НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут комп’ютерних інформаційних технологій

Кафедра комп'ютерних систем та мереж

# Лабораторна робота № 2

з дисципліни «Комп’ютерні мережі»

Виконав: Клокун В. Д. Роботу захищено:

Група: СП-425

Перевірив: Зіньков Ю.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2019

**Основні протоколи транспортного і міжмережного рівнів стеку TCP/IP**

**Протоколи транспортного рівня**

Протоколи транспортного рівня можуть вирішувати проблему негарантованої доставки повідомлень ("чи дійшло повідомлення до адресата?"), а також гарантувати правильну послідовність приходу даних. У стеці TCP / IP транспортні протоколи визначають, для якого саме застосування призначені ці дані.

Протоколи автоматичної маршрутизації, логічно представлені на цьому рівні (оскільки працюють поверх IP), насправді є частиною протоколів мережного рівня; наприклад OSPF (IP ідентифікатор 89).

TCP (IP ідентифікатор 6) - "гарантований" транспортний механізм з попереднім встановленням з'єднання, що надає додатком надійний потік даних, що дає впевненість у безпомилковості одержуваних даних, перезапитує дані у випадку втрати та усуває дублювання даних. TCP дозволяє регулювати навантаження на мережу, а також зменшувати час очікування даних при передачі на великі відстані. Більш того, TCP гарантує, що отримані дані були відправлені точно в такій же послідовності. В цьому його головна відмінність від UDP.

UDP (IP ідентифікатор 17) протокол передачі датаграм без встановлення з'єднання. Також його називають протоколом "ненадійної" передачі, в сенсі неможливості упевнитися в доставку повідомлення адресату, а також можливого перемішування пакетів. У додатках, що вимагають гарантованої передачі даних, використовується протокол TCP.

UDP зазвичай використовується в таких додатках, як потокове відео і комп'ютерні ігри, де допускається втрата пакетів, а повторний запит утруднений або не виправданий, або в додатках виду запит-відповідь (наприклад, запити до DNS), де створення з'єднання займає більше ресурсів, ніж повторна відправка.

І TCP, і UDP використовують для визначення протоколу верхнього рівня число, зване портом.

**Мережевий рівень**

Мережевий рівень спочатку розроблений для передачі даних з однієї (під) мережі в іншу. Прикладами такого протоколу є X.25 і IPC в мережі ARPANET.

З розвитком концепції глобальної мережі в рівень були внесені додаткові можливості по передачі з будь-якої мережі в будь-яку мережу, незалежно від протоколів нижнього рівня, а також можливість запитувати дані від віддаленої сторони, наприклад в протоколі ICMP (використовується для передачі діагностичної інформації IP -з'єднання) і IGMP (використовується для управління multicast -потоками).

ICMP і IGMP розташовані над IP і повинні потрапити на наступний - транспортний - рівень, але функціонально є протоколами мережевого рівня, і тому їх неможливо вписати в модель OSI.

Пакети мережевого протоколу IP можуть містити код, який вказує, який саме протокол наступного рівня потрібно використовувати, щоб витягти дані з пакета. Це число - унікальний *IP-номер протоколу.* ICMP і IGMP мають номери, відповідно, 1 і 2.

До цього рівня відносяться: DHCP , DVMRP, ICMP, IGMP, MARS, PIM, RIP, RIP2, RSVP

**Порти і сокети, UDP-дейтаграми, TCP-сегменти, їх формати.**

Як і у протоколів мережевого і канального рівнів, одна з найважливіших функцій протоколу транспортного рівня полягає в ідентифікації протоколу або процесу, передачі їм даних. В TCP і UDP ідентифікація здійснюється за номером порту (Port), який призначений даному процесу організацією IANA (Internet Assigned Numbers Authority). Номери портів опубліковані в RFC 1700. Крім того, список часто використовуваних портів міститься в текстовому файлі SERVICES, що поставляється з кожним клієнтом TCP / IP. Коли пакет TCP / IP досягає мети, протокол транспортного рівня, який прийняв дейтаграмму, зчитує номер порту з поля Destination Port і передає інформацію з поля даних програмою або протоколу, асоційованого з цим портом.

Всім основних програм Інтернету привласнені номери портів, які називаються добре відомими портами (Well-known ports). Їх список наведено в таблиці 7.1. Наприклад, порт Web-сервера - 80, а порт сервера DNS - 53. TCP і UDP підтримують окремі списки номерів добре відомих портів. Наприклад, протокол FTP використовує порти TCP 20 і 21. Так як на транспортному рівні протокол FTP користується тільки протоколом TCP, інші протоколи прикладного рівня можуть використовувати порти 20 і 21 з протоколом UDP. У деяких випадках протокол прикладного рівня може користуватися обома протоколами транспортного рівня. **DNS,**наприклад, пов'язаний як з портом TCP 53, так і з портом UDP 53.

Адресуючи трафік іншій системі, TCP / IP-система використовує комбінацію IP-адреси і номера порту, яка називається сокетом (Socket).

Добре відомі порти, опубліковані в RFC 1060, відносяться, головним чином, до серверів. Оскільки зв'язок з даним сервером ініціює зазвичай клієнт (а не навпаки), клієнтам постійні номери портів не потрібні. На час зв'язку з конкретним сервером клієнтська програма зазвичай випадковим чином вибирає тимчасовий номер порту (Ephemeral port number). IANA контролює тільки номери портів в межах від 1 до 1023, тому тимчасові номери портів приймають значення від 1024 і вище. Сервер, який отримав пакет від клієнта, використовує значення з поля Source Port заголовка TCP, щоб адресувати відповідь правильному тимчасовому порту клієнтської системи.

Передача потоків даних по ТСР відбувається з встановленням з'єднання між взаємодіючими станціями. При встановленні ТСР з'єднання станції обмінюються номерами портів, синхронізують номери переданих пакетів і резервують пам'ять (буфер) для прийому і передачі інформаційних повідомлень. При передачі по протоколу ТСР відбувається перевірка прийнятих дейтаграм, і, якщо дейтаграмма втрачена або прийнята з помилками, то відбувається її повторна передача. Для цього при передачі сегмента даних копію цих даних поміщають в чергу повтору передачі і запускають таймер очікування підтвердження. Коли станція отримує підтвердження прийому (зворотна дейтаграмма, що містить прапор АСК), то цей сегмент видаляють з черги. Якщо підтвердження не надходило до спрацьовування таймера, сегмент відправляють ще раз.

При передачі інформації кожному байту даних привласнюють порядковий номер. Тому, незалежно від порядку прийнятих даних все байти будуть зібрані в приймаючої станції в початкової послідовності.

Формат заголовка ТСР пакета :

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| порт відправника | порт одержувача |  | | | | | | | | | |  |
| Порядковий номер |  |
| номер підтвердження |  |
| зсув даних | резерв | U  R  G | A  C K | P  C  H | R  S  T | F  I  N | вікно |  | | |  | |
| Контрольна сума | покажчик терміновості |  |  |  |  |  |  |
| Опції | заповнення |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | |

**порти відправника** и **одержувача** визначають порти станцій, куди слід адресувати дейтаграми при двосторонньому обміні.

При передачі фрагментованих файлів за допомогою декількох дейтаграм **порядковий номер** (32 біта) - це номер першого октету даних в цьому пакеті, якщо не встановлено прапор SYN. Якщо прапор SYN встановлений, т. Е. Передають першу дейтаграмму фрагментированного файлу, то порядковий номер стає номером початку послідовності ISN, а номер першого октету буде ISN +1.

**номер підтвердження** (32 біта) - номер наступного першого октету даних, очікуваних одержувачем пакета.

**зсув даних** (4 біта) - число 32 бітових слів заголовка TCP.

**резерв** (6 біт) - зарезервоване поле.

Далі йдуть 6 прапорців по 1 біту кожен.

**URG** - Прапор терміновості.

**ACK** - Прапор пакета, що містить отримання підтвердження.

**PCH** - Прапор форсованої відправки.

**RST** - Скидання з'єднання.

**SYN** - Синхронізація порядкових номерів.

**FIN** - Прапор кінця передачі з боку відправника.

**Контрольна сума** (16 біт) дозволяє виявити наявність помилок в прийнятій дейтаграми.

**вікно** (16 біт) - число байт, які відправник може прийняти до свого буфер, рахуючи від байта з номером, зазначеному в полі **номер підтвердження**.

**покажчик терміновості** (16 біт) - містить номер пакета, починаючи з якого йдуть пакети підвищеної терміновості.

**Опції** - Додаткові параметри.

**Протокол UDP** забезпечує негарантовану доставку даних, тобто не вимагає підтвердження їх отримання. Даний протокол не вимагає встановлення з'єднання між джерелом і приймачем повідомлень.

Формат заголовка UDP пакету

|  |  |
| --- | --- |
| порт відправника | порт одержувача |
| довжина | Контрольна сума |

Крім **портів відправника** и **одержувача**, Заголовок UDP дейтаграми містить **довжину** UDP пакету і **контрольну суму** - Поле перевірки правильності передачі заголовка пакета і поля корисного навантаження. Якщо це поле не використовують, його заповнюють нулями.

**Управління потоком**

**Управління потоком передачі даних** - в комп'ютерних мережах, механізм, який пригальмовує передавач даних при неготовності приймача.

Існує два підходи до вирішення цієї проблеми:

**Керування потоком зі зворотнім зв’язком (**[**англ.**](https://uk.wikipedia.org/wiki/Англійська_мова)***feedback-based flow control*)**коли отримувач відсилає передавачу інформацію що дозволяє йому продовжити передачу, чи повідомляє як загалом йдуть справи.

**Керування потоком з обмеженням(**[**англ.**](https://uk.wikipedia.org/wiki/Англійська_мова)***rate-based flow control*)** коли передавачі обмежуються в швидкості передачі даних, а зворотній зв’язок з приймачем відсутній.

Розрізняють три основних способи вирішення цієї проблеми:

* Апаратний, при якому сигнали «готовий /зайнятий» передаються по окремих фізичних лініях зв'язку. Найбільш відома така реалізація в інтерфейсі RS-232.
* Програмний, при якому програмний прапорець «готовий /зайнятий» зводиться і скидається вставкою в потік даних спеціальної унікальної послідовності (XOn / XOff). Застосовується в програмних драйверах інтерфейсу RS-232 як альтернатива апаратному контролю потоку у випадках неповного з'єднувального кабелю.
* Протокольний, при якому програмний прапорець «готовий /зайнятий» зводиться і скидається спеціальними угодами в рамках протоколу обміну даними. На сьогодні є практично єдиним застосовуваним способом контролю потоку. Найбільш відомий приклад - реалізація контролю потоку в протоколі TCP методом ковзного вікна.

**Формат IP-пакету**

За специфікацією протоколу, пакет має бути не більший за 65535 бітів (з заголовком та даними включно).

* **Версія (Version)** — 4-бітове поле, що описує використовувану версію протоколу IP. Всі пристрої зобов'язані використовувати протокол IP однієї версії, пристрій що використовує іншу версію буде відкидати пакети.
* **Довжина IP-заголовку (IP header Length — HLEN)** — 4-бітове поле, що описує довжину заголовку пакету в 32-бітових блоках. Це значення — це повна довжина заголовку з врахуванням двох полів змінної довжини.
* **Тип обслуговування (Type of Service — TOS)** — 8-бітове поле, що вказує на ступінь важливості інформації, що привласнена протоколом верхнього рівня.
* **Загальна довжина (Total Length)** — 16-бітове поле, що описує довжину пакету в байтах, із заголовком та даними включно. Для того щоб вирахувати довжину блока даних, потрібно від повної довжини відняти значення поля HLEN.
* **Ідентифікація (Identification)** — шістнадцятибітове поле, що зберігає ціле число, яке описує даний пакет. Це число являє собою послідовний номер.
* **Прапорці (Flags)** — 3-бітове поле, в якому два молодших біта контролюють фрагментацію пакетів. Перший біт визначає чи було пакет фрагментовано, а другий чи є цей пакет останнім фрагментом в серії фрагментів.
* **Зміщення фрагментації (Fragment Offset)** — 13-бітове поле, що допомагає зібрати разом фрагменти пакетів. Це поле дозволяє використовувати 16 бітів в сумі для прапорів фрагментації.
* **Час життя (Time-to-Live — TTL)** — 8-бітове поле — лічильник, в якому зберігаються послідовно зменшуване значення кількості пройдених вузлів (роутерів, що їх ще іноді в цьому випадку називають хопами (hops)) на шляху до місця призначення. У випадку коли лічильник пройдених хопів дорівнюватиме нулю — пакет буде відкинуто, таким чином попереджується нескінченна циклічна пересилка пакетів.
* **Протокол (Protocol)** — 8-бітове поле, що вказує на те, який протокол верхнього рівня отримає пакет, після завершення обробки IP-протоколом. Наприклад TCP або UDP.
* **Контрольна сума заголовку (Header Checksum)** — 16-бітове поле, що допомагає перевірити цілісність заголовку пакету.
* **IP-адреса відправника (Source IP address)** (адресант, сорс, відправник) — 32-бітове поле, що зберігає IP-адресу вузла-відправника.
* **IP-адреса отримувача (Destination IP adress)** (адресат, дест, отримувач) — 32-бітове поле, що зберігає адресу вузла призначення (отримувача).
* **Опції (Options)** — поле змінної довжини, що дозволяє протоколу IP реалізувати підтримку різних опцій, зокрема засобів безпеки.
* **Підкладка (Padding)** — поле, що використовується для вставки додаткових нулів, для гарантування кратності IP-заголовку 32 бітам.
* **Дані (Data)** — поле змінної довжини (64 Кбіт макс.), що зберігає інформації для верхніх рівнів.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Біти 0-3** | **4-7** | **8-15** | **16-18** | | **19-23** | | **24-31** | |
| Версія | HLEN | Тип обслуговування | Загальна довжина | | | | | |
| Ідентифікація | | | | Прапорці | | Зміщення фрагментації | | |
| Час життя | | Протокол | Контрольна сума заголовку | | | | | |
| IP-адреса відправника | | | | | | | | |
| IP-адреса отримувача | | | | | | | | |
| Опції | | | | | | | | Додаток |
| Дані (65535 мінус заголовок) | | | | | | | | |
| … | | | | | | | | |

IP-пакет складається з даних протоколу верхнього рівня і заголовку, що має описану вище структуру. Хоча основною частиною заголовку є адреси відправника і призначення, саме інші частини заголовку роблять протокол таким надійним і гнучким. Інформація, що зберігається в полях заголовку задає дані пакету і призначена для протоколів верхніх рівнів.

**Фрагментація IP-пакетів**

Протокол IP дозволяє виконувати фрагментацію пакетів, що надходять на вхідні порти маршрутизаторів.

Слід розрізняти фрагментацію повідомлень у вузлі-відправнику і динамічну фрагментацію повідомлень у транзитних вузлах мережі - маршрутизаторах. Практично у всіх стеках протоколів є протоколи, які відповідають за фрагментацію повідомлень прикладного рівня на такі частини, які укладаються в кадри канального рівня. У стеці TCP / IP це завдання вирішує протокол TCP, який розбиває потік байтів, що передається йому з прикладного рівня на повідомлення потрібного розміру (наприклад, на 1460 байт для протоколу Ethernet). Тому протокол IP у вузлі-відправнику не використовує свої можливості по фрагментації пакетів.

А ось при необхідності передати пакет в наступну мережу, для якої розмір пакета є занадто великим, IP-фрагментація стає необхідною. У функції рівня IP входить розбиття занадто довгого для конкретного типу складової мережі повідомлення на більш короткі пакети зі створенням відповідних службових полів, потрібних для подальшого складання фрагментів у вихідне повідомлення.

У більшості типів локальних і глобальних мереж значення MTU, тобто максимальний розмір поля даних, в яке повинен инкапсулировать свій пакет протокол IP, значно відрізняється. Мережі Ethernet мають значення MTU, що дорівнює 1500 байт, мережі FDDI - 4096 байт, а мережі Х.25 найчастіше працюють з MTU в 128 байт.

IP-пакет може бути позначений як і фрагментований. Будь-пакет, позначений таким чином, не може бути фрагментований модулем IP ні за яких умов. Якщо ж пакет, позначений як і фрагментований, не може досягти одержувача без фрагментації, то цей пакет просто знищується, а вузлу-відправнику надсилається відповідне ICMP-повідомлення.

Протокол IP допускає можливість використання в межах окремої підмережі її власних коштів фрагментирования, невидимих для протоколу IP. Наприклад, технологія АТМ ділить надходять IP-пакети на осередки з полем даних в 48 байт за допомогою свого рівня сегментування, а потім збирає осередки в вихідні пакети на виході з мережі. Але такі технології, як АТМ, є скоріше винятком, ніж правилом.

Процедури фрагментації і збірки протоколу IP розраховані на те, щоб пакет міг бути розбитий на практично будь-яку кількість частин, які згодом могли б бути знову зібрані. Одержувач фрагмента використовує поле ідентифікації для того, щоб не переплутати фрагменти різних пакетів. Модуль IP, що відправляє пакет, встановлює в поле ідентифікації значення, яке має бути унікальним для даної пари відправник - одержувач, а також час, протягом якого пакет може бути активним в мережі.

Поле зсуву фрагмента повідомляє одержувачеві положення фрагмента в початковому пакеті. Зсув фрагмента і довжина визначають частину вихідного пакета, принесену цим фрагментом. Прапор «more fragments» показує поява останнього фрагмента. Модуль протоколу IP, що відправляє нерозбитий на фрагменти пакет, встановлює в нуль прапор «more fragments» і зміщення у фрагменті.

Ці поля дають достатню кількість інформації для складання пакета.

Щоб розділити на фрагменти великий пакет, модуль протоколу IP, встановлений, наприклад, на маршрутизаторі, створює кілька нових пакетів і копіює вміст полів IP-заголовка з великого пакета в IP-заголовки всіх нових пакетів. Дані з старого пакета діляться на відповідне число частин, розмір кожної з яких, окрім самої останньої, обов'язково повинен бути кратним 8 байт. Розмір останньої частини даних дорівнює отриманому залишку.

Кожна з отриманих частин даних поміщається в новий пакет. Коли відбувається фрагментація, то деякі параметри IP-заголовка копіюються в заголовки всіх фрагментів, а інші залишаються лише в заголовку першого фрагмента. Процес фрагментації може змінити значення даних, розташованих в поле параметрів, і значення контрольної суми заголовка, змінити значення прапора «more fragments» і зміщення фрагмента, змінити довжину IP-заголовка і загальну довжину пакета, В заголовок кожного пакета заносяться відповідні значення в поле зміщення « fragment offset », а в поле загальної довжини пакету поміщається довжина кожного пакета. Перший фрагмент матиме в полі «fragment offset» нульове значення. У всіх пакетах, крім останнього, прапор «more fragments» встановлюється в одиницю, а в останньому фрагменті - в нуль.

Щоб зібрати фрагменти пакета, модуль протоколу IP (наприклад, модуль на хост - комп'ютері) об'єднує IP-пакети, що мають однакові значення в полях ідентифікатора, відправника, одержувача та протоколу. Таким чином, відправник повинен вибрати ідентифікатор таким чином, щоб він був унікальний для даної пари відправник-одержувач, для даного протоколу і протягом того часу, поки даний пакет (або будь-який його фрагмент) може існувати в складовою IP-мережі.

Очевидно, що модуль протоколу IP, що відправляє пакети, повинен мати таблицю ідентифікаторів, де кожен запис співвідноситься з кожним окремим одержувачем, з яким здійснювався зв'язок, і вказує останнє значення максимального часу життя пакета в IP-мережі. Однак, оскільки поле ідентифікатора допускає 65 536 різних значень, деякі хости можуть використовувати просто унікальні ідентифікатори, які не залежать від адреси одержувача.

У деяких випадках доцільно, щоб ідентифікатори IP-пакетів вибиралися протоколами більш високого, ніж IP, рівня. Наприклад, в протоколі TCP передбачена повторна передача ТСР - сегментів, з яких-небудь причин не дійшли до адресата. Імовірність правильного прийому збільшувалася б, якби при повторній передачі ідентифікатор для IP-пакета був би тим же, що і в початковому IP-пакеті, оскільки його фрагменти могли б використовуватися для складання правильного ТСР - сегмента.

Процедура об'єднання полягає в приміщенні даних з кожного фрагмента в позицію, зазначену в заголовку пакета в поле «fragment offset».

Кожен модуль IP повинен бути здатний передати пакет з 68 байт без подальшої фрагментації. Це пов'язано з тим, що IP-заголовок може включати до 60 байт, а мінімальний фрагмент даних - 8 байт. Кожен одержувач повинен бути в змозі прийняти пакет з 576 байт в якості єдиного шматка або у вигляді фрагментів, що підлягають зборці.

Якщо біт прапора заборони фрагментації (Do not Fragment, DF) встановлено, то фрагментація даного пакета заборонена, навіть якщо в цьому випадку він буде втрачений. Даний засіб може використовуватися для запобігання фрагментації в тих випадках, коли хост - одержувач не має достатніх ресурсів для складання фрагментів.

**IPv6 як розвиток стеку TCP/IP**

Спільнота Internet вже кілька років [працює](http://ua-referat.com/Працює) над розробкою нової специфікації для базового протоколу стека - протоколу IP. Вироблено вже досить багато пропозицій, від простих, передбачають лише розширення адресного простору IP, до дуже складних, що призводять до істотного збільшення вартості реалізації IP у високопродуктивних (і так недешевих) маршрутизаторах.

Основним [пропозицією](http://ua-referat.com/Пропозиція) щодо [модернізації](http://ua-referat.com/Модернізм) протоколу IP є [пропозиція](http://ua-referat.com/Пропозиція), розроблена групою IETF. Зараз прийнято називати її пропозицію версією 6 - IPv6, а всі інші пропозиції групуються під назвою IP Next Generation, IPng.

У пропозиції IETF [протокол](http://ua-referat.com/Протокол) IPv6 залишає основні принципи IPv4 незмінними. До них відносяться дейтаграммний метод роботи, фрагментація пакетів, дозвіл відправнику задавати максимальну кількість хопов для своїх пакетів. Однак, в деталях реалізації протоколу IPv6 є істотні відмінності від IPv4. Ці відмінності коротко можна описати таким чином.

Використання більш довгих адрес. Новий розмір адреси - найбільш помітне відміну IPv6 від IPv4. Версія 6 використовує 128-бітові адреси.

Гнучкий формат заголовка. Замість заголовка з фіксованими полями фіксованого розміру (за винятком поля Резерв), IPv6 використовує базовий заголовок фіксованого формату плюс набір необов'язкових заголовків різного формату.

Підтримка резервування пропускної здатності. В IPv6 механізм резервування пропускної здатності замінює механізм класів сервісу версії IPv4.

Підтримка розширюваності протоколу. Це одне з найбільш значних змін у підході до побудови протоколу - від повністю деталізованого опису протоколу до протоколу, який дозволяє підтримку додаткових функцій.

## Адресація в IPv6

Адреси призначення і джерела в IPv6 мають довжину 128 біт або 16 байт. Версія 6 узагальнює спеціальні типи адрес версії 4 в таких типах адрес:

Unicast - індивідуальний адресу. Визначає окремий вузол - комп'ютер або порт маршрутизатора. Пакет повинен бути доставлений вузлу за найкоротшим маршрутом.

Cluster - адреса кластеру. Позначає групу вузлів, які мають загальний адресний префікс (наприклад, приєднаних до однієї фізичної мережі). Пакет повинен бути маршрутизувати групі вузлів по найкоротшому шляху, а потім доставлений тільки одному з членів групи (наприклад, найближчого вузла).

Multicast - адреса набору вузлів, можливо в різних фізичних мережах. Копії пакету повинні бути доставлені кожному вузлу набору, використовуючи апаратні можливості групової або широкомовної доставки, якщо це можливо.

Як і у версії IPv4, адреси у версії IPv6 діляться на класи, залежно від значення кількох старших біт адреси.

Велика частина класів зарезервована для майбутнього застосування. Найбільш цікавим для практичного використання є клас, призначений для провайдерів послуг Internet, названий Provider-Assigned Unicast.

Адреса цього класу має наступну структуру:

010 Ідентифікатор провайдера Ідентифікатор абонента Ідентифікатор підмережі-Ідентифікатор вузла

Кожного провайдера послуг Internet призначається унікальний ідентифікатор, яким позначаються всі підтримувані їм мережі. Далі провайдер призначає своїм абонентам унікальні ідентифікатори, і використовує обидва ідентифікатора при призначенні блоку адрес абонента. Абонент сам призначає унікальні ідентифікатори своїм подсетям і вузлів цих мереж.

Абонент може використовувати техніку підмереж, застосовувану у версії IPv4, для подальшого розподілу поля ідентифікатора підмережі на більш дрібні поля.

Описана схема наближає схему адресації IPv6 до схем, що використовуються в територіальних мережах, таких як [телефонні](http://ua-referat.com/Телефон) мережі або мережі Х.25. Ієрархія адресних полів дозволить магістральним маршрутизаторам працювати тільки зі старшими частинами адреси, залишаючи обробку менш значущих полів маршрутизаторам абонентів.

Під поле ідентифікатора вузла потрібно виділення не менше 6 байт, для [того](http://ua-referat.com/Того) щоб можна було використовувати в IP-адреси МАС-адреси локальних мереж безпосередньо.

Для забезпечення сумісності зі схемою адресації версії IPv4, у версії IPv6 є клас адрес, що мають 0000 0000 в старших бітах адреси. Молодші 4 байти адреси цього класу повинні містити адресу IPv4. Маршрутизатори, підтримуючі обидві версії адрес, повинні забезпечувати трансляцію при передачі пакету з мережі, що підтримує адресацію IPv4, в мережу, підтримуючу адресацію IPv6, і навпаки.