

# Тема. IP-адресація в стеку протоколів TCP/IP. Багатоадресне розсилання

IP-адреса (Internet Protocol адреса) використовується 2-м рівнем стеку (профілю) TCP/IP. Відповідно, мережі з IP-адресацією називаються IP-мережами.

Зокрема, Internet, базуючись на профілі TCP/IP, представляє собою з'єднання маршрутизаторами (роутерами) окремих IP-мереж (які більш точно треба називати *IP-підмережами*, або *первинні мережі*, або *приватні мережі*). IP-мережа повинна мати унікальну IP-адресу, а вузол (*хост*, *робоча станція* тощо) в такій мережі повинен мати адресу (номер) в мережі.

Отже, IP-адреса є парою (*Адреса\_мережі*, *Адреса\_вузла\_в\_мережі*). Ще використовується термінологія для такої пари (*Префікс*, *Суфікс*), тобто *префікс* адреси є адресою мережі, а *суфікс*, відповідно, адресою вузла в мережі. Також адресація передбачає множину адрес спеціального призначення, інтерпретація котрих має особливості, зокрема, такі адреси "працюють" лише в первинній мережі та за межі її не повинні взагалі "виходити".

## Розмір адреси

З 1969 р. почала використовуватися 32-бітна адреса, яка з початку 1980-х років отримала позначення IPv4 і використовується досі. Коли Internet був у зародковому стані (приблизно до початку 1990-х, до появи Web та браузерів), 32-бітна адреса з більше ніж 4 мільярдами значень здавалася з нескінченим запасом адрес на багато десятиліть. Проте вже буквально через кілька років розвитку WWW та е-пошти, стало зрозуміло, що адрес вистачить до року 2010.

З 2000 р. почався поступовий перехід до 128-бітної адреси, позначення якої IPv6. Планувалося, що 2012 року всі провайдери 1-го рівня повністю перейдуть на IPv6, проте цей процес тривав до 2016 року. Зрештою, планується замінити IPv4 повністю на IPv6 на рівні всіх провайдерів. Поточний залишок IP-адрес v4 (тобто 32-бітні) у світі див. <https://ipv4.potaroo.net/>.

## Версія адресації IPv4

32-бітна версія IPv4 неминуче потребує заміни на IPv6, але професіоналам в сфері IT з нею ще доведеться працювати не один рік. Справа тут в тому, що IPv6 почне використовуватися провайдерами 1-го рівня, потім 2-го і т.д., до тих, що "роздають" Internet в офіси та квартири. А на цьому рівні користувачів заміна на IPv6 програм та маршрутизаторів, розрахованих на IPv4, може рухатися дуже повільно. Тому провайдери "останньої милі" будуть змушені з клієнтами продовжувати працювати на IPv4 ще досить довго. **Отже, добре знати IPv4 ще потрібно років з 5!**

Адресація IPv4 пройшла 2 фази поділу адресного простору між користувачами:

- 1) адресація з розподілом на класи мереж, яка виявилася неошадливою при швидкому поширенні Internet;
- 2) сучасна (поки немає повного переходу на IPv6) безкласова адресація з масками підмереж.

## Запис адрес IPv4 в десятковій точковій формі

32-бітне число в оригінальному вигляді важко записувати та запам'ятовувати, хоча програми та обладнання потребують оригінального 32-бітного представлення. Для спрощення та скорочення запису використовується **десятькова точкова форма**:

- 1) 32 біти розбивають на 4 октети (байти по 8 біт);
  - 2) кожний октет записують не в бітах, а відповідним 10-м беззнаковим числом в діапазоні значень 0..255;
  - 3) між 10-ми значеннями октетів ставляться точки. Октети із нулів, залишаючи замість них лише точки, пропускати не можна.
- Нумерація октетів (і бітів) йде зліва-направо, самим старшим є 1-й октет, потім 2-й і т.д. (тобто як на письмі - самою старшою цифрою є сама ліва).

Наприклад:

32-бітна адреса така (для зручності кольорами розбита на октети): 00011001110010101110011010100110

№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
№ октетів (байтів) адреси	1								2								3								4									
Біти адреси	0	0	0	1	1	0	0	1	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0
10-ві значення октетів	25								202								230								166									

Отже, десятковий точковий запис адреси буде такий: 25.202.230.166

## Адресація на основі розділення IP-мереж на класи

Ця адресація є застарілою і вже давно не використовується. Про неї просто потрібно мати уявлення.

В адресації з розділенням на класи вся множина адрес розбивається на 5 класів мереж A, B, C, D, E.

Три перших класи A, B та C є основними для присвоєння первинним мережам, класи D, E мають спеціальне призначення. Класи визначають розміри мережі (кількість вузлів).

### Клас A

A № бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Поля адреси мережі й хоста	мережа								хост																							
Значення певних бітів	0																															

Мереж класу A може бути: 126

Вузлів (хостів) може бути: 16'777'214

Значення 1-го октету: 1..126 (мережі 0 та 127 мають спеціальне призначення)

Зрозуміло, що ефективно оперувати мережею у більше ніж 16 мільйонів адрес важко. У свій час вважалося, що це буде державна мережа з "гігантською" кількістю вузлів. І якби класова система адресація діяла б і зараз, то 126 мереж не вистачило не тільки на великі за населенням країни, а і на малі, в яких легко може бути комп'ютерів (смартфонів, комунікаторів, планшетів) більше за 16 млн. Якщо на країну виділити кілька A-мереж, то можуть залишатися у резерві багато адрес останньої мережі, які передати у використання іншій країні важко реалізуємо (на рівні маршрутизаторів).

### Клас B

В	№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	№ байтів (октетів) адреси	1								2								3								4							
	Поля адреси мережі й хоста	мережа																хост															
	Значення певних бітів	1	0																														

Мереж класу А може бути: 16384  
 Вузлів (хостів) може бути: 65534  
 Значення 1-го октету: 128..191

#### Клас С

C № бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Поля адреси мережі й хоста	мережа																								хост							
Значення певних бітів	1	1	0																													

Мереж класу С може бути: 2'097'151  
 Вузлів (хостів) може бути: 254  
 Значення 1-го октету: 192..223

Ці мережі планувались для малих офісів/будинків. Але що таке 2 млн. офісів на весь світ? Для України це вже буде мало.

#### Клас D

D № бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Поля адреси мережі й хоста					адреса групової розсилки																											
Значення певних бітів	1	1	1	0																												

Значення 1-го октету: 224..239  
 Ці адреси не присвоювалися первинним мережам.

#### Клас E

E № бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Поля адреси мережі й хоста					значення не уточнюється																											
Значення певних бітів	1	1	1	1																												

Значеннями 1-го октету: 240..255  
 Це експериментальний клас адрес. Толком невідомо, чи він встиг взагалі якось використовуватися чи ні.

Як видно, адресація з розбивкою на класи дає можливість оперувати мережами не менше ніж із 254 хостами, і таких мереж, до того ж, трошечки більше 2 млн. Процес розширення Internet почав дуже швидко вичерпувати наявну кількість адрес. Зрозуміло, що краще взагалі відмовитися від розбивки адрес на класи. Фактично про адресацію з класами мереж достатньо знати, що так було

## Адресація на основі масок IP-підмереж (безкласова адресація)

Проблема неощадливості адресації з розділенням на класи криється в розбивці 32 біт адреси на префікс та суфікс по границі октету - вони можуть бути рівними 1, 2 або 3 октетам. Надалі було запропоновано на префікс та суфікс відводити довільну кількість бітів (із суфіксом не менше 2 біт). Щоб знати де в конкретній IP-адресі префікс (адреса мережі), а де суфікс (адреса хоста) до адреси добавили так звану *маску підмережі* (Subnet Mask), яка:

- 1) складається теж із 32 біт,
- 2) на місці префікса (бітів адреси мережі) мають стояти одинички, а на місці суфікса нулі,
- 3) в запису маска розбивається на октети з десятковим точковим форматом.

Тоді IP-адреса (як мережі, так і конкретного хоста) записується з такими двома варіантами:

- 1) IP-адреса/Маска\_підмережі або
- 2) IP-адреса/Бітів у префіксі.

На значення адреси хоста накладаються додаткові обмеження:

- 1) всі біти суфікса із одиниць не є конкретною адресою хоста (вузла, станції), а зверненням до всіх в мережі, так звана, **широкомовна адреса**;
- 2) всі біти суфікса із нулів стосуються лише тих станцій, які є **шлюзами** в мережі. Шлюзами є станції, які зв'язані з іншими мережами, тобто через них можливий вихід за межі своєї мережі, а ззовні потрапити в дану мережу.

Наприклад, адреса **156.248.27.141/255.255.224.0**, або вона ж в такій формі **156.248.27.141/19** :

№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
IP-адреса	156								248								27								141							
	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	1
Маска підмережі	префікс адреси																суфікс адреси															
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	255								255								224								0							
Адреса мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	156								248								0								0							

Отже, про адресу **156.248.27.141/19** ми можемо сказати, що це є адреса хоста в мережі 156.248.0.0/19 .

Діапазон адрес хостів в мережі 156.248.0.0/19 такий: 156.248.0.1-156.248.31.254, широкомовна адреса в цій мережі - 156.248.31.255:

№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Адреса мережі	156								248								0								0							
	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Маска підмережі	префікс адреси																суфікс адреси															
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	255								255								224								0							
Найменша адреса хоста в мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	156								248								0								1							

Найбільша адреса хоста в мережі	1 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0	156	248	31	254
Широкомовна адреса в мережі	1 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1	156	248	31	255

## Зверніть увагу, що:

- 1) по адресі 156.248.27.141 без вказання маски підмережі ми нічого (тобто *адреса мережі*, *адреса вузла* в ній) сказати не можемо! В залежності від маски це будуть різні мережі та їх хости!
- 2) потрібно вирізняти адресу мережі (156.248.0.0) і адресу вузла в мережі (156.248.27.141), але це все лише за наявності маски.
- 3) адреса 156.248.27.141 не має ніякого відношення до мереж класу В, бо це вже безкласова адресація! На запитання *про клас мережі* потрібно вже не звертати увагу на перший октет, відповідаючи про відсутність прив'язки до класів в адресації з масками підмереж (зараз часто вже говорять *мереж* замість довшого слова *підмереж*)

Наприклад, візьмемо ту ж комбінацію 32 бітів адреси, але маску в 26 біт. Тоді адреса повинна виглядати так: **156.248.27.141/255.255.225.192**, або **156.248.27.128/26**, тому що:

№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
IP-адреса	156								248								27								141							
	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	1
Маска підмережі	префікс адреси																										суфікс					
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	255								255								255								192							
Адреса мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
	156								248								27								128							

Отже, про адресу **156.248.27.128/255.255.225.192** (вона ж **156.248.27.128/26**) тепер потрібно сказати, що це є хост (вузол) в мережі **156.248.27.128/26**.

Діапазон адрес хостів в мережі 156.248.27.128/26 такий: 156.248.27.129-156.248.27.190, широкомовна адреса в цій мережі - 156.248.27.191:

№ бітів адреси	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
№ байтів (октетів) адреси	1								2								3								4							
Адреса мережі	156								248								27								128							
	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
Маска підмережі	префікс адреси																										суфікс					
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	255								255								255								192							
Найменша адреса хоста в мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	1
	156								248								27								129							
Найбільша адреса хоста в мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	1	0
	156								248								27								190							
Широкомовна адреса в мережі	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1
	156								248								27								191							

Звичайно ж маска вибирається не довільно, - це виконується на рівні Internet комітетом **IANA** (Адміністрацією адресного простору Internet, яка є підрозділом [ICANN](#)) при

виділенні адрес провайдерам (ISP). Тут різні маски - взяті лише для демонстрації механізму "вилучення" даних із адреси. Маршрутизатори в своїх таблицях інформації про мережі (таблицях маршрутизації) зберігають адреси мереж з їх масками. Тоді IP-пакет, який поступає в маршрутизатор, може бути ідентифікований зі своєю мережею.

## Спеціальні адреси

Це адреси, які або потребують спеціальної інтерпретації і стосуються групи вузлів, або не повинні попадати в глобальний трафік.

- 1) 0.0.0.0 - адреса власника пакета.
- 2) 255.255.255.255 - загальна широкомовна адреса.
- 3) 127.0.0.0 - 127.255.255.255, або [127.0.0.0/8](#) - адреси для хоста, вони не покидають навіть вузол і використовуються для відладки/використання мережесхем програм навіть без наявності мережі. Зокрема, адреса 127.0.0.1 є адресою *localhost*, ім'ям комп'ютера, програми якого використовують адреси мережі 127. Ще одне ім'я адреси 127.0.0.1 - *loopback* адреса (замкнена на себе адреса)

**Адреси, які призначені лише для адресації в підмережах (приватних мережах)**, наприклад, для внутрішньої адресації в локальній, корпоративній мережі, мережі провайдера доступу в Інтернет тощо:

- 4) 10.0.0.0 - 10.255.255.255, або [10.0.0.0/8](#)
- 5) 192.168.0.0 - 192.168.255.255, або [192.168.0.0/16](#)
- 6) 169.254.0.0 - 169.254.255.255, або [169.254.0.0/16](#)
- 7) 172.16.0.0 - 172.31.255.255, або [172.16.0.0/16](#) - [172.31.0.0/16](#)

Самі вживані перші дві спецадреси.

В глобальну мережу пакети з такими адресами не повинні попадати. При виході за межі первинних мереж адреси в пакетах мають підмінятися на реальні IP-адреси через протокол [NAT](#) (протокол трансляції мережесхем адрес). Якщо "випадково" пакети із внутрішніми адресами все ж попадуть за межі первинної мережі, то вони мають ігноруватися роутерами. І це означатиме неправильну настройку шлюзів приватної мережі, зокрема, що не включена NAT.

Можна адреси, маска в яких проходить по границі октету, записувати у вигляді *net.host*, наприклад, 10.h.h.h, 192.168.h.h тощо, вказуючи явно лише октети префіксу.

Якраз в приватних мережах з префіксами адрес 10, 192.168 тощо можна розбивати їх масками на сегменти адрес довільним чином (тобто без дозволу IANA). Наприклад, в LAN створюємо сегмент 10.40.h.h для одного підрозділу, 10.63.17.h для іншого підрозділу, 192.168.h.h для всіх інших користувачів (як правило, з динамічним присвоюванням адреси). На рівні корпоративної (приватної, первинної) мережі є достатній запас адрес для самостійного використання.

*Дивіться про IPv4 у [Wikipedia](#).*

## Версія адресації IPv6

128-бітна адресація IPv6, маючи, як зараз здається, необмежений ресурс адрес, все значно спрощує. Бо основна проблема IPv4 крилася в економії адрес, а тепер і в їх повному вичерпанні. (Правда, у 1980-х роках IPv4 теж вважався з невичерпною кількістю адрес :-)) Зокрема, запас адрес дозволить в разі невдалого поточного принципу їх використання легко перейти на інший. В IPv6 немає необхідності в NAT.

## Запис адреси в IPv6

16 октетів адреси IPv6 для десяткової точкової форми мали б незручний громіздкий вигляд і він не використовується. В IPv6 використовується більш лаконічний запис, наприклад, ось типова адреса:  
2001 : 05e3 : b11f : 00ca : d6b8 : ca9a : 7cc2 : 5d5a  
Отже значення розбиваються символами : на 8 груп по 2 октети, октети записуються цифрами 16-ї системи числення (символи : записані через пропуски лише для зручності).

## Особливості запису:

1	Ведучі нулі у групі можуть пропускатися. Так наведена вище адреса може бути скорочена: 2001: <b>5e3</b> :b11f: <b>ca</b> :d6b8:ca9a:7cc2:5d5a
2	Якщо група складається із нулів, -вона може бути пропущена, і на її місці має бути :: . Також всі підряд нульові групи заміщаються одним скороченням :: і таке :: може бути лише одне на всю групу. Наприклад, 2001:: - всі групи після першої із нулів, або ::5d39 - всі групи, крім останньої, із нулів
3	Розмір у бітах префікса адреси, як і в IPv4, записується після адреси через слеш: 2001:05e3:b11f:00ca:d6b8:ca9a:7cc2:5d5a/ <b>64</b>
4	В URL-адресах значення IP-адреси IPv6 береться в []: http://[2001::5d5a]
5	Аналог адреси 0.0.0.0 в IPv4 може бути в IPv6 записана так: ::
6	loopback адреса (аналог 127.0.0.1) позначається так: ::1
7	Прямий аналог адреси IPv4 xx.xx.xx.xx в десятковій точковій формі записується так: ::xx.xx.xx.xx Для таких адрес префікс /96 .

## Структура адреси IPv6

128 біт розбиті на такі поля:

ПРЕФІКС (номер мережі)					ІНТЕРФЕЙСИ				
gggg	:	rrrr	:	rrrr	:	rrrr	:	rrrr	:
Тип адреси	Дані провайдера (адреса провайдера)					Ідентифікатор інтерфейсу (типово за основу береться MAC-адреса)			

Тут для зручності 8 груп адреси представлені як gggg : rrrr : rrrr : rrrr : aaaa : aaaa : aaaa : aaaa , де на місці *g*, *p* та *a* може бути будь-яка 16-ва цифра.  
Перша група (на схемі gggg) задає тип адреси. Значення, які достатньо пам'ятати:

Значення gggg	Яка це адреса
2001	реальна (глобальна) адреса, має оброблятися всіма маршрутизаторами
2002	реальна (глобальна) адреса, яка отримана з адреси IPv4, і призначена для проходження (тунелювання) через підмережі (провайдерів), що не підтримують IPv6. Формат таких адрес (специфікація <i>6to4</i> ): <b>2002:rrrr:rrrr::/48</b> , де замість <i>p</i> мають бути відповідні 16-ні цифри чотирьох байтів IPv4 адреси. Тобто на місці даних провайдера (rrrr:rrrr:rrrr) і підставляються 4 байти IPv4 адреси, а наступні 2 байти із нулів (rrrr:rrrr:0000). Наприклад, адреса 92.11.1.1 в IPv6 буде такою: 2002:5c0b:101::/48
fe80 - febf	адреса первинної мережі (типово fe80), пакети з такою 1-ю групою не повинні попадати в глобальний трафік. Є 2 варіанти для локальних адрес. 1) специфікація <i>6to4</i> , де адресу представляємо так: <b>fe80:rrrr:rrrr::/48</b> , коли 16-й аналог IPv4 знаходиться зразу за fe80. Наприклад, для 192.168.1.107 маємо fe80:c0a8:16b::/48 2) на практиці частіше використовується представлення: <b>fe80::aaaa:aaaa/1xx</b> , коли 16-й аналог IPv4 знаходиться в кінці (замість MAC-адреси). Наприклад, для 192.168.1.107 маємо fe80::c0a8:16b/120. Тут маска підмережі 120, бо для мереж 192.168.n.h маскою є 255.255.255.0 (тобто адреси хостів знаходяться лише у 8 бітах в кінці).
ffxx	широкомовна адреса (xx - будь-які цифри)

Дивіться про IPv6 у [Wikipedia](#).  
[Тест роботи і швидкості з'єднання з використанням IPv6](#)

## Утиліти ОС, пов'язані з IP-адресацією

Як мінімум, потрібно вміти користуватися утилітами (командами) *ipconfig*, *ping* та *tracert* (в MS Windows, або аналогічними для Unix/Linux).

### ipconfig

Утиліта командного процесора **ipconfig** відображає конфігурацію IP-адресації хоста. Ось скорочений лістинг запуску утиліти:

```
d:\>ipconfig
Windows IP Configuration

Ethernet adapter Підключення через локальну мережу:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::ddfb:f58d:3468:5be3%11
    IPv4 Address. . . . . : 192.168.0.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.11

Wireless LAN adapter Безпроводове мережне підключення:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Tunnel adapter isatap.{B36310C6-421A-428A-A4A0-71A9F398DF21}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Утиліта дає інформацію по всім підключенням, зокрема, щодо провідного підключення Ethernet маємо:

```
IPv4 адреса:      192.168.0.100
маска підмережі:  255.255.255.0
шлюз за умовчанням: 192.168.0.11
IPv6 адреса:      fe80::ddfb:f58d:3468:5be3
```

Потрібно пам'ятати, що комп'ютер може мати скільки завгодно IP-адрес (по кожному підключенню (безпроводне, кабельне, мобільне тощо) буде своя адреса, яку присвоїть відповідний провайдер). Так шлюз в мережі повинен мати по адресі в кожній мережі, з якою він зв'язаний (це вже буде мінімум 2 адреси - у "своїй" мережі і в яку має вихід).

Утиліта має багато опцій, з якими потрібно самостійно познайомитися через підказку:

**ipconfig /?**

Аналогом *ipconfig* в Linux є команда *ip*.

### ping

Утиліта дозволяє встановити існування певної IP-адреси, відсилкою до неї певної кількості луна-пакетів (Echo-Request). Вузол-адресат має дати відповідь на кожний отриманий луна-пакет. Відсутність відповіді означає або відсутність адреси, або неможливість дати відповідь (перевантаженість, відключенням луна-пакетів на сервері тощо).

Додатково по IP-адресі можна встановити [доменне ім'я](#) (hostname), якщо таке є у адреси. Рівно і як по доменному імені його IP-адресу.

Командний рядок утиліти **ping** (можна отримати по *ping /?*):

```
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] | [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name

Options:
  -t          Ping the specified host until stopped.
              To see statistics and continue - type Control-Break;
              To stop - type Control-C.
  -a          Resolve addresses to hostnames.
  -n count    Number of echo requests to send.
  -l size     Send buffer size.
  -f          Set Don't Fragment flag in packet (IPv4-only).
  -i TTL      Time To Live.
```



-v TOS	Type Of Service (IPv4-only. This setting has been deprecated and has no effect on the type of service field in the IP Header).
-r count	Record route for count hops (IPv4-only).
-s count	Timestamp for count hops (IPv4-only).
-j host-list	Loose source route along host-list (IPv4-only).
-k host-list	Strict source route along host-list (IPv4-only).
-w timeout	Timeout in milliseconds to wait for each reply.
-R	Use routing header to test reverse route also (IPv6-only).
-S srcaddr	Source address to use.
-4	Force using IPv4.
-6	Force using IPv6.

Адреса вузла може бути задана як числова так і доменна.

Утиліта виставляє тайм-аут на відповідь "пінгуємо!" адреси. Якщо цього значення за умовчанням виявиться мало, то опцією -w можна задати новий тайм-аут (як у прикладі 30 секунд):

```
ping -w 30000 univ.kiev.ua
```

Ось результат пінгування адреси univ.kiev.ua:

```
d:\>ping univ.kiev.ua

Pinging univ.kiev.ua [91.202.128.71] with 32 bytes of data:
Reply from 91.202.128.71: bytes=32 time=2ms TTL=58
Reply from 91.202.128.71: bytes=32 time=1ms TTL=58
Reply from 91.202.128.71: bytes=32 time=1ms TTL=58
Reply from 91.202.128.71: bytes=32 time=1ms TTL=58

Ping statistics for 91.202.128.71:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Опція -t встановлює режим постійного пінгування, опція -l задає розмір пінг-пакета (повинен бути меншим за 1500 байт). За допомогою таких опцій команда *ping* часто використовується для кібератак. Наприклад:

```
ping -t -l 1400 адреса_вузла
```

запущена з декількох тисяч вузлів (наприклад, бот-мережі = [ботнет](#), навіть без відома власників комп'ютерів) здатна "покласти" будь-який мережевий ресурс. Це одна із самих простих в реалізації **DDoS/DoS**-атак. Є поширеною практикою не відповідати на пінги (як захисний механізм, але адреса приймати їх буде все одно!), тому відсутність позитивного пінгування ще не означає відсутність реального ресурсу. Так у браузері адреса буде відкриватися, а для пінгів - нібито як відсутня.

**В експерименті** з пінгуванням для досягнення відчутного навантаження на мережеве з'єднання певного хоста потрібно запустити на атакуючому вузлі значну кількість утиліт *ping*. Це зручно зробити у вікні командного рядка введенням такого рядка:

```
for /L %j in (1,1,50) do start ping -t -l 30000 адреса_вузла
```

В наведеному прикладі на паралельне виконання запускається 50 команд *ping -t -l 30000 адреса\_вузла*. Кожна така команда буде запускатися в окремому екземплярі командного рядка і, маючи параметр -t, працюватиме постійно. (Якщо наведений вище цикл запускати в окремому командному файлі (з розширенням .bat чи .cmd), то змінну циклу %j потрібно задавати з двома знаками %: %%j.) Щоб не закривати вручну всі такі вікна, можливо, є сенс запускати утиліту не з параметром -t, а з -n (тобто виконати задану кількість пінгів, після цього завершиться утиліта із автоматичним закриттям вікна екземпляру командного рядка):

```
ping -n 500 -l 60000 адреса_вузла
```

Аналогом *ping* в Linux є така ж команда *ping* (до речі, з Unix вона і пішла світом).

Про *ping* у [Wikipedia](#)

## tracert

Утиліта дозволяє побачити маршрут проходження пакету до заданого адресою (IP чи доменна) вузла. Враховуючи, що профіль TCP/IP використовується в дейтаграмних пакетних мережах, що означає індивідуальну маршрутизацію кожного пакета (дейтаграми). Тому два запуски підряд *tracert* можуть відрізнятись маршрутами! Утиліта використовує [хоп](#)-метрику мережі, в якій проходження вузла (тут буде маршрутизатора) = 1 хопу. Отже результатом буде відстань в хопах до вказаної адреси та адреса кожного хопу (стрибка) і час очікування хопу.

Командний рядок утиліти *tracert*:

```
Usage: tracer [-d] [-h maximum_hops] [-j host-list] [-w timeout]
           [-R] [-S srcaddr] [-4] [-6] target_name
```

Options:

```
-d          Do not resolve addresses to hostnames.
-h maximum_hops  Maximum number of hops to search for target.
-j host-list  Loose source route along host-list (IPv4-only).
-w timeout    Wait timeout milliseconds for each reply.
-R          Trace round-trip path (IPv6-only).
-S srcaddr    Source address to use (IPv6-only).
-4          Force using IPv4.
-6          Force using IPv6.
```

Опція *-h* може задавати максимальну кількість хопів, які буде чекати утиліта. Річ у тому, що дейтаграма може "заблудитися" (попасти в цикл із декількох роутерів), і очікування завершення встановлення маршруту до цільової адреси може бути непередсказуємим. Зазвичай, хопів буває менше ніж 30 за умовчанням.

Опція *-w* регулює час очікування кожного хопу.

Ось трасування адреси **google.com.ua**:

```
d:\trace google.com.ua

Tracing route to google.com.ua [74.125.232.216]
over a maximum of 30 hops:

 1  <1 ms  <1 ms  <1 ms  192.168.0.11
 2  <1 ms  <1 ms  <1 ms  10.20.5.1
 3  1 ms   2 ms   1 ms   gate.access.net.ua [195.3.158.9]
 4  2 ms   1 ms   2 ms   UA-9816-vl3.bg.net.ua [193.227.206.46]
 5  1 ms   1 ms   1 ms   united-bg.bg.net.ua [193.111.9.121]
 6  5 ms   1 ms   1 ms   adamant-10G-gw.ix.net.ua [195.35.65.223]
 7  1 ms   2 ms   2 ms   google-adamant.adamant.ua [212.26.128.246]
 8  25 ms  26 ms  25 ms  72.14.239.14
 9  26 ms  25 ms  25 ms  216.239.46.88
10  25 ms  25 ms  26 ms  bud01s08-in-f24.1e100.net [74.125.232.216]

Trace complete.
```

А ось та ж доменна адреса **google.com.ua** через деякий час:

```
d:\trace google.com.ua

Tracing route to google.com.ua [173.194.70.94]
over a maximum of 30 hops:

 1  <1 ms  <1 ms  1 ms   192.168.0.11
 2  1 ms   <1 ms  <1 ms  10.20.5.1
 3  1 ms   <1 ms  <1 ms  gate.access.net.ua [195.3.158.9]
 4  2 ms   2 ms   2 ms   UA-9816-vl3.bg.net.ua [193.227.206.46]
 5  1 ms   2 ms   3 ms   united-bg.bg.net.ua [193.111.9.121]
 6  2 ms   1 ms   2 ms   adamant-10G-gw.ix.net.ua [195.35.65.223]
 7  1 ms   2 ms   1 ms   google-adamant.adamant.ua [212.26.128.246]
 8  26 ms  3 ms   2 ms   209.85.241.55
 9  26 ms  38 ms  25 ms  66.249.94.139
10  40 ms  39 ms  32 ms  64.233.175.213
11  39 ms  42 ms  39 ms  209.85.254.116
12  *     40 ms *    72.14.236.68
13  38 ms  38 ms  40 ms  209.85.254.112
14  *     *    39 ms fa-in-f94.1e100.net [173.194.70.94]

Trace complete.
```

Як бачимо, навіть IP-адреса змінилася, відповідно і маршрут (проте маршрут може змінюватися і без зміни IP-адреси!). Це пов'язано з тим, що великі портали виконують балансування навантаження, перенаправляючи запити на різні сервери, але для користувачів це все **google.com.ua**.

Аналогом *tracert* в Linux є команда *traceroute*.

## Контрольні питання

1. Яка локальна IP-адреса комп'ютера?
2. Яка глобальна IP-адреса комп'ютера?
3. Яка поточна швидкість з хостом в *Сінгапурі*?
4. Виписати всі IPv4 адреси мережі **184.4.96.90/255.255.255.240** .
5. Скоротити IPv6 адресу *2002:0003:b100:000a:06b8:c00a:0002:5000* .
6. Записати IPv6 адресу *2002:0003::5000* в URL-посиланні.
7. Записати IPv4 адресу *10.0.253.11* у форматі IPv6.
8. Записати IPv4 адресу *105.0.253.11* у форматі IPv6.
9. Що можна сказати про адресу IPv4 *10.40.253.211* ?
10. Що можна сказати про адресу IPv6 *fe80:0003::5000* ?
11. Для первинної мережі потрібний пул не більше ніж 64 адрес в мережі *10.h.h.h* . Запропонуйте варіант такої мережі (тобто *адреса/маска*).
12. Знайти IP-адресу ресурсу *tv.net.ua* .
13. Пропінгувати сусідній комп'ютер з 5% його завантаження; визначити параметри атаки (розмір пакета, кількість атакуючих вузлів).
14. Як на протязі 5-10 запусків tracert змінювався маршрут до *tv.net.ua* ?
15. Запустити програму Wireshark.Відібрати для аналізу 3 кадри: 1 персональний (unicast), 1 груповий (multicast), 1 ширококомовний (broadcast). Всі ці кадри скопіювати у звіт, позначивши в кожному з них MAC-адресу відправника, MAC-адресу отримувача, тип або довжину кадру. Для фільтрації ширококомовних кадрів необхідно в меню Analyze/Display Filter вибрати фільтр «Ethernet broadcast». Для фільтрації групових кадрів необхідно створити новий фільтр з назвою «Ethernet multicast». Для цього необхідно в меню Analyze/Display Filter вибрати New, у полі Filter Name набрати «Ethernet multicast», напроти Filter String натиснути кнопку Expression. У полі Field Name знайти рядок «Ethernet». Далі вибрати підрядок eth.multicast, у полі «Relation» вибрати «=», в полі «Predefined Values» вибрати «This is a multicast frame». Аналогічно створюється фільтр для unicast.

Для спостереження за трафіком multicast почергово використовуйте:фільтр захоплення, фільтр відображення, фільтр мережевого рівня, фільтр канального рівня.