УДК 654(045)

Ткаліч О.П., канд. техн. наук
Нечипорук О.П., канд. техн. наук
Андрухович П.О.
Таран Є.С.
Бліновська М.Л.

АНАЛІЗ ЗАТРИМОК В КАНАЛІ ЗВ'ЯЗКУ В ЗАЛЕЖНОСТІ ВІД ЙОГО ЯКОСТІ

Національний авіаційний університет

Проведено аналіз затримок в каналі зв'язку в залежності від його якості (використовуючи утиліту ping), що дозволило дослідити способи використання утиліти ping, яка входить до стандартного комплекту усіх операційних систем й поставки мережевого обладнання та реалізації простого контролю цілісності каналу зв'язку

Вступ

Утиліта *ping* заснована на міжмережевому протоколі *ICMP* (*Internet Control Message Protocol*) керуючих повідомлень, що входить до стеку *TCP/IP*. Протокол *ICMP* не гарантує доставку повідомлень, для цих цілей використовуються інші мережеві протоколи, наприклад *TCP*. *ICMP* не є самостійним протоколом, формально він використовує *IP* пакети для передачі *ICMP* повідомлень методом інкапсуляції *ICMP* пакетів в *IP* пакети. Головним завданням даного протоколу є передача повідомлень про помилки, а також використання в інших виняткових ситуаціях, які виникають при передачі даних:

– знаходження помилки у заголовку *IP* пакету, відсутність адреси;

– відсутність маршруту до адресата;

– перевірка можливості доставки *IP* пакетів (*exo* запит) – утиліта *ping*;

– утиліта traceroute показує маршрут IP пакетів до адресата;

– оновлення таблиць маршрутизації;

– керування швидкістю.

Основні матеріали

Також на нього покладають деякі службові функції. Однією з таких функцій є утиліта *ping*, що досліджується. Утиліта *ping* відправляє запити *ICMP Echo-Request* протоколу *ICMP* до певних вузлів, та фіксує відгуки, які надходять від них *ICMP Echo-Reply*. Робота утиліти визначена стандартом *RFC* 2925 (*Definitions of Managed Objects for Remote Ping, Traceroute and Lookup Operations*).

Важливими параметрами є час між відправкою запиту та отриманням відгуку (*RTT – Round Trip Time*), що дає змогу визначати двосторонні затримки по маршруту слідування пакета. Не слід плутати *ping* з передачею інформації від адресата до адресанта і навпаки. У випадку відсутності відгуку від вузла найчастіше адресат блокує дані запити або блокує. Третім варіантом розвитку подій є блокування чи ігнорування цих пакетів будь-яким маршрутизатором на маршруті слідування пакетів (запитів та відгуків як поодиноко, так і відразу обидва) від відправника до одержувача.

Дана утиліта хоч і має порівняно простий устрій та призначення, але допомагає у розв'язанні багатьох проблем у практичному застосуванні в реальних мережах. При розгляді способів використання утиліти вважатимемо, що *ICMP* пакети не блокуються і не ігноруються на жодному з вузлів.

Усі операції можливо проводити з певними параметрами. До цих параметрів відносяться:

- *t* надсилання запитів безперервно. Для паузи, перегляду статистики і продовження необхідно натиснути *Control-Break*. Для зупинки натиснути *Control-C*;

- *a* передача адреси до імені хостів;

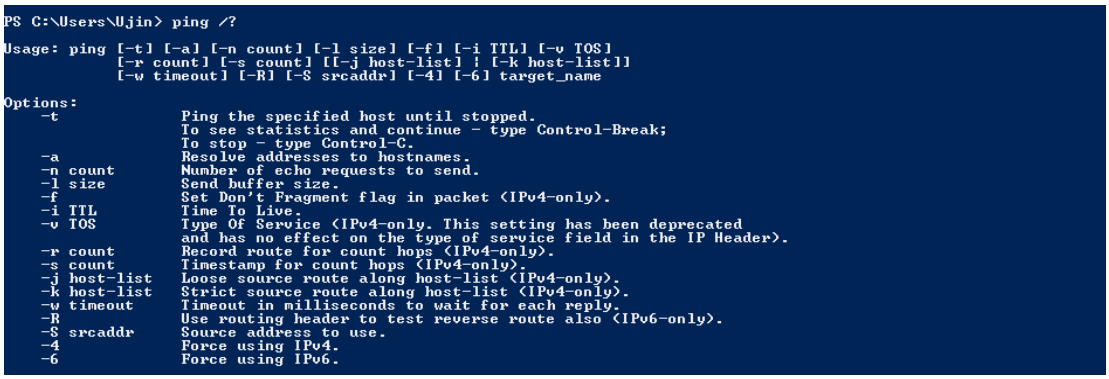

Рис. 1. Опис аргументів утиліти

- *n* кількість запитів;

- *l* розмір exo запитів (від 0 до 65500 байт). Без його зазначення стандартним є розмір в 32 байта;

- *f* забороняє фрагментацію пакета exo запиту (лише для *IPv4*);

- *I* встановлення «часу життя» пакету exo запиту;

- *v* *TOS* являється застарілим і майже не впливає. Він розглядатись не буде;

- *r* для запису кількості переходів по маршруту до цільового пристрою (лише для *IPv4*);

- *s* кількість часу для стрибків по маршруту слідування exo запиту (лише для *IPv4*);

- *j* звільнити вихідний маршрут вздовж певного списку (лише для *IPv4*);

- *k* заборонити маршрут вздовж певного списку (лише для *IPv4*);

- *w* час очікування відповіді від цільового пристрою в мілісекундах;

- *R* використання заголовка для тестування реверсивного марштруту; (лише для *IPv6*);

- *S* використання адреси джерела;

- 4 примусове використання *IPv4*;

- 6 примусове використання *IPv6*.

Ці параметри дають можливість конфігурувати утиліту *ping* в залежності від конкретно поставленої задачі. Для автоматичного періодичного збору даних стосовно стабільності каналу зв'язку слід використовувати параметр – *n*. Регулювання розміру та заборона фрагментації матиме сенс при специфічному налаштуванні мережевого обладнання. Заборона фрагментації – при тестуванні зв'язку з серверами *Oracle* в силу високої довжини пакетів, які вони використовують при обміні. Так для об'єкту, який знаходиться за відносно довгим маршрутом, є сенс застосувати параметр збільшення часу очікування та збільшення *TTL*. Пакети великого розміру можливо використовувати для перевірки коректної фрагментації їх на маршрутизуючому обладнанні мережі.

За допомогою даної утиліти можна дізнатися про стан роботи сервера наступним чином: з деякого пристрою достатньо відправити відповідний *ping* – запит та отримати час відгуку та кількість відгуків, відповідно до кількості виконаних запитів. У ідеальному випадку, коли адресант працює у нормальному режимі та на маршруті слідування до нього не відбудеться жодної втрати *ICMP* пакету з exo-запитом, то при цьому кількість відправлених запитів має дорівнювати кількості одержаних відгуків від адресата запиту. Слід відмітити, що чим менший час відгуку, тим краще.

Для наглядної демонстрації перевірки стану роботи сервера чи будь-якого мережевого пристрою використали сервер з *IPv4* адресою 192.168.199.13/24 (маска підмережі 255.255.255.0) та продемонструємо різні випадки стану його роботи.

У ідеальному випадку, коли мережеве обладнання працює справно та без затримок, отримали наступні результати:

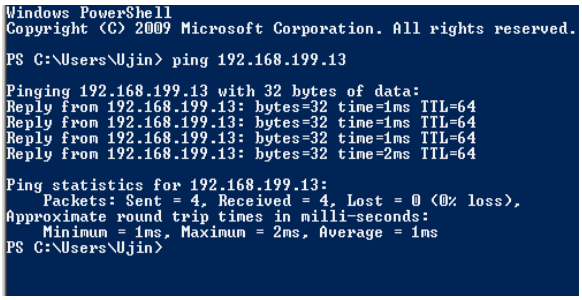

Рис. 2. Приклад результатів «ідеальної» роботи мережі

Як бачимо, усі запити надійшли до об'єкта, про що свідчить усі 4 відповіді. Слід відмітити час відгуку 1мс, що вказує на високу швидкість обладнання чи близьке розташування вузлів між якими виконувався тест. В нашому випадку вагомішу роль зіграв другий фактор, тому що в даному випадку був взятий сервер з нашої ж підмережі.

Параметр *TTL* (*Time to live*) «Час життя пакета» – максимальний період часу (число інтерацій, переходів), за який набір даних (пакет) може існувати до зникнення. Кожний маршрутизатор по маршруту слідування може як не змінювати це число, так і змінювати на одиницю чи більше значення в залежності від його позиції, як географічної так і мережевої. Коли *TTL* стає рівним нулю, то відправнику відправляється *ICMP* пакет з кодом «11» «Перевищення часового інтервалу». Такий підхід дозволяє уникнути ситуацій, коли пакет циркулює завжди по мережі, перевантажуючи її.

В випадку недоступності вузла чи блокуванні на ньому *ICMP* пакетів отримали наступний результат:

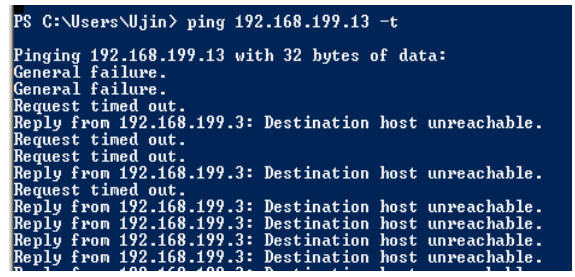

Рис. 3. Приклад недоступності вузла

Варіанти отримання відповідей типу «*General failure*», що свідчить про помилку, а також «*Request timed out*», що свідчить про перевищення часу очікування відповіді. Виходячи з таких результатів, можна розглядати недоступність вузла чи блокування ICMP з певних причин. Але частіше система доповідає про помилку «*Request timed*» *out* чи «*General failure*», а не в їх комбінації, як показано в прикладі.

Утиліта *ping* також може визначити стан *DNS* серверу. Відомо, що в мережах активно використовуються доменні імена, які за допомогою сервісу *DNS* конвертуються у *IP* адреси вузла. Таким чином, відправивши запит *ping* до певного вузла, наприклад *example.com*, *DNS* сервер має конвертувати ім'я у *IP* адресу сервера і відправити за цією адресою відповідний *ICMP* ехо-запит. При нормальній роботі *DNS* все відбудеться прозоро, тобто спрацює *DNS*, відправляться запити та одержаться відгуки. При збої в цьому сервісі система видасть помилку про неможливість знайти вузол *example.com*. І, звісно, ніяких відгуків ми не одержимо, по причині того, що навіть не відбудеться передача ехо-запитів, так як не буде відомо кому їх відправляти (немає *IP* адреси одержувача).

Перевіряємо роботу *DNS* серверу на доменному імені *google.com*. (рис. 4.)

Як бачимо, при запиті адресованому вузлу *google.com* система звернулась до *DNS* серверу та одержала *IPv4* адресу цього вузла 74.125.87.104, якому далі і відправляла *ICMP* запити. Тут ми бачимо позитивні результати: адреса отримана, вузол доступний, зв'язок здається стабільним.

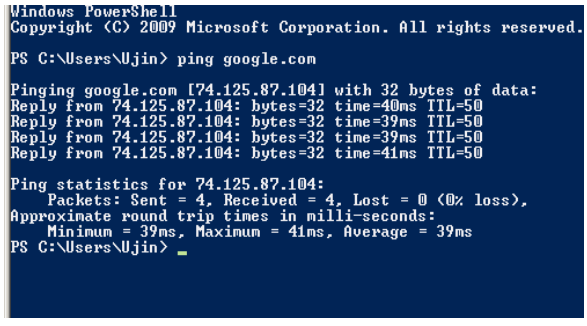

Рис. 4. Приклад результатів перевірки роботи *DNS*

При недоступності *DNS* серверу у мережі отримали:

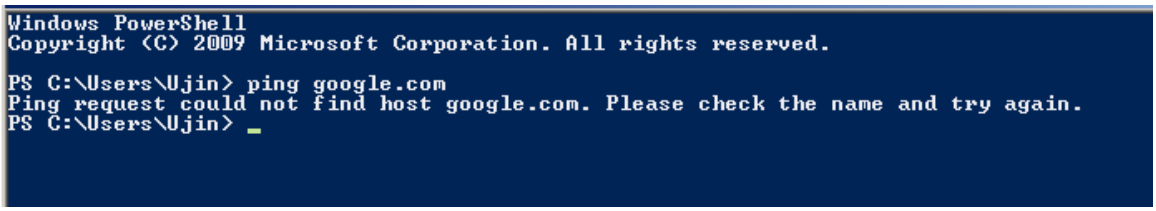

Рис. 5. Приклад недоступності *DNS*

Система має виконувати запити адресовані певному вузлу, адресою має виступати *IPv4* або *IPv6* адреса. Система не знайшла відповіді на запит цієї адреси щодо вузла google.com тому і видала користувачу вищевказану помилку.

За допомогою цієї команди (*ping*) можна оцінити якість роботи мережі. Утиліта *ping* відправляє кілька запитів, а точніше 4. В ідеалі всі 4 відправлені запити мають спричинити відповідні ехо-відгуки. Так всі 4 відгуки мають дійти по зворотному маршруту, причому з мінімальним часом відгуку. Такі показники вкажуть на високу якість роботи мережі. Ідеальними результатами можна умовно вважати час відгуку менше 1 мс (<1 *ms*) та 100% одержання усіх 4-х ехо-відгуків. В цьому випадку можна зробити висновки про достатню швидкість роботи мережевого обладнання за даного навантаження (навантаження в момент часу від відправки до одержання відгуку). В залежності від особливостей мережевого обладнання (наявність пам'яті та черги, алгоритму обробки мережевих запитів) в неідеальному випадку при перевантаженні можлива як часткова передача цих запитів та відгуків, так і передача з затримкою запитів та відгуків. При наявності у маршрутизатора ПК чи іншого мережевого пристрою пам'яті, який призначений для зберігання необроблених «мережевих запитів» при перевантаженні, пристрій «вистроїть чергу» і буде обробляти запити у порядку черги. В результаті чого ми отримаємо затримку сигналу ехо-відгуку (*Echo-Reply*) набагато більшу, ніж за відсутності навантаження на цей пристрій. Якщо ж пам'ять вже переповнена чи відсутня взагалі та при недостатній швидкості обробки мережевого пристрою, запити, які пристрій не встигає адекватно обробити, будуть відкинуті, і ми не отримаємо відгуків взагалі.

Причому важливо розуміти, що якщо з певних причин пристрій не отримає запит, то і відгук він не буде надсилати по причині непотрібності останнього. При затримках відгуків слід зазначити, що від відправки запиту до одержання відгуку система чекає порядку однієї секунди, потім відправляє наступний запит і так поки не буде відправлено 4 запити.

Також *ping* надає можливість оцінки мережевого з'єднання. Відомо, що загальна швидкість мережі визначається швидкістю найповільнішого (найслабшого) мережевого обладнання в мережі (маршрутизатор, світч, хаб, комутатор). У випадку коли до деякого вузла існують й альтернативні маршрути, то тут при перевантаженні одного пристрою запити будуть надсилатись по іншому маршруту чи відразу ділитись пропорційно пропускним здатностям обох чи більше маршрутів. У іншому випадку доведеться тестувати кожен вузол окремо (в залежності від топології мережі) чи оцінювати швидкість мережі взагалі. Тобто чим більше запитів в секунду вдалося обробити без втрати швидкості обробки, тим більшою є швидкість «найслабшого» вузла мережі. Чим більша швидкість, зрозуміло, тим краще. Існує немало прикладів, коли швидкість мережі є важливим параметром, а саме: для шифрування інформації, для мережі з підвищеною надійністю передавання інформації.

Оцінюючи якість мережі, по часу відгуку судять про затримки, а по кількості відгуків, що не прийшли, – про якість. У даної утиліти є можливість безперервно відправляти та одержувати відповідні за-

пити. Звісно, що кожному відправленому запиту має відповідати відгук від мережевого обладнання. Для оцінки якості слід скористатися вищевказаною можливістю, тобто запустити механізм безперервної відправки запитів та виконувати моніторинг результатів. Звісно, ідеальним результатом є одержання усіх відповідних відгуків до кожного запиту. По відсутності запитів часто судять про збої в роботі мережі, наприклад: збої в роботі мережевого обладнання, відсутність маршруту до адресата, збої *DNS*, рівень навантаження та перевантаження мережі взагалі, перевантаження конкретного вузла.

Перевіримо якість мережі, обравши об'єктом для перевірки домен google.com. Для більшої наглядності результату в наведеному прикладі відправляємо більше, ніж 4 стандартні запити:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Ujin> ping google.com -t

Pinging google.com [74.125.87.104] with 32 bytes of data:
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=41ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
```

Рис. 6. Приклад результатів перевірки з використанням DNS

З наведених результатів видно доволі високу якість роботи мережі. Усі запити 100% були оброблені вузлом з відповідною реакцією, про що свідчать стабільні результати в часі порядку 40мс. Стабільність такого показника як TTL, дорівнює в даному прикладі 50 і не змінюється, це свідчить про сталість маршруту передавання. Сукупність цих показників та географічної протяжності мережі (віддалений вузол розташований в США) дає змогу зробити висновок про стабільність мережі та її високу швидкість.

Наведемо приклад нестабільної мережі. У ній ми бачимо постійні збої у зв'язку та нестабільність обробки запитів:

```
Reply from 74.125.87.104: bytes=32 time=38ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=39ms TTL=50
Reply from 74.125.87.104: bytes=32 time=40ms TTL=50
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
Reply from 74.125.87.104: bytes=32 time=73ms TTL=50
Reply from 74.125.87.104: bytes=32 time=87ms TTL=50
Reply from 74.125.87.104: bytes=32 time=59ms TTL=50
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
General failure.
Reply from 74.125.87.104: bytes=32 time=42ms TTL=50
Reply from 74.125.87.104: bytes=32 time=54ms TTL=50
Reply from 74.125.87.104: bytes=32 time=106ms TTL=50
Reply from 74.125.87.104: bytes=32 time=41ms TTL=50
```

Рис. 7. Приклад нестабільної роботи мережі

Ми бачимо 2 обриви зв'язку, що, звісно, не може бути позитивним. Але, напевне, одним з найгірших варіантів можна надати такий приклад наведений на рис. 8.

Перебої у зв'язку, нестабільний та надзвичайно довгий час відгуку та високий TTL свідчить про надзвичайно низьку якість мережевого з'єднання. Таке з'єднання формально наявне, але по причині високих затримок і постійних зривів зв'язку вся його користь зводиться до нуля. Такий результат може виникнути з причини неприпустимої якості роботи майже будь-якого з компонентів мережі між 2-ма вузлами. Велика протяжність мережі, велика кількість проміжних вузлів маршрутизації та інших мережевих пристроїв, що активно використовуються в мережевому обміні, низька якість обладнання, невірні та неоптимальні налаштування, повне чи часткове блокування ICMP погіршують результати. Повільні вузли будуть збільшувати час відгуку та завдавати суттєвого впливу на значення TTL. Слід пам'ятати, що час відгуку може доходити аж до 2000 мс, а *TTL* до 300 мс, тому можливі значно гірші варіанти. «Пінгування» завжди виконується у режимі максимального пріоритету каналу, тому надмірне використання може призвести до перевантаження каналу зв'язку лише самими *ICMP* пакетами. В такому випадку про передачу інформації не може йти і мови. Саме тому часто адміністратори

мереж детально налаштовують мережеве обладнання, в тому числі і у даному питанні для уникнення зайвого мережевого трафіку та неможливості використання даних запитів для «загорання» каналу зв'язку зайвою та непотрібною інформацією як певної мережевої атаки, наприклад *DoS (Denial of Service*, відмова в обслуговуванні) чи *DDoS (Distributed Denial of Service*, розподілена атака типу «відмова в обслуговуванні») атаки. Часто *ICMP* частково чи повністю блокують на зовнішніх пристроях, таких як шлюзи, що розташовані у зовнішній мережі в тому числі і *NAT (Network Address Translation* – перетворення мережевих адрес).

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Ujin> ping google.com -t
Ping request could not find host google.com. Please check the name and try again.
PS C:\Users\Ujin> ping google.com -t
Ping request could not find host google.com. Please check the name and try again.
PS C:\Users\Ujin> ping google.com -t

Pinging google.com [74.125.87.147] with 32 bytes of data:
Reply from 74.125.87.147: bytes=32 time=41ms TTL=51
Reply from 74.125.87.147: bytes=32 time=40ms TTL=51
General failure.
General failure.
Reply from 74.125.87.147: bytes=32 time=229ms TTL=51
Request timed out.
Reply from 192.168.199.5: Destination host unreachable.
Reply from 74.125.87.147: bytes=32 time=41ms TTL=51
Reply from 192.168.199.5: Destination host unreachable.
Reply from 74.125.87.147: bytes=32 time=66ms TTL=51
Reply from 192.168.199.5: Destination host unreachable.
Reply from 74.125.87.147: bytes=32 time=77ms TTL=51
General failure.
General failure.
Reply from 74.125.87.147: bytes=32 time=56ms TTL=51
Reply from 74.125.87.147: bytes=32 time=48ms TTL=51
Reply from 192.168.199.5: Destination host unreachable.
Reply from 74.125.87.147: bytes=32 time=132ms TTL=51
```

Рис. 8. Приклад нестабільної роботи мережі та DNS

Висновки

В силу високого пріоритету, за результатами моніторингу за допомогою даної утиліти можливо проводити контроль як пошкоджень каналу зв'язку та обладнання так і атак на сам канал так і його сегменти. За певних обставин нестабільність зв'язку, як у прикладах наведених вище, можна розцінювати як одну з ознак хакерської атаки на мережу, наприклад типу «*DDOS*» або «спуфінг». Обрив зв'язку з подальшим його відновленням (без втручання спеціалістів) можна розцінювати як підключення зловмисника в мережу типу «людина посередині». Рекомендується використовувати такий метод як допоміжний, або резервний

В результаті дослідження даної утиліти на основі протоколу *IPv4* слід підкреслити її відносно простий принцип роботи, що підвищує її надійність. При цьому важливо, щоб мережеве обладнання не блокувало і не ігнорувало пакети *ICMP* запитів та відгуків. В противному випадку застосування утиліти стає неможливим, по причині відсутності вихідних даних. Ця утиліта має широкі можливості в сфері діагностики мереж, що використовують стек протоколів *TCP/IP*.

Перспективи подальших досліджень у даному напрямку слід спрямувати на аналіз нововведень протоколу *ICMPv6*.

Список літератури

1. *Таненбаум Э.* Компьютерные сети. – 4-е изд. – СПб.: Питер, 2003. – 992 с.

2. *Буров Є.* Комп'ютерні мережі. – 2-е основне і доповн. вид. – Львів: Бак, 2003. – 548 с.

Подано до редакції 11.10.2010