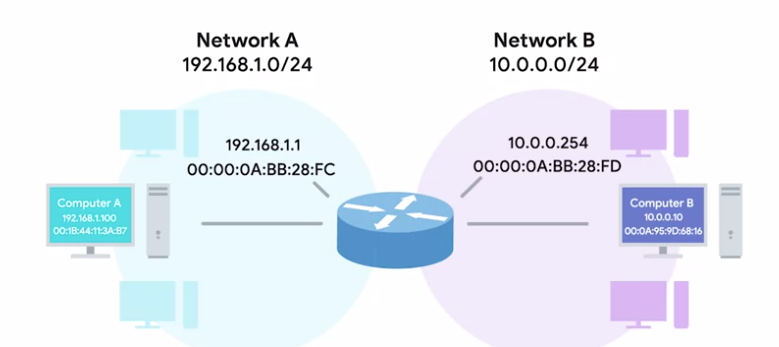


# Основні поняття маршрутизації

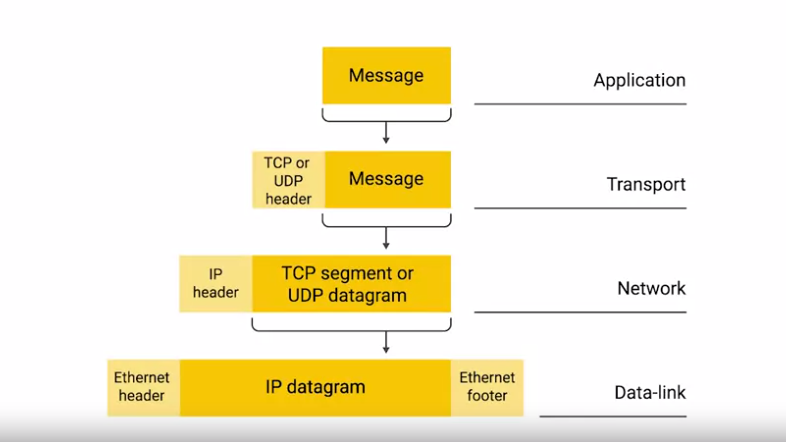
Інтернет – це надзвичайне технологічне досягнення. Він обʼєднує мільйони окремих мереж та забезпечує зв’язок між ними. З майже будь-якої точки світу ви можете отримати доступ до даних, що знаходяться в практично будь-де. Часто всього за долю секунди. Спосіб, у який відбувається обмін даними по цих мережах, і який дозволяє отримувати дані з іншого куточка планети, називається маршрутизацією. У кінці уроку ви зможете описати основи маршрутизації й роботу таблиць маршрутизації, дізнаєтеся про деякі основні протоколи маршрутизації й про те, що вони роблять. Будете знати, що таке немаршрутизований адресний простір, і як ним користуються. Ви також зрозумієте, що таке система RFC і як вона зробила Інтернет таким, яким він є зараз. Усі ці навички необхідні для усунення несправностей у мережі, які зустрічаються в роботі фахівця з ІТ-підтримки. Маршрутизація – це одна з речей, що прості й складні водночас. На дуже високому рівні насправді просто зрозуміти, що таке маршрутизація та як працюють маршрутизатори. Але якщо "зазирнути під капот", маршрутизація – це дуже складна й технологічно просунута тема. На цю тему писали цілі книги. Сьогодні найбільш гострими питаннями маршрутизації майже виключно займаються інтернет-провайдери (ISPs), і лише найбільші з них. Ми дамо вам базовий огляд маршрутизації, щоб ви володіли всебічними знаннями про мережу. Оскільки це обовʼязкова й важлива для розуміння тема. Проте наш матеріал, звичайно, не є вичерпним. Якщо дуже спрощено, маршрутизатор – це мережевий пристрій, що пересилає трафік залежно від адреси призначення цього трафіку. Це пристрій, який має принаймні два мережеві інтерфейси, оскільки повинен бути підключений до двох мереж, щоб виконувати свою роботу. Базова маршрутизація має лише кілька кроків. Перший – маршрутизатор отримує пакет даних на одному з інтерфейсів. Другий – маршрутизатор визначає IP-адресу призначення цього пакету. Третій – маршрутизатор шукає цільову мережу цієї IP-адреси у своїй таблиці маршрутизації. Четвертий – маршрутизатор передає трафік через найближчий до віддаленої мережі інтерфейс, як визначено додатковою інформацією в таблиці маршрутизації. Ці кроки повторюються доти, доки трафік не досягне адреси призначення.



Уявімо маршрутизатор, підключений до двох мереж. Назвімо першу мережею А, і дамо їй адресний простір 192.168.1.0/24. Друга мережа – мережа B із адресним простором 10.0.0.0/24. Маршрутизатор має інтерфейс у кожній мережі. У мережі А він має IP 192.168.1.1 А в мережі B його IP-адреса – 10.0.0.254. Пам’ятайте, що IP-адреси належать мережам, а не окремим вузлам у мережі.



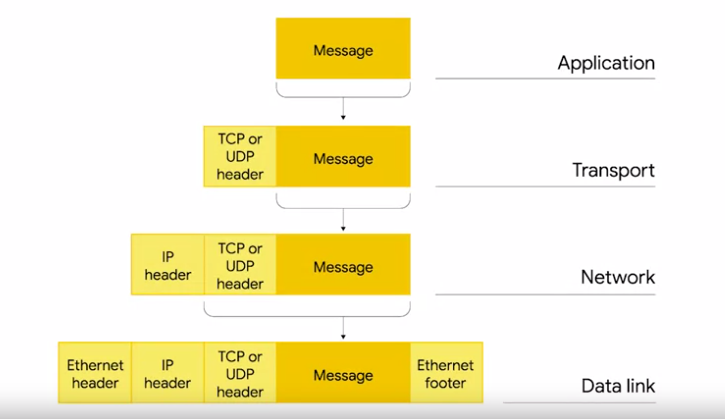
Комп’ютер у мережі A з IP-адресою 192.168.1.100 надсилає пакет на адресу 10.0.0.10. Цей комп’ютер знає, що 10.0.0.10 знаходиться в локальній підмережі. Таким чином, він надсилає цей пакет на MAC-адресу свого шлюзу (маршрутизатора).

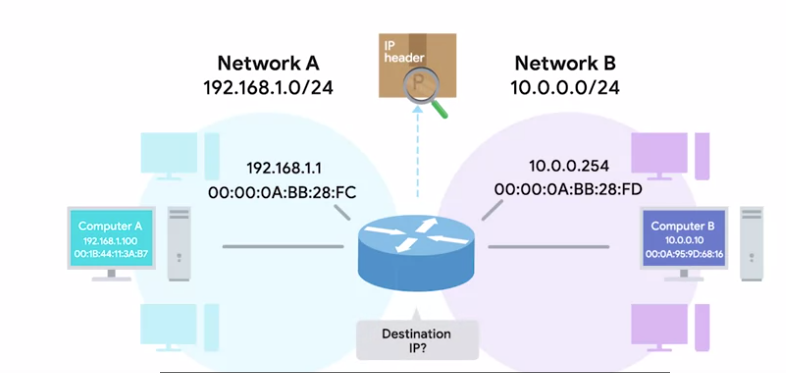


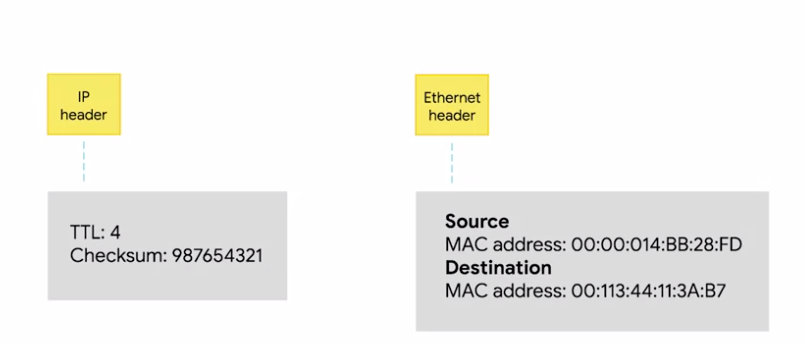
Інтерфейс маршрутизатора в мережі А отримує пакет, тому що бачить, що MAC-адреса призначення належить йому. Потім маршрутизатор прибирає інкапсуляцію канального рівня, залишаючи контент мережевого рівня – IP-дейтаграму.



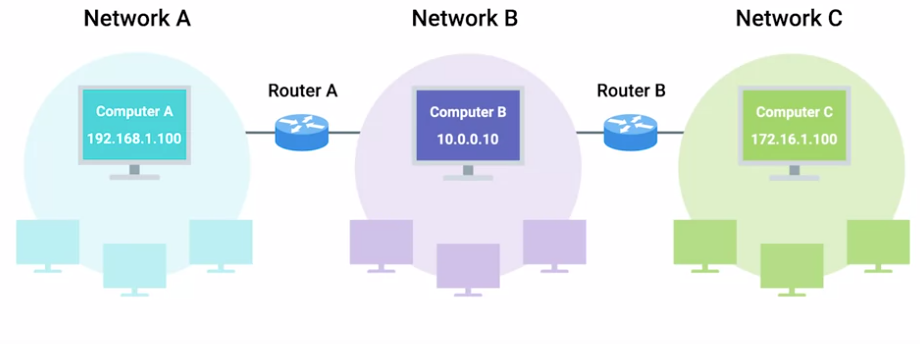
Тепер маршрутизатор може напряму перевірити заголовок IP-дейтаграми на предмет поля IP-адреси призначення. Він знаходить IP-адресу призначення 10.0.0.10. Маршрутизатор дивиться в таблицю маршрутизації та бачить, що мережа B, або мережа 10.0.0.0/24, є правильною мережею для IP-адреси призначення. Він також бачить, що ця мережа перебуває на відстані одного транзитного шлюзу. І оскільки вони поєднані напряму, маршрутизатор навіть має MAC-адресу для цієї IP-адреси у своїй ARP-таблиці. Далі маршрутизатору потрібно сформувати новий пакет, щоб переслати його мережі B. Він бере всі дані з першої IP-дейтаграми й дублює їх, але зменшує TTL-поле на 1 та обчислює нову контрольну суму. Потім він інкапсулює цю нову IP-дейтаграму всередині нового кадру Ethernet. Цього разу він встановлює свою MAC-адресу інтерфейсу в мережі B як вихідну MAC-адресу. Оскільки він має MAC-адресу 10.0.0.10 у своїй ARP-таблиці, він встановлює її як MAC-адресу призначення. Нарешті, пакет відправляється з інтерфейсу в мережі B, і дані нарешті доставляються до вузла у 10.0.0.10.







Це досить простий приклад того, як працює маршрутизація, але зараз ми його дещо ускладнимо, додавши третю мережу. Усе інше залишається незмінним. У нас є мережа А, адресний простір якої – 192.168.1.0/24. У нас є мережа B з адресним простором 10.0.0/24. Маршрутизатор який зʼєднує ці дві мережі, так само має IP-адреси: 192.168.1.1 – у мережі А та 10.0.0.254 – у мережі B. Але додаймо третю мережу – мережу C. Вона має адресний простір 172.16.1.0/23. Є другий маршрутизатор, що з’єднує мережі В і C. Його інтерфейс в мережі B має IP-адресу 10.0.0.1, а його інтерфейс у мережі C має IP-адресу 172.16.1.1. Цього разу наш комп’ютер з IP 192.168.1.100 хоче надіслати певні дані на комп’ютер, що має IP 172.16.1.100. Ми пропустимо матеріал про канальний рівень, але ви, звичайно, пам’ятаєте, що він також є. Комп’ютер з IP 192.168.1.100 знає, що 172.16.1.100 не перебуває у своїй локальній мережі. Тому він посилає пакет на свій шлюз – маршрутизатор між мережами A та B. І знову маршрутизатор перевіряє контент цього пакета, бачить адресу призначення 172.16.1.100 і після пошуку у своїй таблиці маршрутизації він визначає, що найшвидший спосіб звʼязатися з мережею 172.16.1.0/23 – це зʼєднатися з нею через інший маршрутизатор з IP 10.0.0.1. Маршрутизатор зменшує TTL-поле й передає його на маршрутизатор 10.0.0.1. Потім цей маршрутизатор проходить через все те ж саме, він знає, що IP-адреса призначення 172.16.100 напряму зʼєднана, і пересилає пакет до його кінцевого місця призначення. Це основи маршрутизації. Єдина відмінність між нашими прикладами та тим, як все працює в Інтернеті, – це масштаби. Зазвичай маршрутизатори підключені до значно більшої кількості мереж. Дуже часто трафіку доводиться пройти через десяток маршрутизаторів, щоб досягнути кінцевого місця призначення. І нарешті, з метою захисту від збою, основні інтернет-маршрутизатори зазвичай поєднані в сітку. Це означає, що може бути багато різних шляхів передачі пакета. Тим не менш, усі поняття однакові. Маршрутизатори перевіряють IP-адресу призначення, шукають її в таблиці маршрутизації, щоб визначити найшвидший шлях передачі, і вже по ньому пересилають пакет. Це відбувається знову й знову з кожним пакетом, що міститься в кожному біті трафіку. У всьому Інтернеті. Постійно. Круто, чи не так?



# Таблиці маршрутизації

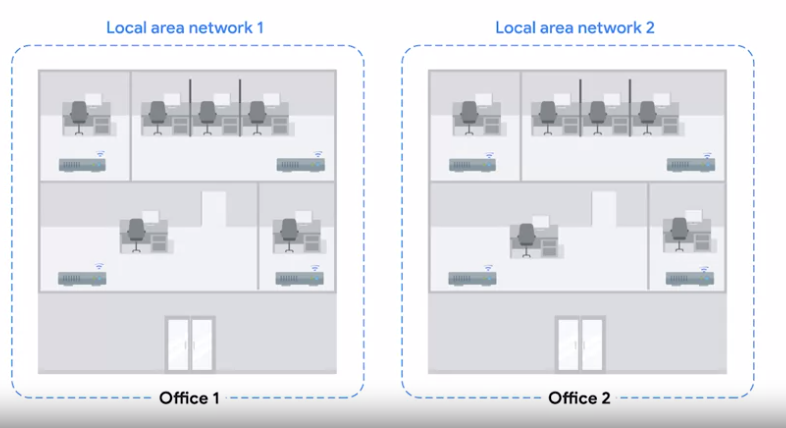
Під час перегляду нашого попереднього відео про основи маршрутизації у вас, мабуть, виникло багато питань про "таблицю маршрутизації". Сама по собі маршрутизація – досить просте поняття. Таблиці маршрутизації не набагато складніші. Перші маршрутизатори були просто звичайними комп’ютерами тієї епохи. Вони мали два мережеві інтерфейси, мережевий міст і таблицю маршрутизації, яка оновлювалась вручну. Насправді, усі основні сучасні операційні системи також мають таблицю маршрутизації, з якою звіряються перед передачею даних. Ви все ще змогли б сьогодні створити власний маршрутизатор, якби мали комп’ютер із двома мережевими інтерфейсами і він оновлював таблицю маршрутизації вручну. Таблиці маршрутизації можуть бути дуже різними залежно від марки та класу маршрутизатора, але всі вони мають кілька спільних речей. Найпростіша таблиця маршрутизації матиме чотири стовпці. "Мережа призначення" – цей стовпець містить по рядку для кожної мережі, про яку відомо маршрутизатору. Це просто визначення віддаленої мережі, ідентифікатора мережі та маски мережі. Вони можуть зберігатися в одному стовпці всередині представлення, або ідентифікатор і маска мережі можуть бути в окремому стовпці. Так чи інакше, це та сама концепція: маршрутизатор має визначення для мережі й тому знає, які IP-адреси можуть до неї належати. Коли маршрутизатор отримує вхідний пакет, він перевіряє IP-адресу призначення і визначає, до якої мережі вона належить. Таблиця маршрутизації зазвичай має універсальний запис, який охоплює всі IP-адреси без чіткого визначення мережі. "Наступний пункт" – це IP-адреса наступного маршрутизатора, який має отримувати дані, призначені для відповідної мережі призначення. Або тут може просто зазначатися, що мережа зв’язана напряму і додаткових кроків не потрібно.



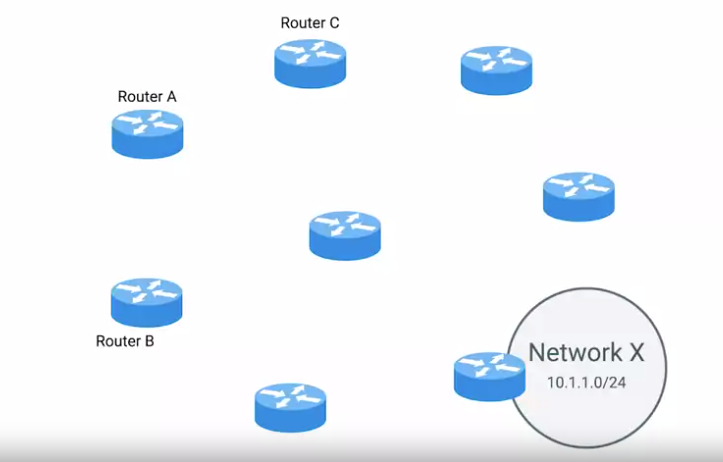
"Усього кроків" – найважливіша частина для розуміння маршрутизації й того, як працюють таблиці маршрутизації. У будь-якій складній мережі, як-от Інтернет, існує багато шляхів, щоб дістатися від точки А до точки В. Маршрутизатори завжди намагаються вибрати найкоротший шлях, щоб забезпечити своєчасне доставлення даних, але найкоротший шлях до мережі призначення може з часом змінюватися, й іноді дуже швидко. Проміжні маршрутизатори можуть виходити з ладу, може втрачатися з’єднання, можуть додаватися нові маршрутизатори, перевантаження мереж уповільнювати певні маршрути. У наступних відео ми дізнаємося, як маршрутизатори визначають найкоротший шлях. Наразі просто пам’ятайте, що для кожного наступного кроку та кожної мережі призначення маршрутизатору доведеться визначати, як далеко цей пункт призначення зараз. Таким чином, коли він отримує оновлену інформацію від сусідніх маршрутизаторів, він знає, чи його поточний маршрут оптимальний, чи є новий кращий маршрут. "Інтерфейс" – маршрутизатор також повинен знати, з якого зі своїх інтерфейсів пересилати трафік відповідно до мережі призначення. У більшості випадків таблиці маршрутизації досить прості. Насправді вражає те, що в багатьох магістральних маршрутизаторах таблиці маршрутизації містять мільйони рядків. Вони мають переглядатися для кожного окремого пакета, який проходить через маршрутизатор на шляху до кінцевого пункту призначення. Також вражає те, як багато ви дізналися про маршрутизатори, маршрутизацію та таблиці маршрутизації. Молодці! Побачимося в наступному відео про протоколи внутрішніх шлюзів.

# Внутрішні протоколи шлюзу

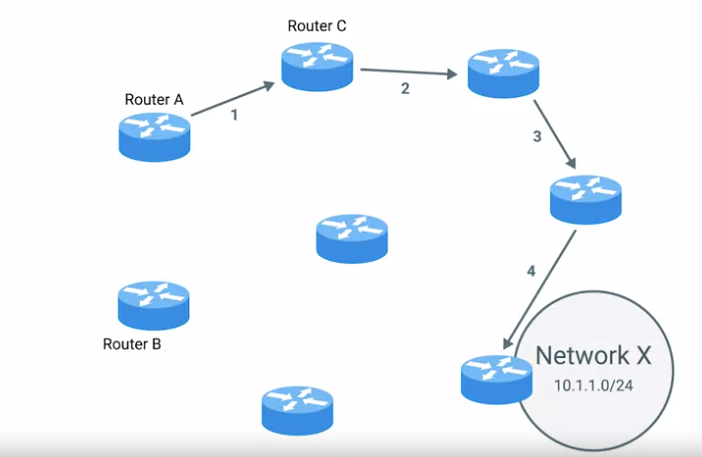
Ми розглянули основи роботи маршрутизації й побудову таблиць маршрутизації. Ці два поняття досить базові. Справжня магія маршрутизації в тому, що в таблицях маршрутизації завжди оновлюється інформація про найшвидші шляхи до мереж призначення. Протоколи, про які ми дізнаємося в цьому відео, допоможуть вам визначити проблеми з маршрутизацією в будь-якій мережі. Щоб краще орієнтуватися в маршрутах, маршрутизатори використовують протоколи маршрутизації. Це спеціальні протоколи, через які маршрутизатори діляться інформацією. У такий спосіб маршрутизатор з одного куточка планети може дізнатися найкращий шлях до мережі на іншому кінці світу. Протоколи маршрутизації поділяються на дві основні категорії: протоколи внутрішніх шлюзів і протоколи зовнішніх шлюзів. Протоколи внутрішніх шлюзів поділяються на ще дві категорії: протоколи стану каналу звʼязку та дистанційно-векторні протоколи. У цьому відео ми розглянемо основне про протоколи внутрішніх шлюзів. Протоколи внутрішніх шлюзів використовуються маршрутизаторами для поширення інформації в рамках єдиної автономної системи. Коли говоримо про мережу, автономна система є сукупністю мереж, які підпорядковуються єдиному оператору мережі. Найкращий приклад цього − велика корпорація, що потребує маршрутизації даних між її багатьма офісами, і кожен з яких може мати власну локальну мережу.



Інший приклад − національний постачальник інтернет-послуг, який зазвичай використовує багато маршрутизаторів. Можна протиставити це протоколам зовнішніх шлюзів, що використовуються для обміну інформацією між самостійними автономними системами. Спойлер: ми розглянемо протоколи зовнішніх шлюзів у наступному відео. Два основні типи протоколів внутрішніх шлюзів – це протоколи стану каналу звʼязку та дистанційно-векторні протоколи. Їхні цілі надзвичайно схожі, але маршрутизатори, які їх використовують, поширюють дані різних типів, щоб виконати свою роботу. Дистанційно-векторні протоколи − це старіший стандарт. Маршрутизатор, що використовує такий протокол, загалом просто має свою таблицю маршрутизації, яка є списком усіх відомих йому мереж і відстаней, на яких ці мережі розташовані з огляду на транзитні лінії. Потім маршрутизатор надсилає цей список до кожного сусіднього маршрутизатора, а це кожен маршрутизатор, що безпосередньо підключений до нього. У комп’ютерних науках такий список відомий як вектор. Ось чому протокол, який просто надсилає список відстаней до мереж, відомий як дистанційно-векторний протокол. Дистанційно-векторний протокол дає маршрутизаторам не так багато даних про загальний стан автономної системи. Вони просто мають певну інформацію про своїх безпосередніх сусідів. Щоб зрозуміти, як працюють, дистанційно-векторні протоколи, погляньмо, як два маршрутизатори можуть впливати на таблиці один одного.

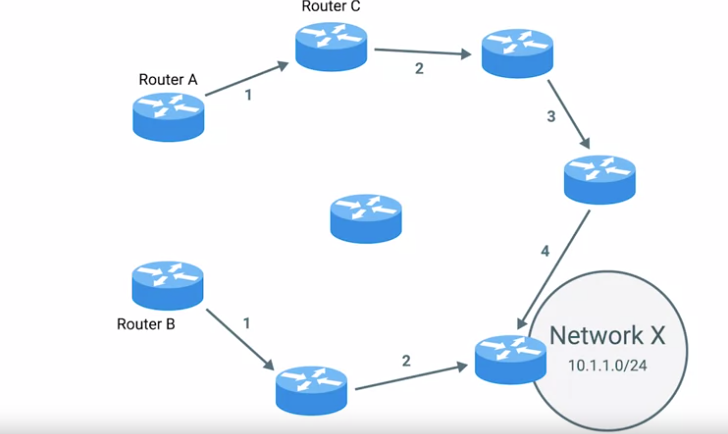


Маршрутизатор А має таблицю маршрутизації з групою компонентів. Один із цих компонентів належить мережі 10.1.1.0/24, яка називається мережею Х. Маршрутизатор А вважає, що найшвидший шлях до мережі X пролягає через його інтерфейс 2, куди підключений маршрутизатор C.

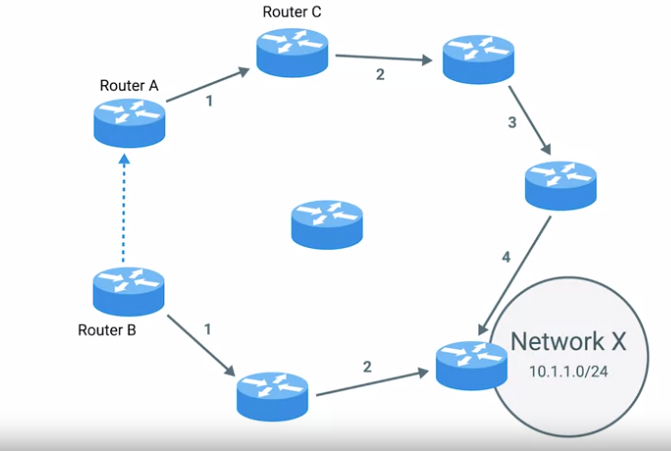


Маршрутизатор А знає, що надсилання даних до мережі X через інтерфейс 2 до маршрутизатора С означає, що треба буде пройти через 4 транзитні лінії, щоб дістатися місця призначення.

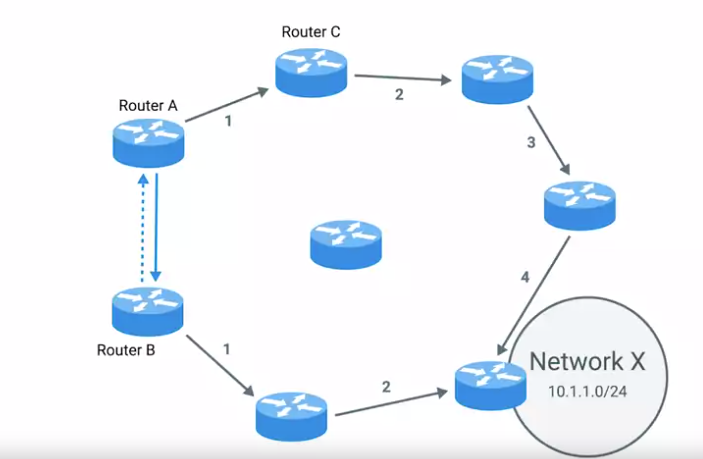
У той час як від маршрутизатора B до мережі Х лише 2 лінії.



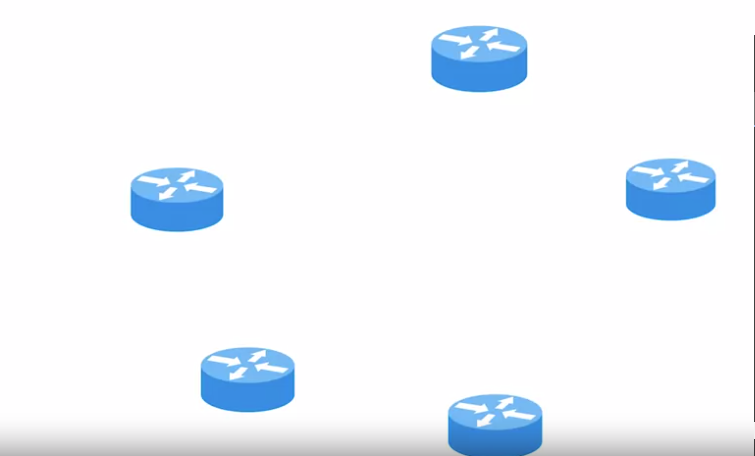
Це зазначено в його таблиці маршрутизації. Маршрутизатор B, використовуючи дистанційно-векторний протокол, надсилає основний вміст своєї таблиці маршрутизації маршрутизатору А.



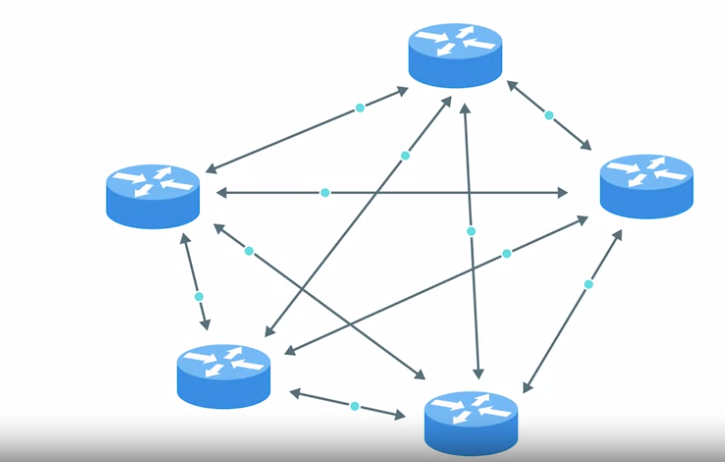
Маршрутизатор А бачить, що мережа X лише на відстані 2 транзитних ліній від Б. Навіть з додатковою лінією між маршрутизаторами А і В мережа Х буде на відстані всього трьох ліній від маршрутизатора А, якщо він надішле дані маршрутизатору В, а не С. Із цією новою інформацією маршрутизатор А оновлює свою таблицю маршрутизації, щоб відобразити це. Для доступу до мережі X якомога швидше він має переадресувати трафік через власний інтерфейс 1 до маршрутизатора B.



Отже, дистанційно-векторні протоколи досить прості, але вони не надають маршрутизатору достатньо інформації про ситуацію за межами їхніх прямих сусідів. Через це маршрутизатор може повільно реагувати на зміни, які відбуваються далеко за його межами. Ось чому врешті-решт придумали протоколи стану каналу звʼязку. Маршрутизатори, що використовують протокол стану каналу зв'язку, застосовують складніший підхід, щоб визначити найкращий шлях до мережі.



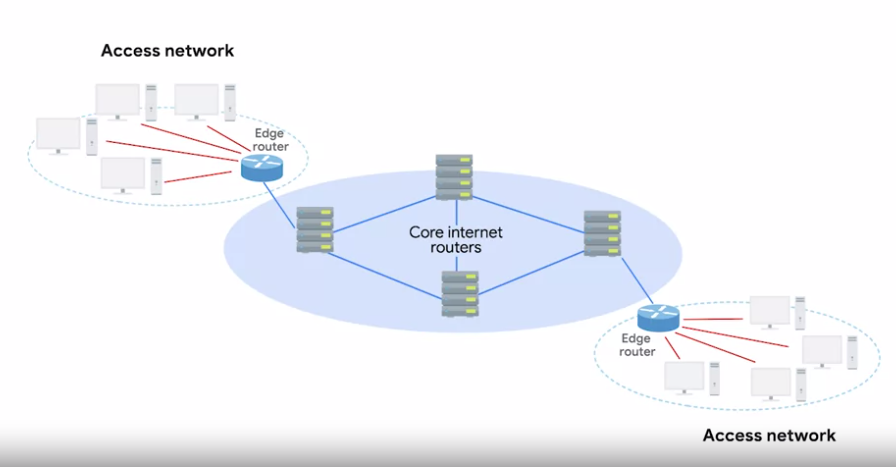
Протоколи стану каналу звʼязку отримали свою назву, тому що кожний маршрутизатор рекламує стан каналу звʼязку кожного зі своїх інтерфейсів.



Ці інтерфейси можуть бути підключені до інших маршрутизаторів або напряму до мереж. Інформація про кожен маршрутизатор поширюється на кожен інший маршрутизатор в автономній системі. Це означає, що кожен маршрутизатор у системі знає кожну деталь про будь-який інший маршрутизатор у системі. Кожен маршрутизатор потім використовує цей значно більший набір інформації та виконує складні алгоритми на його основі, щоб визначити, який шлях може бути найкращим до будь-якої мережі призначення. Протоколи стану каналу зв'язку потребують як більшої кількості пам'яті, щоб утримувати всі ці дані, так і значно більше обчислювальних потужностей. Це через те, що вони повинні виконувати алгоритми на основі цих даних з метою визначення найшвидшого шляху оновлення таблиць маршрутизації. Оскільки комп’ютерне обладнання з роками стало потужнішим та дешевшим, протоколи стану каналу зв’язку витіснили старі дистанційно-векторні протоколи. Далі ми поговоримо про протоколи зовнішніх шлюзів.

# Зовнішні протоколи шлюзу

Протоколи зовнішніх шлюзів використовуються для передачі даних між маршрутизаторами, які є межами автономної системи. Оскільки всі маршрутизатори, що обмінюються даними за допомогою протоколів внутрішніх шлюзів, перебувають у мережах тієї самої організації, то протоколи зовнішніх шлюзів маршрутизатори використовують, коли їм потрібно ділитися інформацією між різними організаціями. Протоколи зовнішніх шлюзів надзвичайно важливі в роботі сучасного Інтернету. Тому дякуємо вам, зовнішні протоколи шлюзу. Інтернет − це величезна сітка автономних систем.



На найвищих рівнях, головні інтернет-маршрутизатори мають знати про автономні системи з метою належної переадресації трафіку. Оскільки автономні системи відомі як колекції мереж, доставлення даних до пограничного маршрутизатора автономної системи є основним призначенням головних інтернет-маршрутизаторів. IANA, або Адміністрація адресного простору Інтернет, є некомерційною організацією, яка допомагає керувати такими діями, як виділення IP-адрес. Інтернет не зміг би функціонувати без жодних повноважень у таких питаннях, інакше усі могли б користуватися будь-якою IP-адресою, що спричинило б тотальний хаос в Інтернеті. Окрім керування призначенням IP-адрес, IANA також відповідає за ASN або за призначення номера автономній системі. ASN − це номер, що присвоюють окремій автономній системі. Як і IP-адреси, ASN − це 32-бітові числа. Але на відміну від IP-адрес, їх зазвичай називають числами десяткової системи, не розбиваючи на читабельні біти. На це є дві причини. Перша − IP-адреси мають показувати частину ідентифікатора мережі й частину ідентифікатора хосту для кожного номера. Цього легше досягти, поділивши номер на 4 частини по 8 бітів. Особливо в ті часи, коли всім керували класи адрес. ASN не має потреби змінюватись, щоб показати більше мереж або хостів. Просто потрібно, щоб головні таблиці маршрутизації були оновлені, щоб знати, що відображає ASN. Друга причина − ми набагато рідше дивимось на ASN, ніж на IP. Оскільки вам може статі в нагоді вміння подивитися на IP 9.100.100.100 і знати, що адресний простір 9.0.0.0/8 належить IBM, ASNs показують цілі автономні системи, і у випадку з ними досить побачити, що AS19604 належить IBM. Якщо ви не будете працювати в компанії, що надає інтернет-послуги, вам, як і більшості людей в ІТ, не обовʼязково знати докладну інформацію про роботу протоколів зовнішніх шлюзів. Але опанування основ автономних систем, ASN і того, як основні маршрутизатори передають між собою трафік важливі для розуміння деяких основних блоків, з яких побудований Інтернет.

Багато **мережевих протоколів** впроваджуються на основі специфікацій, опублікованих [організацією Internet Engineering Task Force (IETF)](https://www.ietf.org/). Ми поговоримо про це детальніше на майбутньому уроці!

Найпоширенішими **протоколами векторів відстані** є [RIP або протокол маршрутної інформації](https://en.wikipedia.org/wiki/Routing_Information_Protocol) ([IETF RFC2453](https://tools.ietf.org/html/rfc2453)) та [EIGRP або дистанційно-векторний протокол маршрутизації](https://en.wikipedia.org/wiki/Enhanced_Interior_Gateway_Routing_Protocol) ([документація Cisco](https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html)). Найпоширенішим протоколом стану каналу зв’язку є [OSPF (Open Shortest Path First)](https://en.wikipedia.org/wiki/Open_Shortest_Path_First) ([IETF RFC2328](https://tools.ietf.org/html/rfc2328)).

Із зовнішніх протоколів шлюзу на сьогодні використовується лише один. Інтернет мав уніфікувати спосіб обміну такою інформацією, тому з’явився єдиний стандарт. Цей стандарт називається [BGP, або протокол прикордонного шлюзу](https://en.wikipedia.org/wiki/Border_Gateway_Protocol) ([IETF RFC4271](https://tools.ietf.org/html/rfc4271)).

# Адресний простір без маршрутизації

А тепер – короткий урок з історії. Ще в 1996 році було очевидно, що Інтернет зростає нестримними темпами. Коли вперше з’явилось визначення IP, то IP-адреса визначалась як єдине 32-бітне число. Єдине 32-бітне число може представляти 4 294 967 295 унікальних чисел, що, безумовно, звучить як дуже багато. Однак станом на 2017 рік населення Землі, за оцінками, становить 7,5 мільярда людей. Це означає, що стандарт IPv4 навіть не має достатньо IP-адрес для кожної людини на планеті. Також слід враховувати цілі центри обробки даних, заповнені тисячами й тисячами комп’ютерів, необхідних для роботи великих технологічних компаній. Тому в 1996 році було опубліковано RFC 1918. RFC з англійської розшифровується як "Запит на коментарі". Для людей, які відповідають за роботу Інтернету, було дуже важливо узгодити стандартні вимоги для цього. У RFC 1918 було окреслено низку мереж, які можна визначити як немаршрутизований адресний простір. Суть немаршрутизованого адресного простору точно відповідає його назві. Це діапазони IP-адрес, виділених для використання тими, до кого неможлива маршрутизація. Не кожен комп’ютер, підключений до Інтернету, повинен мати можливість обмінюватися даними з іншим комп’ютером, підключеним до Інтернету. Немаршрутизований адресний простір дозволяє вузлам у такій мережі обмінюватися даними між собою, але жоден маршрутизатор шлюзу не намагатиметься надсилати трафік у таку мережу. Це може сприйматися як надзвичайне обмеження, і в деякому сенсі це так і є. У наступному модулі ми розглянемо технологію, відому як NAT, або перетворення мережевих адрес. Вона дає змогу комп’ютерам у немаршрутизованому адресному просторі комунікувати з іншими пристроями в Інтернеті. Але наразі давайте просто розглянемо немаршрутизований адресний простір у вакуумі. Стандарт RFC 1918 визначив три діапазони IP-адрес, на які магістральні маршрутизатори не спрямовуватимуть трафік. Це означає, що вони нікому не належать і ними ніхто не може користуватися. Фактично, оскільки вони відокремлені від руху трафіку в Інтернеті, кількість людей, які можуть використовувати ці адреси для своїх внутрішніх мереж, не обмежена. Три основні діапазони немаршрутизуваного адресного простору – 10.0.0.0/8, 172.16.0.0/12 і 192.168.0.0/16. Ці діапазони безкоштовні для всіх, хто використовує їх для внутрішніх мереж. Слід зазначити, що протоколи внутрішніх шлюзів спрямовуватимуть трафік у ці адресні простори. Таким чином, вони підходять для використання в межах автономної системи, але не з протоколами зовнішніх шлюзів. Ми багато чого дізналися в цьому модулі. Вітаю вас із цим успіхом! Далі ми розглянемо транспортний рівень і рівень додатків. Але спочатку – ще один тест.  У вас усе вийде!

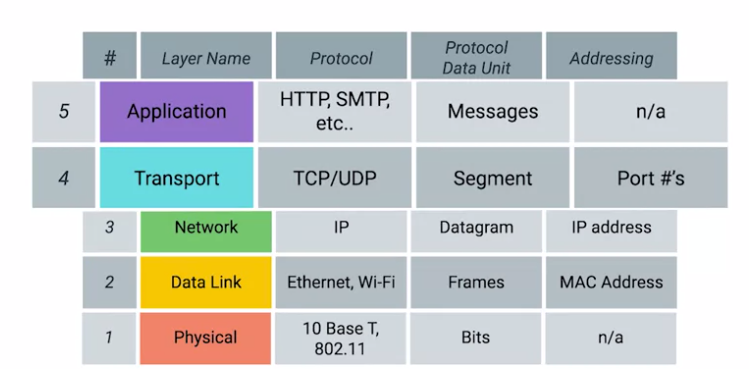
Пам’ятайте: ви завжди можете повернутися й переглянути матеріал, якщо потрібно.

Ви можете переглянути величезну колекцію документів RFC [тут](https://www.ietf.org/standards/rfcs/).

До речі, RFC [має довгу історію першоквітневих жартів](https://en.wikipedia.org/wiki/April_Fools%27_Day_Request_for_Comments). Моїми особистими улюбленцями є [RFC 1149](https://tools.ietf.org/html/rfc1149) та [RFC 3514](https://www.ietf.org/rfc/rfc3514.txt). Сенс жартів стане зрозумілий вам після того, як ми пройдемо наступний модуль.

# Транспортний і прикладний рівень. Вступ

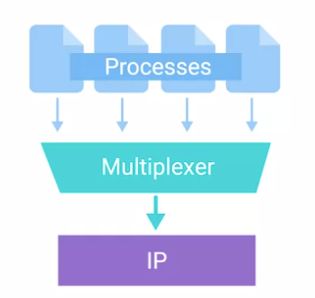
Перші три рівні нашої мережевої моделі допомогли нам описати, як окремі вузли в мережі можуть спілкуватися з іншими вузлами у власній мережі, або ж у віддаленій. Але ми не розглянули, як окремі комп’ютерні програми спілкуються між собою. Прийшов час дізнатися про це, тому що це — справжня мета комп’ютерних мереж. Ми об’єднуємо комп’ютери в мережу не лише, щоб вони обмінювалися даними, але й для того, щоб програми на цих комп’ютерах могли обмінюватися даними. Це робота транспортного та прикладного рівнів нашої мережевої моделі.



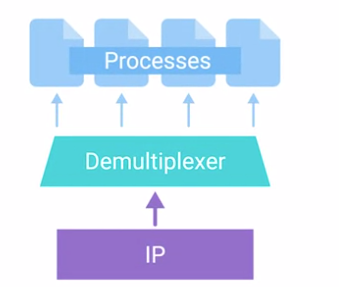
Якщо коротко, транспортний рівень дозволяє трафіку спрямовуватись на конкретні мережеві додатки. А прикладний рівень дозволяє цим додаткам спілкуватися їхньою мовою. Пройшовши цей модуль, ви навчитеся описувати порти TCP і сокети, та визначати різні компоненти заголовка TCP. Ви також знатимете різницю між протоколами, орієнтованими на з’єднання, і протоколами без з’єднання. Ви зможете пояснити, як використовують TCP для забезпечення цілісності даних. Готові перейти до наступного уроку? Сподіваюся, що так, тому що далі в нас транспортний рівень. Побачимось!

# Транспортний рівень

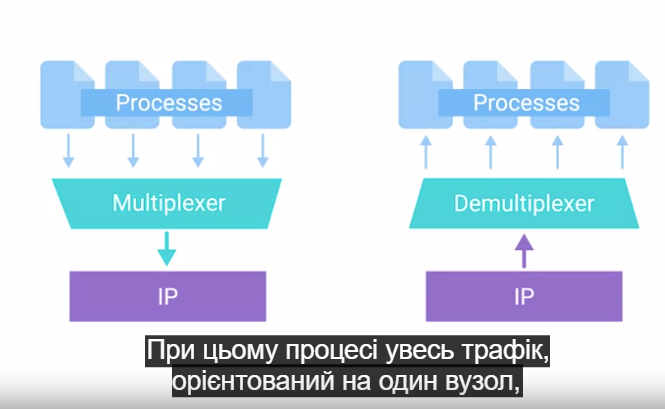
Транспортний рівень відповідає за різні функції, необхідні надійної роботі в комп’ютерній мережі. До них належать мультиплексування й демультиплексування трафіку, встановлення довготривалих з’єднань і забезпечення цілісності даних шляхом відстеження помилок і верифікації даних. На цьому уроці ви дізнаєтеся, що таке мультиплексування й демультиплексування, та як вони працюють. Ви зможете визначити відмінності між TCP та UDP, пояснити тристороннє представлення, розумітимете, як позначки TCP використовуються в цьому процесі. Крім того ви знатимете, як брандмауери забезпечують роботу мереж. Транспортний рівень має здатність до мультиплексування й демультиплексування, що відокремлює цей шар від усіх інших. Мультиплексування в транспортному рівні означає, що вузли в мережі можуть спрямовувати трафік до багатьох різних служб-отримувачів.

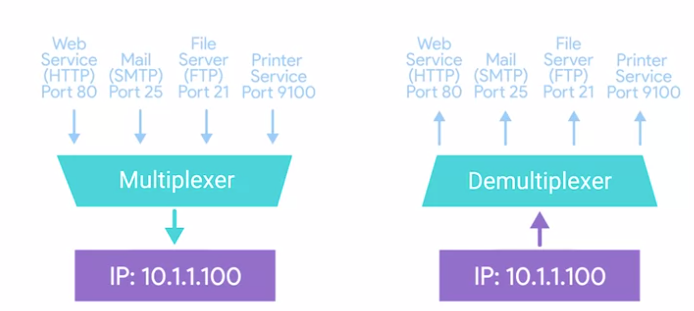


Демультиплексування – це те ж поняття, тільки для сторони отримувача.



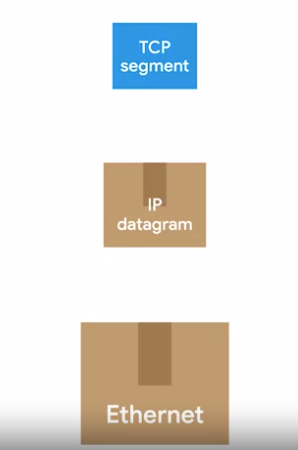
При цьому процесі увесь трафік, орієнтований на один вузол, доставляється до належної служби-отримувача.



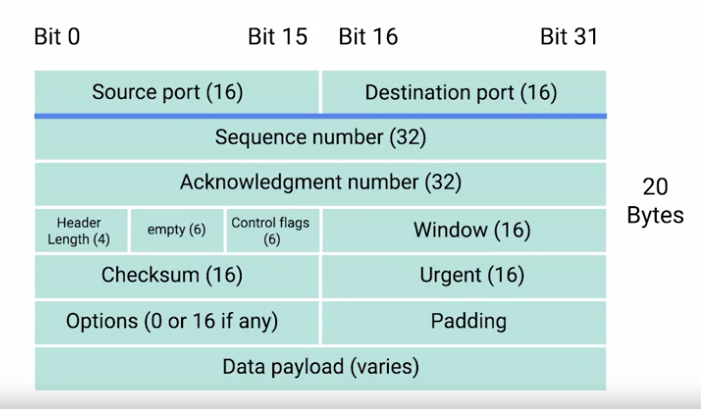
Мультиплексування й демультиплексування відбувається через порти. Порт – це 16-бітне число, що використовується для спрямування трафіку на певні служби, запущені на мережевому комп’ютері. Пам’ятаєте концепцію сервера й клієнтів? Сервер або служба – це програма, що працює на комп’ютері, який чекає запиту даних. Клієнт – це інша програма, яка запитує ці дані. Різні мережеві служби одночасно прослухують певні порти, шукаючи вхідні запити. Наприклад, традиційним портом для HTTP або незашифрованого вебтрафіку є 80. Якщо слід надіслати запит на сторінку з сервера на прослуховуючому комп’ютері на IP 10.1.1.100, трафік буде спрямовано в порт 80 на цьому пристрої. Порти зазвичай позначаються двокрапкою після IP-адреси. Тож повну IP-адресу й порт у цьому випадку можна написати як 10.1.1.100:80. Написаний таким чином номер називається адресою або номером сокета. Можливо, на тому ж пристрої працює також сервер FTP або протоколу передавання файлів. FTP – це старіший спосіб перенесення файлів з одного комп’ютера на інший, але він і досі іноді використовуєтьсяFTP традиційно веде прослуховування на порту 21, тому якщо треба з’єднатися з FTP-сервером, що працює на тій же адресі IP, що й вебсервер з нашого прикладу, трафік ведеться на 10.1.1.100, порт 21. Уявіть, що ви працюєте в службі ІТ-підтримки в невеликій компанії. У таких ситуаціях один сервер може прийняти майже всі програми, необхідні для ведення бізнесу. На одному комп’ютері може розміщуватися внутрішній вебсайт, поштовий сервер компанії, файловий сервер для обміну файлами, сервер друку для спільного доступу до мережевих принтерів тощо. Це все є можливим завдяки мультиплексуванню й демультиплексуванню та додаванню портів до схеми адресації.

# Склад сегмента TCP

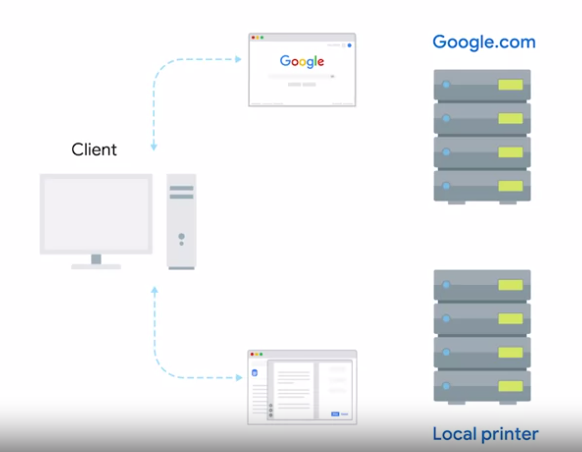
Увага! В цьому відео ми розглянемо сегмент TCP зсередини. В ІТ-підтримці, якщо мережевий трафік працює не так, як користувачі очікують, вам доведеться ретельно проаналізувати його, щоб усунути несправності. Отже, приготуйтеся зазирнути в усі внутрішні роботи. Так само, як кадр Ethernet інкапсулює IP-дейтаграму, IP-дейтаграма інкапсулює сегмент TCP.



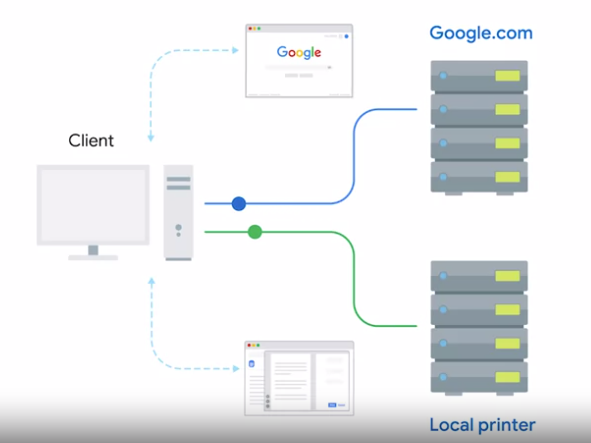
Пам’ятайте, що кадр Ethernet має секцію корисного навантаження, яка є лише вмістом IP-дейтаграми. Також пам’ятайте, що IP-дейтаграма має розділ корисного навантаження, що складається із сегмента TCP. Сегмент TCP складається із заголовка TCP та розділу даних. Як ви могли вже здогадатися, це область корисного навантаження, де прикладний рівень розміщує свої дані. Сам заголовок TCP має безліч полів, що містять безліч інформації.



По-перше, ми маємо поля порту джерела та порту призначення. Порт призначення – це порт послуги, для якої призначений трафік, про це ми говорили в минулому відео. Порт джерела – це порт з високою нумерацією, обраний із спеціального розділу портів, відомих як ефемерні порти. Пізніше ми розглянемо ці порти детально. Наразі достатньо знати, що порт джерела необхідний для збереження великої кількості вихідних з’єднань окремо. Ви знаєте, як порт призначення, скажімо порт 80, потрібен, щоб переконатися, що трафік досягає вебсервера, що працює на певній IP-адресі?



Аналогічно, порт джерела необхідний, щоб комп’ютер, який робить запит, відповідав серверу й надсилав ці дані до програми, яка насправді їх запитувала.



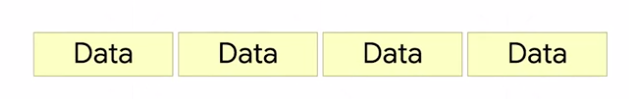
Саме так, коли вебсервер відповідає на ваші запити про перегляд вебсторінки, цю відповідь отримує браузер, а не текстовий процесор. Далі йде поле з номером послідовності.



Це 32-бітне число, яке використовується для відстеження, де в послідовності TCP сегментів буде знаходитися наш сегмент. Пам’ятаєте, що нижче в нашому стеку протоколів існують межі загального розміру того, що ми можемо надіслати через кабель.

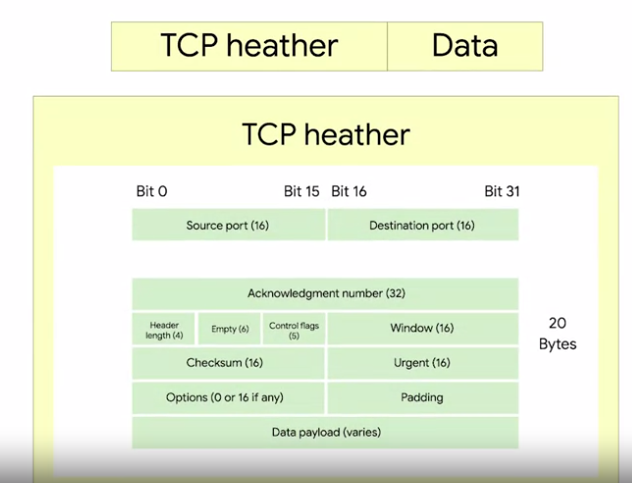
Кадр Ethernet зазвичай обмежений розміром 1518 байтів, але часто нам потрібно надсилати набагато більше даних. На транспортному рівні TCP ділить усі ці дані на багато сегментів.



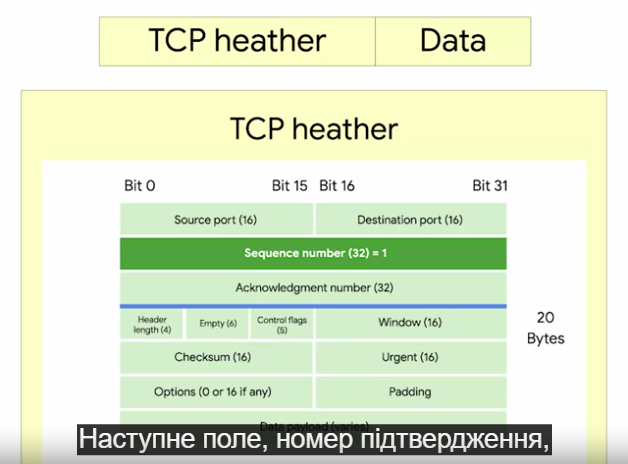




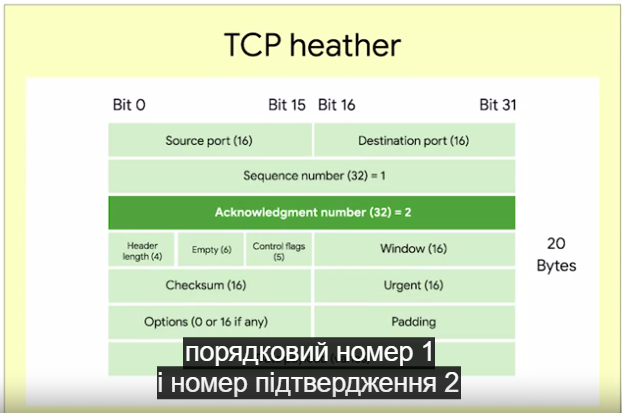




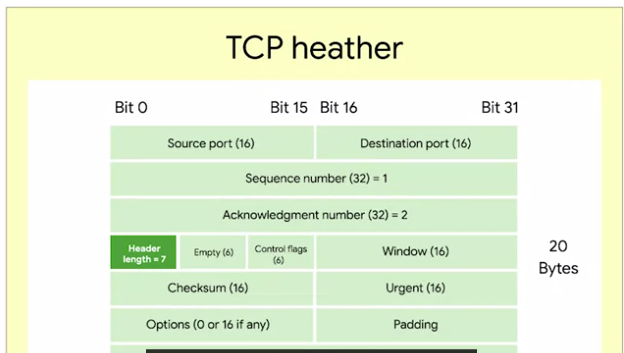
Номер послідовності в заголовку використовується для відстеження, яким із багатьох сегментів може бути цей конкретний сегмент. Наступне поле, номер підтвердження, дуже схоже на номер послідовності.



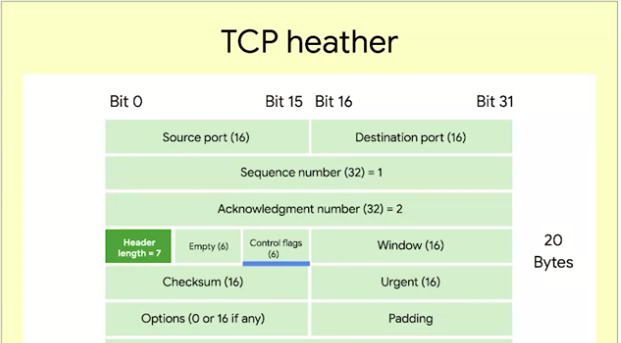
Номер підтвердження – це номер наступного очікуваного сегмента. Простою мовою, порядковий номер 1 і номер підтвердження 2 можна прочитати так: наш сегмент — це сегмент 1, очікуємо наступним сегмент 2.



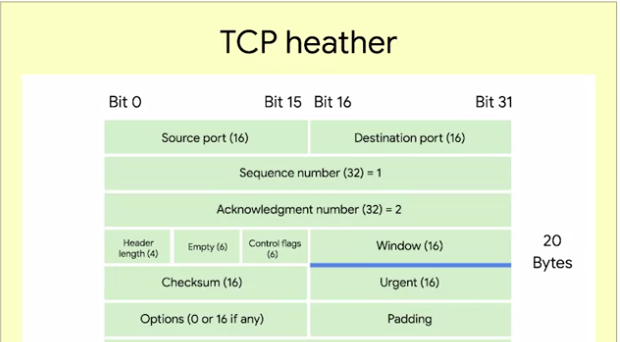
Далі йде поле зміщення даних. Це – чотирибітове число, яке показує, наскільки довгим є заголовок TCP для цього сегмента.



Завдяки цьому мережевий пристрій призначення розуміє, де починається фактичне корисне навантаження даних. Далі маємо шість бітів, зарезервованих для шести прапорців керування TCP.



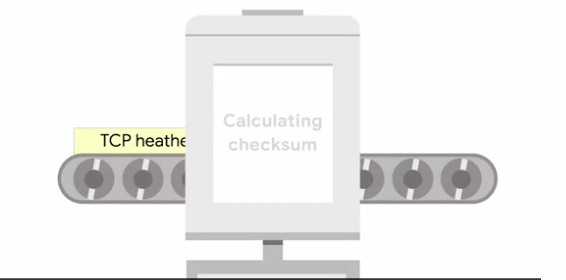
У наступному відео ми розглянемо прапорці керування та що вони означають, залишимо це на потім. Наступне поле – це 16-бітне число, відоме як вікно TCP.

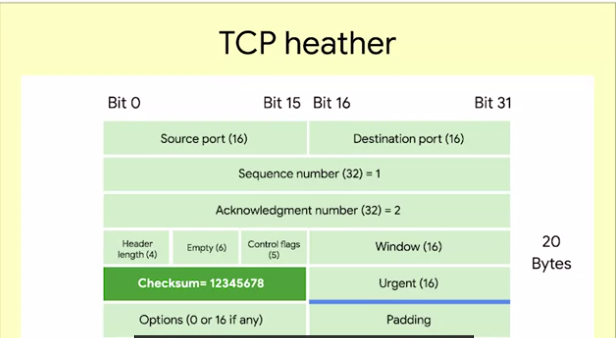


Вікно TCP визначає діапазон порядкових номерів, які можна надіслати перед потребою в підтвердженні. Як ми дізнаємося незабаром, TCP – це протокол, який дуже залежить від підтверджень. Це робиться для того, щоб переконатися, що всі очікувані дані насправді отримано, і що надсилаючий пристрій не витрачає час на надсилання даних, які не надходять. Наступне поле – 16-бітна контрольна сума.



Воно працює так, як і поле контрольної суми на рівні IP й Ethernet. Після того, як одержувач отримав увесь сегмент, контрольна сума обчислюється по сегменту й порівнюється з контрольною сумою в заголовку, щоб впевнитися, що дані не було втрачено або пошкоджено.

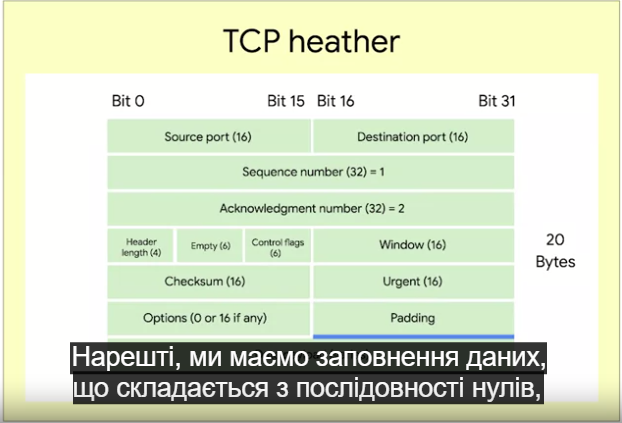




Поле показника важливості використовується разом з одним із прапорців керування TCP, щоб вказати ті сегменти, які можуть бути важливішими за інші. Цю особливість TCP ви ніколи не знайдете в сучасних мережах. Незважаючи на це, ви маєте розуміти, що означають усі розділи заголовка TCP. Далі маємо поле параметрів.

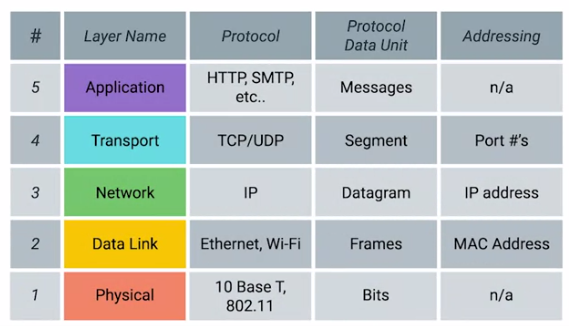


Як і поле показника важливості, воно рідко використовується, але іноді таки використовується для складніших протоколів керування потоком. Нарешті, ми маємо заповнення даних, що складається з послідовності нулів, потрібних, щоб розділ навантаження даних почався в певному місці.

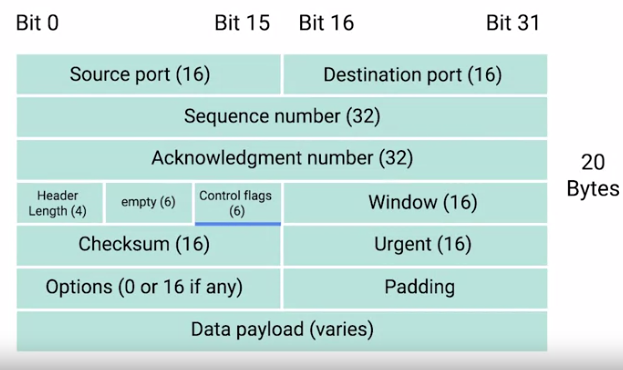


# Контрольні прапорці TCP і тристороннє рукостискання

Як протокол, TCP встановлює з’єднання, що використовують для надсилання довгих ланцюгів сегментів даних. Ви можете протиставити це з протоколами, що є нижче в мережевій моделі.



До них належать IP та Ethernet, які надсилають окремі пакети даних. Як фахівець з ІТ-підтримки, ви повинні розуміти, як усе працює, щоб ви могли усувати проблеми, де мережевий трафік поводиться не так, як очікується. TCP встановлює з’єднання через використання прапорців керування TCP, які використовуються в певному порядку. Перш ніж ми розглянемо встановлення та закінчення з’єднання, давайте визначимо шість прапорців керування TCP.



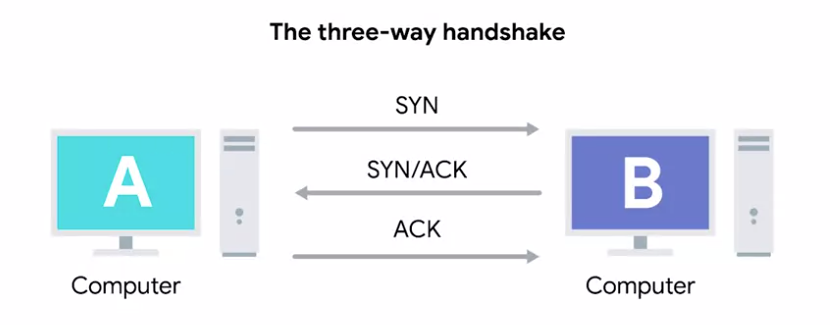
Ми розглянемо їх у тому порядку, в якому вони йдуть у заголовку TCP. Вони не обов’язково стоять у порядку частоти використання чи важливості. Перший прапорець відомий як **URG** – це скорочення від Urgent (терміново). Якщо він має значення 1, то це означає, що сегмент вважається терміновим і поле показника терміновості має більше даних. Як ми згадували в останньому відео, ця функція TCP використовується рідко й ми не часто можемо її побачити. Другий прапорець – **ACK**, скорочено від Acknowledge (підтвердити). Значення 1 в цьому полі означає, що поле з номером підтвердження потрібно перевірити. Третій прапорець – **PSH**, це скорочення від Push (просування). Це означає, що пристрій надсилання хоче, щоб пристрій призначення просунув дані, які знаходяться в буфері, до додатку на кінці призначення якнайшвидше. Буфер – це метод обчислення, де певний обсяг даних зберігається перед надсиланням в інше місце. Це має безліч практичних застосувань. З точки зору TCP, він використовується для ефективнішого надсилання великих частин даних. Зберігаючи певний обсяг даних у буфері, TCP може надавати значущі частини даних програмі, що чекає на них. Але в деяких випадках ви можете надіслати дуже малу кількість інформації, на яку програма, що слухає, має відповісти негайно. Саме це робить прапорець просування. Четвертий прапорець – **RST**, скорочено від Reset (скинути). Це означає, що одна зі сторін у TCP-з’єднанні не змогла відновитися після серії відсутніх або неправильно сформованих сегментів. Це спосіб для одного з членів TCP-з’єднання сказати: "Зачекай, я не розумію, про що йдеться, давай почнемо спочатку". П’ятий прапорець – **SYN**, що означає Synchronize (синхронізувати). Використовується при встановленні нового TCP-з’єднання, щоб сторона призначення знала, як читати поле порядкового номера. І, нарешті, наш шостий прапорець – **FIN**, що є скороченням від Finish (завершити). Коли цей прапорець має значення 1, це означає, що комп’ютер, який надсилає дані, більше не має що надсилати, і з’єднання можна закрити. Для хорошого прикладу, як використовуються прапорці контролю TCP, погляньмо, як встановлюється TCP-з’єднання. Комп’ютер А буде комп’ютером-відправником, а Комп’ютер В – комп’ютером-отримувачем.



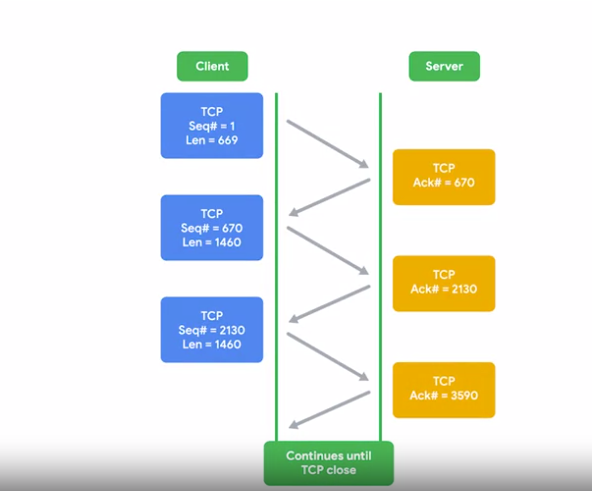
Щоб почати процес, Комп’ютер А надсилає сегмент TCP Комп’ютеру B зі встановленим прапорцем SYN. Це спосіб Комп’ютера А сказати: "Давай встановимо зв’язок і подивимося на моє поле порядкових номерів, щоб знати, де почнеться ця розмова". Далі Комп’ютер B відповідає сегментом TCP, в якому є прапорці SYN та ACK.



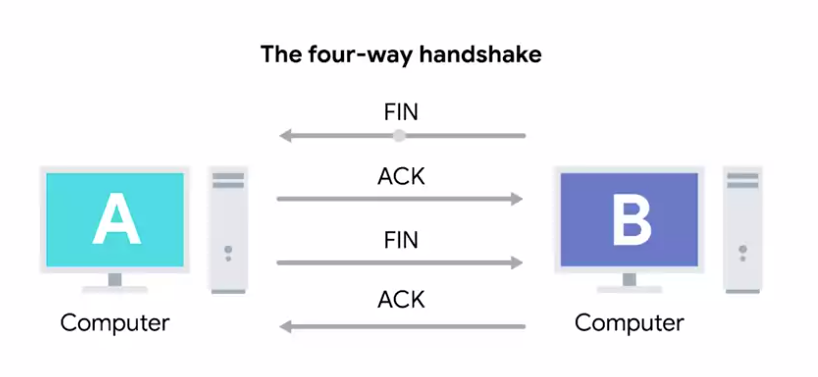
Так Комп’ютер B каже: "Звичайно, давай встановимо зв’язок, і я підтверджую твій порядковий номер". Потім Комп’ютер А ще раз відповідає прапорцем ACK і каже: "Я підтверджую твоє підтвердження.



Починаймо надсилати дані". Люблю їхню ввічливість. Цей обмін за участю сегментів SYN, SYN/ACK і ACK відбувається щоразу, коли встановлюється TCP-з’єднання, і воно настільки відоме, що має власну назву — тристороннє рукостискання. Рукостискання – це спосіб двох пристроїв упевнитися, що вони говорять одним протоколом і розуміють один одного. Після тристороннього рукостискання встановлюється TCP-з’єднання. Тепер Комп’ютер А може надсилати будь-які дані Комп’ютеру В й навпаки. Оскільки обидві сторони тепер надіслали одна одній свої SYN/ACK пари, TCP-з’єднання працює в повному дуплексі. Кожен сегмент, надісланий в будь-якому напрямку, має відповідати сегменту TCP з набором полів ACK. Так інша сторона завжди знає, що було отримано.



Як тільки один з пристроїв TCP-з’єднання готовий зупинити зв’язок, відбувається дещо, відоме як чотиристороннє рукостискання.



Комп’ютер, який готовий зупинити з’єднання, надсилає прапорець FIN, на який інший комп’ютер реагує прапорцем ACK. Якщо цей комп’ютер також готовий зупинити з’єднання, що майже завжди й відбувається, він надішле прапорець FIN. На це він в свою чергу отримає відповідь з прапорцем ACK. Теоретично, TCP-з’єднання може залишатися відкритим у симплексному режимі, навіть якщо одна сторона зупинила з’єднання. Але ви зустрічатимете таке дуже рідко. Наступного разу я розповім вам про стани сокетів TCP. Там буде багато мережевих каламбурів. Тож, до зустрічі!