Тема. Аналіз просування даних по стеку TCP/IP
Транспортний і мережевий рівні

При просуванні повідомлення з використанням протоколів стеку TCP/IP від прикладного рівня до фізичного рівня воно фрагментується (розбивається на сегменти) залежно від обмежень у каналі передачі. В процесі обробки відповідними протоколами до кожного блоку даних на кожному рівні додається службова інформація у вигляді заголовків певного формату.

На прикладному рівні реалізовані сервіси, які широко використовуються на практиці (рис. 2.1). До них належать: протокол передачі файлів між віддаленими системами FTP (File Transfer Protocol), протокол емуляції віддаленого терміналу telnet, поштові протоколи, протокол визначення імен DNS (Domain Name System) тощо. Кожна прикладна програма вибирає тип транспортування: або безперервний потік байтів, або послідовність окремих повідомлень. Прикладна програма передає дані транспортному рівню через порти, які можуть бути як стандартизованими, так і динамічними, значення яких змінюється від сеансу до сеансу.

Транспортний рівень підтримує два базових типи комунікації: сервіс, орієнтований на встановлення з'єднання, та сервіс без встановлення з'єднання (дейтаграмний). Сервіс комунікації з встановленням з'єднання передбачає процедуру встановлення, підтримки та розриву з'єднання між об'єктами верхніх рівнів. Цей тип комунікації використовують багато протоколів верхніх рівнів і називають надійним. Перевагами цього типу сервісу є: можливість виправлення помилок, управління потоками даних, контроль послідовності передачі блоків даних. Однак у деяких випадках більш ефективним є сервіс передачі даних без встановлення з'єднання. Це, в першу чергу, стосується програм, які потребують мінімалььних сукупних витрат, пов'язаних з передачею даних. Прикладами таких користувачів транспортного сервісу є системи збору даних, системи розповсюдження інформації, системи, що функціонують у діалоговому режимі «запит-відповідь» тощо.

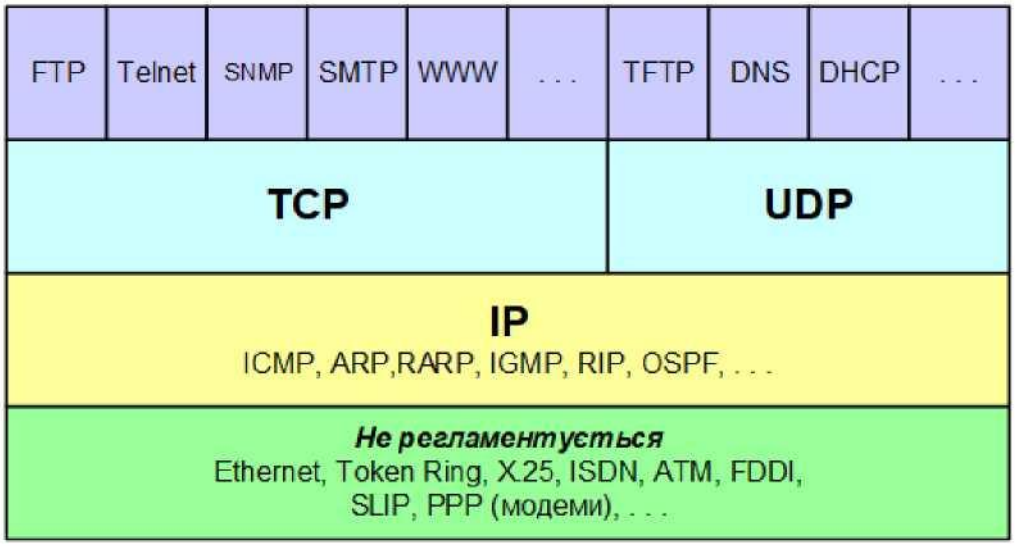
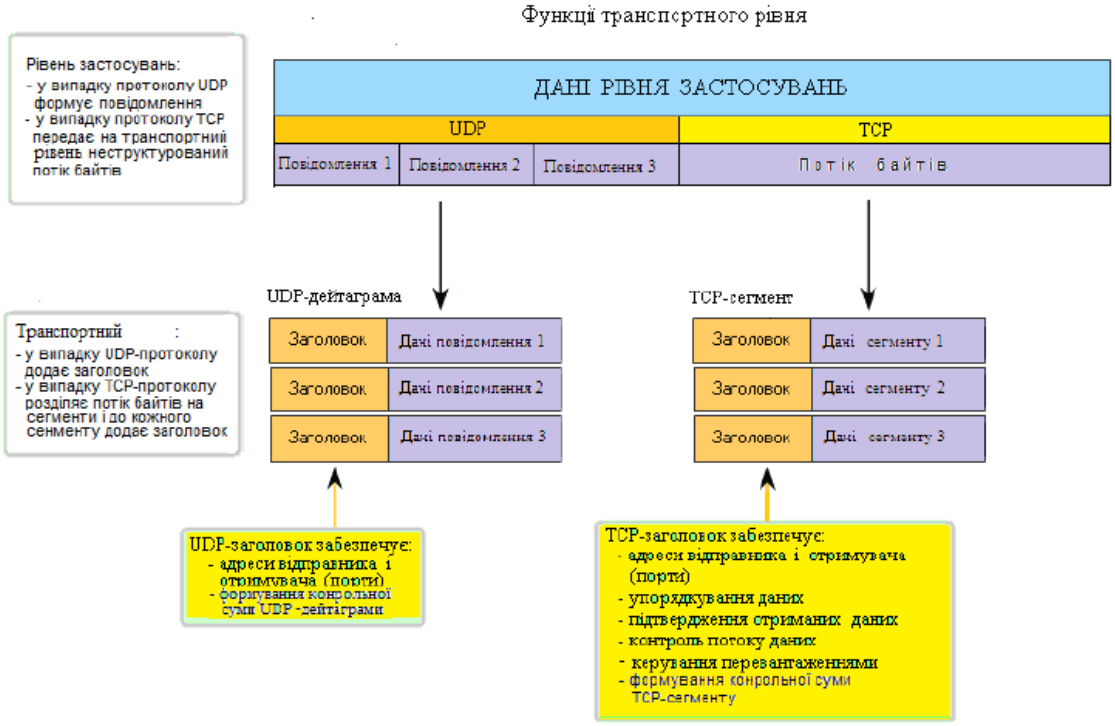


Рисунок 2.1- Стек протоколів TCP/IP

Передача повідомлення через мережу передбачає поділ його на блоки. Розміри цих блоків визначаються максимальним розміром блоку даних канального рівня MTU (Maximum Transfer Unit) або налаштуваннями, встановленими мережевим адміністратором. Ця процедура називається сегментацією (рис. 1.4). Саме тому блоки даних транспортного рівня називають сегментами. Зазвичай термін «сегмент» застосовують до одиниць передачі даних у межах надійного (гарантованого) транспортного сервісу, наприклад, TCP- сегменти. Блоки даних негарантованого транспортного сервісу називають UDP-дейтаграмами або просто дейтаграмами (рис. 1.6). На стороні одержувача виникає зворотна задача: формування повідомлення з отриманих сегментів.

Аналізуючи заголовок пакету, який отримано від мережевого рівня, транспортний модуль визначає за номером порту отримувача, якому із процесів- застосувань направлені дані, і пропонує ці дані відповідному процесу (можливо, після перевірки їх на наявність помилок і т.п.). Номери портів отримувача і відправника записуються в заголовок транспортним модулем, що відправляє дані; заголовок транспортного рівня містить також і іншу службову інформацію; формат заголовку залежить від транспортного протоколу, що використовується (UDP чи TCP).

Основні функції протоколів UDP та TCP, які найчастіше використовуються на практиці, представлено на рисунку 2.2. Опрацювання даних прикладного рівня на транспортному рівні відбувається по-різному і залежить від протоколу, який використовується на транспортному рівні: TCP чи UDP.



У випадку використання протоколу TCP дані прикладного рівня (потік байтів) можуть бути розділені на блоки, які після додавання до кожного з них TCP-заголовків називаються сегментами. Поділ потоку байтів на сегменти відбувається без врахування їхньої логічної структури, тобто межі між записами не зберігаються. Поділ або сегментація даних рівня застосувань гарантує по- перше можливість їх передачі в рамках технічних обмежень середовища передачі, а по-друге те, що дані із різних застосувань можуть бути мультиплексовані в каналі передачі. В TCP кожен заголовок сегменту містить порядковий номер. Цей порядковий номер дозволяє протоколам транспортного рівня на кінцевому хост-вузлі збирати сегменти у тому порядку, в якому вони були відправлені.

У випадку використання протоколу UDP опрацювання даних прикладного рівня обмежується приєднанням до повідомлення, яке надійшло із прикладного рівня, UDP-заголовку. Утворений таким чином блок даних називається дейтаграмою.

Хоча служби, які використовують UDP, також відслідковують взаємодію застосувань, вони не слідкують за порядком, в якому інформація була передана. Застосування, яке використовує UDP, повинне враховувати той факт, що дані, можливо, не надійдуть у тому порядку, в якому були відправлені.

1.1. Протокол UDP. Заголовок UDP-сегменту

Протокол передачі дейтаграм користувача UDP (User Datagram Protocol) фактично не виконує ніяких особливих функцій додатково до

функцій мережевого рівня (протоколу IP). Протокол UDP використовується або при пересиланні коротких повідомлень, коли витрати на встановлення сеансу і перевірку коректної доставки даних виявляються вищими за витрати на повторну (при невдачі) пересилку повідомлення, або тоді, коли сама організація процесу-додатку забезпечує встановлення з'єднання і перевірку доставки пакетів (наприклад, NFS).

До даних користувача, що надійшли від прикладного рівня, додається UDP-заголовок (рис. 2.3), і сформована таким чином UDP-дейтаграма відправляється на мережевий рівень.

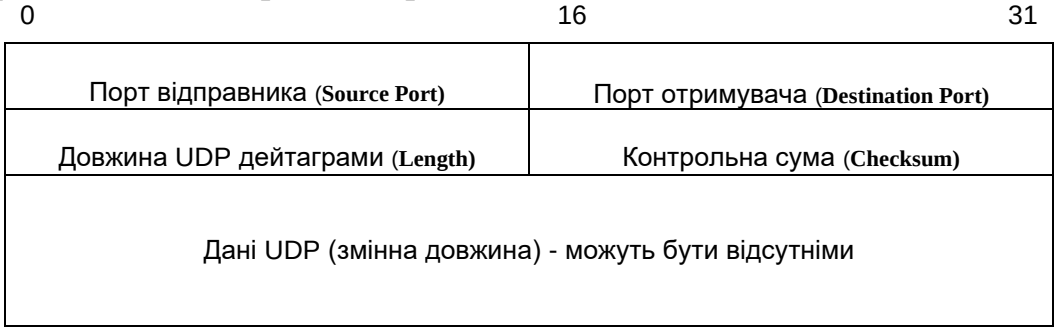


Рисунок 2.3 - Формат UDP-дейтаграми

- Source Port - номер порту процесу-відправника;
- Destination Port - номер порту процесу-отримувача;
- Length - довжина UDP-дейтаграми разом із заголовком, в октетах;
- Checksum - контрольна сума(за наявністю).. Поле Checksum містить контрольну суму пакету UDP, яка визначається для всього пакету UDP з доданим псевдозаголовком.

Після заголовку безпосередньо ідуть дані користувача, передані модулю UDP прикладним рівнем. Протокол UDP розглядає ці дані як одне повідомлення; він ніколи не розбиває повідомлення для передачі на кілька фрагментів і не об'єднує кілька повідомлень для пересилки в одному сегменті.

При отриманні пакету мережевого рівня модуль UDP перевіряє контрольну суму і передає вміст повідомлення прикладному процесу, номер порту якого вказаний у полі «Destination Port».

Якщо перевірка контрольної суми виявила помилку або якщо процесу, що підключений до потрібного порту, не існує, пакет ігнорується. Якщо пакети надходять швидше, ніж модуль UDP встигає їх опрацювати, то ці пакети також ігноруються. Протокол UDP є ненадійним протоколом без встановлення з'єднання, освільки:

- не має ніяких засобів підтвердження безпомилкового прийому даних або повідомлення про помилку;
- не забезпечує надходження повідомлень про порядок відправки;
- не виконує попереднього встановлення сеансу зв'язку між прикладними процесами.

Максимальна довжина UDP-дейтаграми дорівнює максимальній довжині IP-дейтаграми (65535 октетів) зменшеної на мінімальну довжину IP-заголовку (20) і UDP-заголовку (8), тобто 65507 октетів. На практиці зазвичай використовуються блоки даних довжиною 8192 октет.

Прикладами прикладних процесів, які використовують протокол UDP, є мережева файлова система NFS (Network File System), простий протокол передачі файлів TFTP (Trivial File Transfer Protocol), простий протокол керування мережею SNMP (Simple Network Management Protocol), доменна служба імен DNS (Domain Name Service).

1.2. Протокол TCP. Заголовок TCP-сегменту

Протокол TCP (Transmission Control Protocol) орієнтований на встановлення з'єднання між взаємодіючими прикладними процесами, який забезпечує надійну передачу та прийом даних. Метою протоколу є забезпечення надійного обміну даними між прикладними сервісами робочих станцій мережі. Для цього TCP забезпечує встановлення логічного з'єднання між прикладними процесами віддалених модулів мережі, контролює послідовність передачі сегментів повідомлення через комунікаційне середовище мережі, виявляє та обробляє помилки передачі.

При прийомі з мережі дейтаграми, в полі Protocol якої вказано код протоколу TCP (6), модуль IP передає дані цієї дейтаграми модулю TCP. Ці дані являють собою TCP-сегмент, що містить TCP-заголовок і дані користувача (дані прикладного сервісу). Модуль TCP аналізує службову інформацію заголовку, визначає, якому саме процесу потрібно передати дані користувача, перевіряє цілісність і порядок надходження даних і підтверджує їх прийом іншій стороні. В разі відсутності помилок і правильної послідовності даних користувача вони передаються прикладному процесу.

Максимальний розмір сегменту MSS (Maximum Segment Size), який TCP- рівень може надіслати на віддалений кінець з'єднання, вибирається таким чином, щоб при інкапсуляції сегменту в IP-пакет він поміщався туди цілком, тобто MSS не повинен бути більшим за максимальний розмір поля даних IP- дейтаграми, і розраховується за наступною формулою:

$$MSS = MTU - sizeof(TCPHDR) - sizeof(IPHDR),$$

де sizeof(TCPHDR) і sizeof(IPHDR) - відповідно розміри TCP та IP заголовків.

Розмір кожного з цих заголовків може бути від 20 до 60 байтів (при відсутності та наявності додаткових опцій відповідно). Тобто при відсутності додаткових параметрів передачі MSS = 1500 - 20 - 20 = 1460 байтів.

При встановленні з'єднання кожна сторона може оголосити своє значення MSS. І тоді вибирається найменше із двох значень.

Максимальний розмір сегменту TCP за замовчуванням - 536 байт. Якщо хост-вузол бажає встановити інше значення максимального розміру сегменту, то воно вказується як опція TCP в сегменті синхронізації TCP SYN під час встановлення з'єднання TCP. Це значення не може бути змінене після того, як з'єднання встановлене.

TCP-сегмент, структура якого наведена на рис. 2.4, містить заголовок і дані і завжди кратний 4 байтам. Розмір заголовку - не менше 20 байт і може бути збільшений за рахунок додаткових опцій (парламентів) передачі до 60 байт.

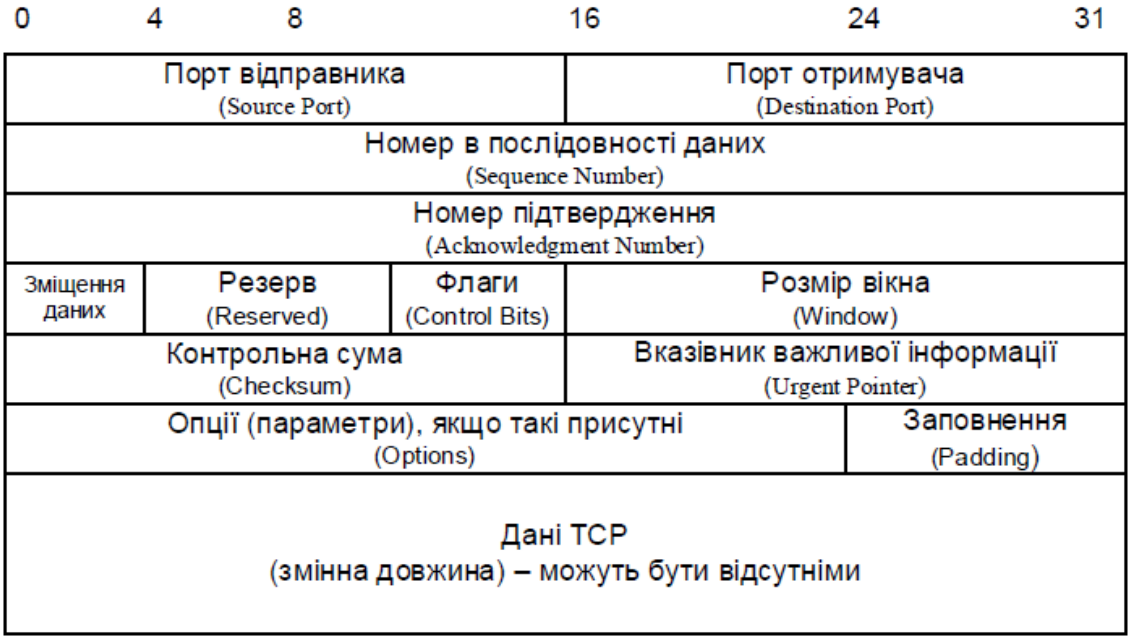


Рисунок 2.4 - Формат TCP-сегменту

- Source Port, Destination Port - номери портів процесу-відправника і процесу-отримувача відповідно;
- Sequence Number- порядковий номер першого октету в полі даних користувача. Значення, що присвоєне сегменту TCP, визначає номер стартового байту пакета, якщо тільки не встановлений флаг SYN; якщо в сегменті встановлено флаг SYN, то поле номера містить значення початкового порядкового номеру послідовності (ISN), а перший октет даних має номер ISN + 1;
- Acknowledgment Number (ACK) - значення, яке відправляється станції- відправнику, яке підтверджує прийом раніше відправленого сегменту(сегментів); задає також наступний порядковий номер байту, який станція очікує отримати; при встановленому з'єднанні

сегмент підтвердження відправляється завжди;

- Data Offset (зміщення даних) - визначає довжину заголовку TCP-сегмента (тобто кількість 32-бітних блоків у заголовку TCP) та вказує, де закінчується заголовок та починаються дані (визначає зміщення даних у сегменті);

- Reserved (6 біт) - зарезервовано; заповнюється нулями.
- Control Bits - біти для управління; активним є стан «біт встановлений».

  - URG - **Флаг терміновості**. Індикатор терміновості, що використовується при відправці повідомлення адресату, який очікує на екстрену інформацію. Таке повідомлення може бути передано станції, що отримує дані, якщо вона закрила вікно прийому для відправника. Однак станція отримувач все ще буде приймати сегменти, в яких цей флаг встановлено.
  - ACK - **Флаг підтвердження**. Якщо даний флаг встановлено, то пакет містить підтвердження для дейтаграми, яка була передана раніше.
  - PSH - **Флаг форсованої відправки**. Сегмент з таким встановленим флагом вимагає виконання операції *push*, тобто включена функція виштовхування і необхідна невідкладна відправка даних після зчитування сегмента (даних цього пакета).
  - RST - **Перевстановлення з'єднання**. Обрив з'єднання з метою відмови на запит з'єднання. Переривання зв'язку (перезавантаження даного з'єднання).
  - SYN - **Флаг синхронізації** номерів сегментів у черзі, використовується для ініціалізації та встановлення порядкового значення.
  - FIN - **Флаг закінчення**. Визначає, що в ініціатора з'єднання (відправника даних) більше немає даних для відправлення, тобто передача інформації завершена.

В документі RFC 3168 запропоновано використовувати два молодших біти зарезервованої області для управління перевантаженням у мережі (рис. 2.5). Проте останнім часом згідно з RFC 3168 пропонується використовувати два старших біти TCP-прапорців для зберігання бітів ECN (Explicit Congestion Notification).

		URG	ACK	PSH	RST	SYN	FIN

CWR

ECN

Рисунок 2.5 - Розташування бітів управління

Біт ECN (Explicit Congestion Notification), який називають exo-бітом, встановлюється, коли в прийнятому сегменті встановлено біт повідомлення про перевантаження в байті «Тип обслуговування» IP-заголовку. Це означає, що обидві сторони, які взаємодіють з використанням протоколу TCP, підтримують повідомлення про перевантаження, що виявляється в процедурі узгодження параметрів TCP-з'єднання. Exo-біт ECN у TCP-заголовку повідомляє відправника про необхідність знизити швидкість передачі даних через перевантаження в мережі між відправником і отримувачем. Після отримання TCP-сегменту з встановленим exo-бітом відправник у два рази зменшує вікно перевантаження - розмір буферу даних що відправляються. Після цього він встановлює біт CWR (Congestion Window Reduced), який свідчить про зменшення вікна перевантаження для повідомлення іншій стороні про виконані дії для зниження рівня перевантаження. Цей механізм дозволяє зменшити кількість відкинутих пакетів, але встановлення додаткових бітів може викликати повідомлення про тривогу від систем виявлення вторгнень. Тому на сьогодні більшість користувачів використовують ці біти тільки з метою сканування. Крім того, деякі пристрої фільтрації пакетів можуть не пропустити вхідні пакети з цими встановленими прапорцями. Тому доведеться виконати ще багато дій для поступового впровадження бітів ECN і для можливості відрізняти їх від спроб сканування.

- Window - величина вікна в октетах містить кількість байтів, які можуть бути відправлені після байту, отримання якого вже підтверджено. За допомогою значення величини вікна хост-отримувач повідомляє відправника про поточний розмір вхідного буфера для конкретного з'єднання. Значення величини вікна змінюється динамічно під час з'єднання. Воно зменшується на розмір прийнятих даних, але ще не опрацьованих хостом- отримувачем. Якщо вхідний буфер отримувача заповнюється повністю, то розмір вікна зменшується до 0, що повідомляє хосту-відправнику про необхідність тимчасово припинити передачу даних. Опрацювавши частину даних із вхідного буфера, хост-отримувач відправляє відправнику оновлену інформацію про розмір вікна, що вказує на можливість відновлення передачі даних. Очевидно, що хост-отримувач управляє потоком TCP-даних, в основному, за допомогою інформації про розмір вікна, тобто потоком даних у мережі керує переважно отримувач.
- Checksum - контрольна сума, яка крім заголовку сегменту і поля даних, враховує ще і псевдозаголовок, який записується між заголовком TCP (або UDP в разі його використання) та заголовком

протоколу IP (рис.2.6). Цей псевдозаголовок містить IP-адресу відправника (4 октети), IP-адресу отримувача (4 октети), нульовий октет, 8-бітне поле «Протокол», яке аналогічне полю в IP-заголовку, і 16 біт довжини TCP сегменту в октетах. Такий підхід забезпечує захист протоколу TCP від сегментів, які «помилилися» в маршруті. Інформація для псевдозаголовку передається через інтерфейс «Протокол TCP/мережевий рівень» як аргументи або як результат обробки запитів від протоколу TCP до протоколу IP.

Метою використання псевдозаголовку є перевірка того факту, що TCP- сегмент **(**UDP-дейтаграмма) переданий в коректне місце призначення, яке в загальному випадку характеризується сукупністю адреси конкретної робочої станції та порту додатка на ньому.

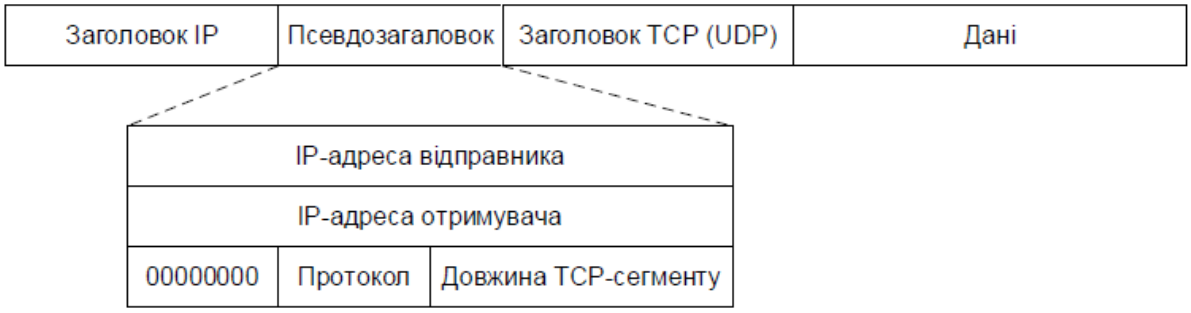


Рисунок 2.6 - Розташування псевдозаголовку та його структура

- Вказівник важливої інформації (Urgent Pointer) служить для зберігання довжини термінових даних, які розміщуються на початку поля даних сегменту. Вказує зміщення октету, який слідує за терміновими даними, відносно першого октету в сегменті. Наприклад, у сегменті передаються октети з 2001-го по 3000-й, при цьому перші 100 октетів є терміновими даними, тоді поле Urgent Pointer = 100. Протокол TCP не визначає, як саме мають опрацьовуватися термінові дані, але припускає, що прикладний процес буде намагатися швидко їх опрацювати. Поле Urgent Pointer обробляється тільки тоді, коли встановлений прапорець URG.
- Опції (Options) - поле змінної довжини; може бути відсутнім або містити одну опцію або список опцій, які реалізують додаткові послуги протоколу TCP. Опція складається із октету «Тип опції», за яким можуть знаходитися октет «Довжина опції в октетах» і октети з даними для опції.

Стандарт протоколу TCP визначає три опції (типи 0, 1, 2).

Опції типів 0 і 1 («Кінець списку опцій» і «Немає операції» відповідно) складаються із одного октету, який містить значення типу опції. При виявленні в списку опції «Кінець списку опцій» аналіз опцій

припиняється, навіть якщо довжина заголовку сегмента (Data Offset) ще не вичерпана. Опція «Немає операції» може використовуватися для вирівнювання між опціями по межі 32 біти.

Опція типу 2 «Максимальний розмір сегменту» складається із 4 октетів: одного октету типу опції (значення дорівнює 2), одного октету довжини (значення дорівнює 4) і двох октетів, які містять максимальний розмір сегменту MSS, який може отримати TCP-модуль, який відправив сегмент з даною опцією. Опцію можна використовувати тільки в SYN-сегментах на етапі встановлення з'єднання.

- Padding - вирівнювання заголовку по межі 32-бітного слова, якщо список опцій займає не ціле число 32-бітних слів, і заповнюється нулями.

1.3. Мережевий рівень і протокол IP

На мережевому рівні стеку протоколів TCP/IP функціонує протокол IP (будь-якої версії). Згідно документу RFC-791 протокол IP забезпечує передачу блоків даних, які називаються дейтаграмами, від відправника до отримувача, де відправниками і отримувачами є комп'ютери (робочі станції), які ідентифікуються адресами фіксованої довжини (IP-адресами). Протокол IP виконує, за необхідності, також фрагментацію і зборку дейтаграм для передачі даних через мережу з невеликою кількістю пакетів.

Протокол IP доставляє блоки даних, які називаються дейтаграмами, від хост-вузла з певною IP-адресою до іншого хост-вузла. IP-адреса - це унікальний 32-бітний (в разі використання протоколу IP версії 4) ідентифікатор комп'ютера або іншого модуля мережі (його мережевого інтерфейсу). Дані для дейтаграми передаються IP-модулю з транспортного рівня. IP-модуль доповнює ці дані заголовком, в який додається IP-адреса відправника і отримувача та інша службова інформація. Сформована таким чином дейтаграма передається на рівень доступу до середовища передачі (наприклад, одному із фізичних інтерфейсів) для відправки в канал передачі даних.

Коли модуль IP отримує дейтаграму з нижнього рівня, він перевіряє IP- адресу призначення. Якщо дейтаграма адресована даному комп'ютеру, то дані з неї передаються на обробку модулю розташованого вище рівня (якому саме - вказано в заголовку дейтаграми). Якщо ж адреса призначення дейтаграми - інший комп'ютер, то модуль IP може прийняти одне з двох рішень: або знищити дейтаграму, або відправити її до місця призначення, визначивши маршрут проходження (таким чином функціонують проміжні станції - маршрутизатори).

Також може виникнути необхідність на границі мереж з різними

характеристиками розділити дейтаграму на фрагменти, а потім зібрати в єдине ціле на комп'ютері-отримувачі.

Якщо модуль IP з будь-якої причини не може доставити дейтаграму, вона знищується. При цьому модуль IP відправляє комп'ютеру-відправнику цієї дейтаграми повідомлення про помилку. Такі повідомлення відправляються за допомогою протоколу ICMP (Internet Control Message Protocol), який є невід'ємною частиною модуля IP. Ніяких інших засобів контролю коректності даних, підтвердження їх доставки, забезпечення коректного порядку проходження дейтаграм, попереднього встановлення з'єднання між комп'ютерами протокол IP не має. Ці задачі виконує транспортний рівень.

Протокол IP є *ненадійним* протоколом *без встановлення з'єднання*, що в свою чергу означає, що протокол IP не підтверджує доставку даних, не контролює цілісність отриманих даних і не виконує операцію квотування (handshaking) - обмін службовими повідомленнями, які підтверджують встановлення з'єднання з вузлом призначення і його готовність до прийому даних. Протокол IP обробляє кожну дейтаграму як незалежну одиницю, яка не має зв'язку з іншими дейтаграмами в Інтернет. Після того, як дейтаграма відправляється в мережу, її подальша доля ніяк не контролюється відправником (на рівні протоколу IP). Якщо дейтаграма не может бути доставлена, вона знищується. Вузол, який знищує дейтаграму, може відправити за зворотною адресою ICMP-повідомлення про помилку. Гарантію правильності передачі даних надають протоколи вищерозташованого рівня (наприклад, протокол TCP), які мають для цього необхідні механізми.

IP-дейтаграма складається із заголовку і поля даних. Заголовок дейтаграми складається з 32-розрядних слів і має змінну довжину, яка залежить від розміру поля додаткових параметрів «Options», але завжди кратна 32 бітам. Довжина заголовка дейтаграми - від 20 до 60 байтів. За заголовком безпосередньо розташовані дані, які передаються в дейтаграму (рис. 2.7).

Рисунок 2.7 - Формат заголовку IP-дейтаграми

Значення полів заголовку наступні:

- Ver - версія протоколу IP, в лабораторній роботі використовується протокол версії 4 (хоча існують інші версії, які мають номери 5-8).
- IHL (Internet Header Length) - довжина заголовку в 32-бітних словах; діапазон допустимих значень від 5 (мінімальна довжина заголовку, поле «Options» відсутнє) до 15 (тобто, може мати максимум 40 байтів опцій).
- TOS (Type Of Service)- значення поля визначає пріоритет дейтаграми і необхідні параметри передачі (тип маршрутизації). Структура байту TOS представлена на рис. 2.8.

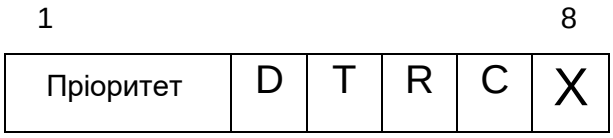


Рисунок 2.8 - Структура байту «Тип сервісу»

Пріоритети ((precedence) на практиці використовується рідко): **0 (000)** - звичайний;
1 (001) - пріоритетний;

2 (010) - негайний;
3 (011) - терміновий (миттєвий);
4 (100) - екстрений (більш, ніж миттєвий);
5 (101) - CRITIC/ECP (обробка критичних і екстерних викликів)**;**
6 (110) - міжмережне управління;
7 (111) - мережне управління.

Біти D, T, R, C визначають бажаний тип маршрутизації (вибір маршруту):
8 - Date - з мінімальною затримкою;
9 - Throughput - з максимальною пропускною спроможністю;
10 - Reliability - з максимально. надійністю;
C - Cost - з мінімально. вартістю.
За замовчуванням всі біти встановлені в 0 (звичайний сервіс).

В дейтаграмі може бути встановлений тільки один із бітів D, T, R, C. Якщо необхідно реалізувати максимальну безпеку передачі, то встановлюються всі ці чотири біти.

Реальне урахування пріоритетів і вибір маршруту у відповідності з значенням байту TOS залежить від маршрутизатора, його програмного забезпечення і налаштувань. Маршрутизатор може підтримувати розрахунок маршрутів для усіх типів TOS, для частини або ігнорувати TOS взагалі. Маршрутизатор може враховувати значення пріоритету при

обробці всіх дейтаграм або при обробці дейтаграм, які надходять тільки від деякої обмеженої множини вузлів мережі, або зовсім ігнорувати пріоритет.

З моменту появи у складі IP-заголовку байт «Тип обслуговування» TOS (Type of Service) зазнав кількох змін. Однією з них стало передбачене в документі RFC 2481 і більш новому RFC 3168 використання двох молодших бітів цього байту для зберігання явного повідомлення про перевантаження ECN (Explicit Congestion Notification). Це пов'язано з використанням деякими маршрутизаторами методу випадкового раннього виявлення RED (Random Early Detection) або активного управління чергами з вірогідністю втрати пакетів.

При високому навантаженні в мережі маршрутизатор може відкидати деякі пакети. Метод RED призначений для зменшення негативного ефекту втрати пакетів за допомогою обчислення вірогідності перевантаження черги до інтерфейсу маршрутизатора і маркування пакетів, які можуть бути відкинуті при виникненні такого перевантаження.

- Total Length - довжина всієї дейтаграми в октетах, разом із заголовком і даними, максимальне значення 65535, мінімальне - 21 (заголовок без опцій і один октет у полі даних).
- ID (Identification), Flags (3 біти), Fragment Offset (13 біт) використовуються для фрагментації та зборки дейтаграм.
  Ідентифікатор дейтаграми містить унікальний код дейтаграми, що визначає порядковий номер дейтаграми в послідовності. Це поле використовується приймаючим вузлом для об’єднання в єдине ціле фрагментів дейтаграми. Значення цього поля встановлюється відправником.

  Флаги

  Біт 0 - зарезервований.

  Біт 1 виконує управління фрагментацією:

  DF=0 - можна фрагментувати,

  DF=1 - фрагментація заборонена.

  Біт 2 показує, чи є даний фрагмент останнім.

  MF=0 - останній фрагмент,

  MF=1 - є наступний фрагмент.

  Зміщення фрагменту показує, де у вихідній дейтаграмі перебуває даний фрагмент. Величина зміщення задається в **64-бітних блоках.** Перший фрагмент має нульове зміщення. Поле загальної довжини вказує довжину вихідного пакету, а поле зміщення показує вузлу, що здійснює збирання, зміщення відносно його початку.

- TTL (Time To Live) - «час життя» дейтаграми. Встановлюється відправником, вимірюється в кількості комунікаційних вузлів, через які може передаватися дейтаграма, або в секундах. Кожен

маршрутизатор, через який проходить дейтаграма, переписує значення TTL, попередньо віднімаючи від нього час, витрачений на обробку дейтаграми. Оскільки швидкість обробки даних на маршрутизаторах велика, на одну дейтаграму витрачається зазвичай менше секунди, тому фактично кожен маршрутизатор зменшує TTL на одиницю. При досягненні значення TTL=0 дейтаграма знищується, при цьому відправнику зазвичай відправляється відповідне ICMP-повідомлення. Використання та контроль TTL запобігає зациклюванню дейтаграми при передачі в мережі.

- Protocol (8 біт) - визначає програму (розташований вище протокол стеку), якій мають бути передані дані дейтаграми для подальшої обробки. Коди деяких протоколів, які найчастіше використовуються на практиці, наведені в таблиці 2.1.
- Header Checksum - контрольна сума заголовку, вираховується як доповнення до суми всіх 16-бітових слів заголовку. Перед підрахунком контрольної суми значення поля «Header Checksum» обнуляється. Оскільки маршрутизатори змінюють значення деяких полів заголовку при обробці дейтаграми (як мінімум, поля TTL), контрольна сума кожним маршрутизатором перераховується заново. Якщо при перевірці контрольної суми виявляється помилка, дейтаграма знищується.
- Source Address - IP-адреса відправника.
- Destination Address - IP-адреса отримувача.

Таблиця 2.1 - Коди протоколів

Код	Протокол	Опис
1	ICMP	Протокол контрольних повідомлень
2	IGMP	Протокол керування групою хостів
3	GGP	Протокол «шлюз-шлюз»
4	IP	IP поверх IP (інкапсуляція)
6	TCP	TCP
8	EGP	Протокол зовнішньої маршрутизації (застарів)
9	IGP	Протокол внутрішньої маршрутизації (застарів)
17	UDP	UDP
27	RDP	Протокол надійних даних
28	IRTP	Протокол міжмережевої надійної передачі
46	RSVP	Протокол резервування ресурсів при мультикастингу
88	IGRP	Протокол внутрішньої маршрутизації компанії CISCO
89	OSPF	Протокол внутрішньої маршрутизації

- Options - опції, поле змінної довжини. Опцій може бути одна, декілька або жодної. Опції визначають додаткові послуги модуля IP по

обробці дейтаграми, в заголовок якої вони включені.

- Padding - вирівнювання заголовку по границі 32-бітного слова, якщо список опцій займає не ціле число 32-бітних слів. Поле «Padding» заповнюється нулями.