

УДК 004.45+004.94

А. В. Сербенюк, О. С. Куроп'ятник, к. т. н., доц.

(Український державний університет науки і технологій, Україна)

ФОРМУВАННЯ МЕРЕЖЕВОГО ТРАФІКУ З ВИКОРИСТАННЯМ ФОРМАЛЬНИХ ГРАМАТИК

A. V. Serbeniuk, O. S. Kuropiatnyk, Ph. D., Assoc. Prof.

TRAFFIC SHAPING BY USING FORMAL GRAMMAR

В сучасному світі важко уявити функціонування суспільства без мережі Інтернет. Її використання дозволяє вирішувати безліч проблем. Тому, забезпечення працездатності мережі Інтернет та її складових є важливим завданням. Це включає в себе безліч дій: від перевірки фізичної справності компонентів та тестування систем на вразливість до атак та нестандартного Інтернет-трафіку, до простої перевірки наявності з'єднання з мережею.

Для тестування трафіку існує чимало інструментів, які можуть його формувати. Здебільшого для їх використання користувачам надається графічний або консольний інтерфейс із фіксованим набором функцій, та способи використання програми обмежені тим, що в ньому надається. У випадку, коли наданого функціоналу недостатньо, виникає питання розширення функціоналу або пошуку альтернативних засобів.

Великим обмеженням в цьому питанні є те, що людина не завжди може точно сформулювати та передати вимоги до генерованого трафіку для програми, яка це робить. Проблема опису і передачі вимог між людьми вирішується засобами природної мови, тому пропонується рішення цієї проблеми з використанням деякої формальної мови і для представлення трафіку програмі-генератору. Такий спосіб повинен дозволити користувачу описувати вимоги до трафіку точніше та зручніше. Для опису такої мови та подальшої її автоматизованої обробки доцільним є використання формальних граматики автоматного типу.

Щоб надати таку ступінь гнучкості опису трафіку, повинна бути можливість доступу до багатьох рівнів моделі OSI. Механізми опису для будь якого рівня або протоколу може бути той же — у вигляді деякого об'єкту зі значеннями. Наприклад, може бути як об'єкт для опису базового мережевого пакету протоколу TCP, так і об'єкт для опису всього потоку пакетів у транзакціях протоколу HTTP.

Судячи із цього, синтаксис граматики, що описуватиме цю мову, повинен надавати можливість оголошення цих об'єктів, присвоєння значення їм та їхнім складовим частинам, доступ до їхніх складових частин. Крім того, ці об'єкти можуть мати власний функціонал, тому потрібно передбачити можливість доступу до нього. Тобто, синтаксис повинен надавати можливість виклику функцій, передачі їм аргументів. Прикладом використання подібних функцій може бути доступ до елементів об'єктів, які були створені або підраховані безпосередньо в процесі обробки вхідного тексту, як розмір деяких об'єктів, кількість пакетів у них, тощо.

Грамматика повинна описувати базові синтаксичні елементи мови, а безпосередньо функціонал буде вибудовуватись на її основі. Функціонал представлення трафіку буде надаватись у вигляді типів даних, які описують його або його складові. Наприклад, тип для представлення TCP пакету. Така грамматика описується четвіркою $G = \langle T, N, A, P \rangle$, де T – множина терміналів, що описує компоненти пакетів та команди, N – множина не терміналів, які використовуються як допоміжні елементи при формуванні ланцюжку-опису пакету, $A \in N$ – початковий символ (аксіома), P – множина правил продукції. Наведемо приклади елементів множини правил, де всі лексеми, що пишуться з великої букви, позначають нетермінали, символ « $\langle \rangle$ » – варіативність підстановки:

$A \rightarrow \text{Line}|\text{LineA}$,
 $\text{Line} \rightarrow \text{Command}|\text{CommandComment}$
 $\text{Command} \rightarrow \text{Declaration}|\text{Assignment}|\text{Access}$
 $\text{Declaration} \rightarrow \text{Type Var}|\text{Type Var Assignment}$

Оскільки мережевих протоколів та способів їх використання багато, то варто врахувати можливість розширення функціоналу програми за рахунок додаткових модулів та бібліотек.

В якості програмної реалізації формування трафіку на основі формальних граматик пропонується консольний додаток мовою C++, розробка якого триває. Розробка орієнтована на UNIX-подібні операційні системи.

Додаток отримуватиме вхідний текст для обробки із стандартного вхідного потоку, що дозволяє інтегрувати його в механізм конвеєрів UNIX-подібних операційних систем та надає свободу в способі безпосередньо передачі тексту в програму, від ручного введення, до зчитування із файлу, або обробки виведення іншого додатку.

Очікуваним результатом виконання програми є обробка вхідного тексту та команд із нього, тому вихідними даними додатку є індикація успішності обробки тексту. Формування описаного трафіку ініціюватиметься командою у вхідному тексті.

Додаток працює зі спроектованою граматикою. Синтаксис цієї граматики підтримує операції оголошення змінних деякого статичного типу, присвоєння значень, доступ до елементів змінних, виклику функцій. Вбудовані типи складаються із цілого числа, рядка тексту, та адреси протоколу IPv4. Типи для представлення трафіку оформлюватимуться у вигляді окремих модулів, які завантажуватимуться додатково. В межах даної розробки, буде розроблено типи для представлення протоколу TCP.

Для забезпечення гнучкості пропонується система плагінів. Плагін – це динамічна бібліотека із розширенням «.so», що містить інформацію про типи даних, які можуть використовуватись основним додатком. Це дозволяє розширювати функціонал додатку без внесення змін в нього самого, що також дозволяє уникати його рекомпіляції при цьому. Для підключення плагіну достатньо помістити його в директорію, де програма зможе його знайти. Шлях до директорії буде задаватись за допомогою змінної оточення, яку, за бажання, користувач також може змінити.

Результатом підвантаження плагіну є доступність нових типів представлення трафіку в мові, якими користувач може негайно ж користуватись. Відсутність сталого програмного інтерфейсу для представлення трафіку дозволяє уникнути необхідності його розширення,

Таким чином, перевагами додатку, що розробляється, є:

- більш природний спосіб опису складових трафіку за рахунок використання для цього граматики;
- простота автоматизації, що досягається завдяки сумісності і відповідності стандартам консольних додатків UNIX-подібних операційних систем;
- можливість розширення існуючого функціоналу за рахунок системи плагінів.

Основними напрямками для подальшої роботи є додавання підтримки різноманітних мережевих протоколів. Наявність функціоналу, відповідного аналогам, дозволить провести їх пряме порівняння.

У підсумку можна дійти висновку, що подібна розробка може бути корисною для представлення широкого спектру видів мережевого трафіку. Навіть якщо деякий вид трафіку не має готової імплементації, щоб його представити, її можна розробити та інтегрувати в існуючий програмний каркас, а текстове представлення трафіку значно полегшує доступність нового функціоналу користувачам, адже не потрібно додавати нові елементи інтерфейсу.