УДК: 004.457 004.728.4 519.684.6

К. С. Дєєв[1]
Ю. В. Бойко[1]

АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ ПАКЕТНОЇ ФІЛЬТРАЦІЇ ДЛЯ ГЛИБОКОГО АНАЛІЗУ МЕРЕЖЕВИХ ПАКЕТІВ

[1]Київський національний університет імені Тараса Шевченка

Наведено приклади застосування пакетних фільтрів для задач пріорітезації окремих типів мережевого трафіку, що може бути використано для дотримання вимог до пропускної спроможності та гарантування високих показників якості обслуговування. Оцінено можливість використання відкритого програмного забезпечення для виконання цих задач.

Ключові слова: мережевий пакетний фільтр, аналіз трафіку, класифікація мережевої взаємодії, апаратний комплекс аналізу.

Вступ

Проблема класифікації Інтернет-трафіку на високошвидкісних з'єднаннях набуває все більшого поширення зі зростанням популярності мережевих додатків, взаємодія між якими ґрунтується на моделі з'єднань точка–точка. Необхідність класифікації такого типу взаємодії зумовлена потребою зменшення пріоритетності такого трафіку у порівнянні із застосуваннями реального часу та телефонії.

Більшість неспеціалізованого апаратного та програмного забезпечення без застосування додаткових заходів не здатне опрацьовувати обміни на швидкостях, які вимірюються одиницями Гбіт/с, не кажучи про операторів послуг, які оперують десятками, а іноді й сотнями Гбіт/с. Існуючі рішення це, як правило, вузькоспеціалізовані комплекси з обмеженою здатністю до оновлення та розширення пропускної здатності. В цій статті розглянуто механізми та навики, отримані під час реалізації гнучкого набору інструментів для обробки мережевих пакетів, який має здатність до розширення та може працювати на швидкостях до 10 Гбіт/с. В роботі даються чіткі вказівки, які компоненти системи мають бути модифіковані для досягнення вказаного результату.

1. Передумови створення систем класифікації трафіку

Необхідність проведення пакетної обробки на високих швидкостях виникає внаслідок зростання попиту на доступ до мережі Інтернет загалом, в тому числі й за допомогою бездротових мобільних пристроїв. Проведення моніторингу такої активності має чіткі переваги — приблизно можна оцінити напрямок розвитку окремої технології, щоб в майбутньому спрогнозувати необхідну ємність для розширення зовнішніх каналів зв'язку.

Застосування таких пристроїв, як NAT чи IDS робить їх центральними сервісними елементами пакетної мережі, оскільки за умовами їх режиму роботи весь трафік має проходити через них. Зі збільшенням середньої швидкості Інтернет-підключення для кінцевого користувача зростають вимоги до систем фільтрації небажаного мереженого трафіку. Проблема захисту граничного периметра мережі має велике значення для якості послуг, які надає оператор. Такі функції покладаються на фаерволи та системи глибокого аналізу пакетів (DPI) [1]. В той час, коли існує велика кількість рішень від відомих розробників апаратних мережених пристроїв, розробники відкритих систем стикаються із труднощами, пов'язаними із необхідністю проведення оптимізації в програмному забезпечення для досягнення достатньої швидкодії, або дефектами реалізації в апаратних схемах мережених адаптерів, які використовуються для аналізу мережевого трафіку. Деякі дослідники скаржаться на те, що зростання швидкості підключення не відповідає закону Мура, інші стверджують, що запропоновані швидкості здебільшого не використовуються глобально. Виходячи з цих особливостей, існує необхідність створення системи, яка б: 1) була спроможна опрацьовувати мережеві з'єднання на гігабітних швидкостях та 2) дозволяла проводити розширення з рос-

том пропускної здатності. Як вже було зауважено, зараз існують складнощі як з доступними апаратними, так і програмними реалізаціями. Існування закритих систем, які здатні вирішувати поставлені задачі зараз не гарантує можливості їх використання для майбутніх швидкостей. З іншої сторони, пропускної спроможності сучасних серверів та навіть персональних станцій в теорії більш ніж достатньо для обробки гігантських обсягів даних. Зокрема пропускна здатність шини PCI-x16 становить 32 Гбіт/с, що більш ніж достатньо для більшості додатків.

Важливою особливістю є іноді неможливість компонування рішень різних проблем з метою створення уніфікованого інструменту, наприклад, створення і використання BPF-фільтрів [2]. Враховуючи їх широке розповсюдження, можна зауважити такий аспект, як неможливість у випадку написання правил для NProbe [3], які можуть використовуватись з метою запису даних на диск для майбутнього відтворення подій чи подальшого аналізу. Поєднання вказаних програмних реалізацій наведено лише для демонстрації загального механізму — об'єднання незалежних інструментів для уніфікованої системи з глобальним відкритим інтерфейсом взаємодії. З розвитком технологій почало з'являтись багато інтелектуальних мережевих карт (зокрема, Intel IXP [4, 5]), які дозволяли переносити частину розрахунків з центрального процесора на мікропроцесор мережевої карти, але відсутність програмного забезпечення, яке б обробляло пакети з єдиним інтерфейсом керування сповільнювало стрімкий розвиток в цій галузі. В роботі буде показано, в яких саме напрямках доцільно проводити оптимізацію шляху обробки пакета для досягнення максимальної пропускної здатності та мінімізації часу обробки. В роботі розглянуто методи та підходи, що використовуються в системах класифікації пакетів, зокрема, як вже відомі, так і ті, що набувають популярності.

Метою дослідження є створення гнучкої системи пакетної класифікації, використання якої дозволило б застосовувати її для широкого спектру задач, і в окремому випадку, — для гарантування якості сервісу для абонентів бездротових мереж третього покоління. За основні метрики обрано пропускну здатність системи та простоту її інтеграції з існуючими комплексами аналізу та обробки.

Одним з поставлених завдань дослідження було проведення порівняльного аналізу систем обробки та класифікації мережевих пакетів з метою визначення найвдалішого методу, який буде доцільно застосовувати у системі аналізу. Вибір оптимального методу аналізу та відповідної програмної реалізації дозволить мінімізувати час обробки одиночного пакета та досягти вищих показників пропускної здатності. Створення системи класифікації мережевих потоків має важливе значення для виділення окремих потоків даних в пріоритетний клас обробки з метою гарантування якості обслуговування для взаємодії в мережі. Можливість проводити аналіз високошвидкісних потоків на доступному апаратному забезпеченні також дозволить відслідковувати аномальні сплески трафіку та виявляти їх джерела. Особливістю таких систем є їх відкритість, дешевизну та можливість самостійного оновлення. Створення такої системи планується в рамках подальшої роботи.

Під час розробки рекомендацій до впровадження увага має приділятись методам, які використовуються у комерційних розробках або у супутніх проектах, наприклад в FFPF [6] — система моніторингу високошвидкісних з'єднань, обробка пакетів не обмежена ядром операційної системи. Навпаки, розробники намагалися проводити розрахунки у мінімальній віддаленості від мережевого інтерфейсу, тобто використовуючи апаратні засоби мережевих карт. Проблеми, які було вирішено в розрізі покращення пропускної здатності, та отримані висновки мають важливе значення для розробників, які в подальшому планують проводити дослідження комплексів пакетної обробки. Проблематика проведення класифікації та методи покращення показників системи викладено у наступному розділі.

2. Інструменти побудови класифікаторів мережевих пакетів

В цьому розділі буде розглянуто основні засоби та особливості реалізації проаналізованих під час дослідження систем. Покращення характерних метрик кожного з підходів несе позитивну зміну в напрямку росту пропускної здатності чи часу обробки. Окрема увага приділяється мінімізації процедур копіювання пакетів для цілей обробки центральним процесором та уніфікації взаємодії між керувальним модулем та відповідним обробником.

2.1. Визначення належності до потоку

Задача пакетного фільтра — допомогти класифікатору обрати ті пакети, які відповідають деякому сформованому наперед відношенню з мінімальним системним навантаженням. Він має спрощувати обробку пакетів, абстрагуючись від складнощів реалізованого апаратного та програмного забезпечення. Підтримка повного набору додатків має відбуватись без огляду цілей, для яких

необхідна статистика пакетів, що надходять. Обмеження деякими типами пакетів не є гарною ідеєю. Системи, на яких фокусується увага в організації прикладного комплексу аналізу пакетного навантаження головною мірою засновані лише на бібліотеках PCAP, що не є достатнім в цьому відношенні.

Запропоноване поняття об'єднаного потоку вигідно використовувати, як абстракцію рівня ядра для пакетної обробки. На відміну від раніше відомих визначень потоку (TCP flow, Cisco NetFlow, IPFix), об'єднаний потік — це набір мережевих пакетів, що підпадають під правила, сформовані адміністратором. Можливість формулювання гнучкого алгоритму виділення пакета в той чи інший потік відрізняє систему аналізу від звичайного мережевого екрану.

Наприклад, такому групуванню можуть відповідати всі пакети, портом призначення яких є порт 80 протоколу TCP зі встановленим значенням SYN TCP — заголовку та мультимедійне навантаження з негарантованою доставкою за допомогою протоколу UDP і протоколом верхнього рівня RTP.

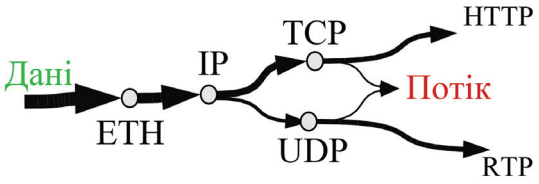

Рис. 1. Класифікація на основі потоку

Схематично класифікацію зображено на рис. 1.

Кожен потік даних після проходження функціональної обробки (наприклад, фільтрація, постановка в чергу, підрахунок) формує структуру, яка називається flowgraph — граф потоку. Якщо необхідно, в гілках графа може відбуватися модифікація пакета (функціонал приховування чи трансляції NAT). Треба зазначити, що поняття потоку досить часто використовується в галузі мережевої взаємодії. Системи обробки пакетів дозволяють проводити над пакетами низку маніпуляцій, групуючи їх в ланцюжки, але водночас мають можливість отримання додаткової інформації у відповідності до результату, отриманого на попередньому кроці [7]. Найяскравішими прикладами таких структур є Linux Netfilter/IPTables.

2.2. Ієрархія зв'язків між компонентами системи

Вивчення морфології системи починається з елементного складу. Він може бути: гомогенним (однотипні елементи); гетерогенним (різнотипні елементи); змішаним. Гетерогенні елементи спеціалізовані, вони економічні і можуть бути ефективними у вузькому діапазоні зовнішніх умов, але що важливо, вони швидко втрачають ефективність поза цього діапазону.

Для спрощення управління інформаційними потоками більшість підходів зосереджені на створенні абстракцій верхнього рівня (наприклад, Netfilter). Тим самим, це позбавляє дослідників можливості використовувати розширені можливості мережевих плат. Карти DAG виробництва Intel мають можливість проведення фільтрації на самій карті, на схожих підходах організовано більшість апаратного мережного обладнання (Juniper) та відкритих програмних маршрутизаторів (Vyatta). Ці функції стають недоступними у використанні абстракцій верхніх рівнів моделі OSI. Існує два підходи у вирішенні проблем апаратної несумісності: 1) мінімізація кількості пакетів, що пересилаються між шиною мережевої карти та оперативною пам'яттю, і 2) реалізація підтримки встановлення програмованих апаратних регістрів, мережеві плати з якими стають все популярнішими. Система є гетерогенною структурою зі зв'язками між різними рівнями.

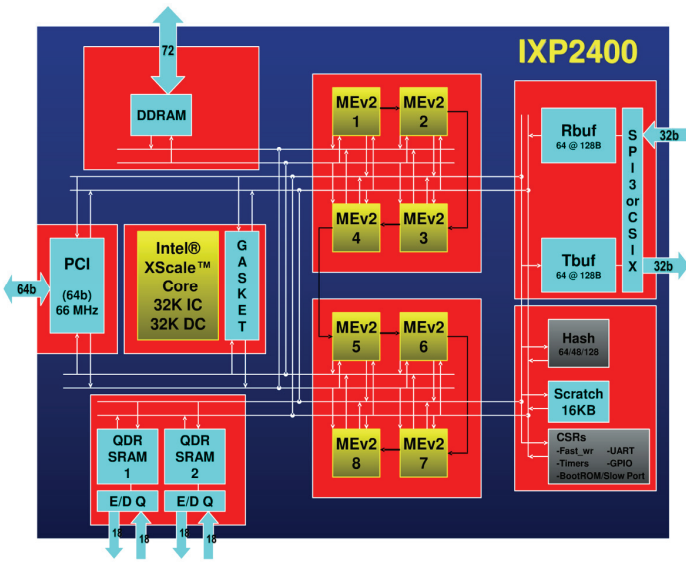

Рис. 2. Блок-схема реалізації плати IXP2400

Розглянемо детальніше структурну схему програмованої мережевої плати Intel IXP2400.

Внутрішня будова карти нагадує побудову ЦП. У ролі пристрою керування виступає SOC-чіп XScale. Кеш-пам'ять складається з трьох рівнів з відповідним функціоналом, об'ємом та швидкістю доступу. QDR SRAM — застосовується для збереження черг пакетів та таблиць маршрутизації, її обсяг складає одиниці Мбайт. Швидкість доступу 150 тактів. Обсяг DRAM складає близько де-

кількох Гбайт і в основному використовується для збереження корисного навантаження пакетів (сегменти даних 4 рівня моделі OSI). Архітектура карт дозволяє проводити їх стекування (об'єднання в один логічний пристрій) для досягнення необхідної пропускної здатності, що забезпечується шиною CSIX (Intel). Зіставлення співвідношень між графом потоку та гетерогенною ієрархією апаратно-програмного забезпечення не є простою задачею — це залежить від доступності апаратних ресурсів та можливостей поєднання потоків даних між собою, з метою визначення належності до одного потоку. Це стає можливим із застосуванням механізму функціонального класу з прив'язкою до окремого графу потоку. Наприклад, в BPF (FreeBSD) — це прив'язка до ядра і його простору пам'яті.

2.3. Інженерія різноманітних граничних умов

Як зауважено раніше, обробка ієрархії зв'язків може включати різні апаратні засоби. Наприклад, системі FFPF не відома загальна кількість потоків, але встановлені запити на переривання неявним чином можуть сигналізувати про пакетне навантаження. Запити виконуються рекурсивно, цей механізм усуває необхідність в централізації й спрощує алгоритм в цілому. Специфіка стосунків верхнього рівня потребує різного поводження з пакетом, тому оптимізація процесу копіювання між різними типами пам'яті є ключовою задачею. Розглянемо кожен з методів детальніше.

«Zero-Copy» — відомий метод покращення продуктивності високошвидкісної обробки, що прозоро відображує дані від мережевої карти до простору пам'яті, доступної користувачеві. Його переваги є найбільш помітними у додатках, які потребують доступу лише до деяких частин пакета, так як він відображує лише запитувані дані. Однак, якщо існує необхідність доступу до пакета більше ніж один раз, або коли необхідно дослідити навантаження в пакеті, — ефективнішим виявляється метод копіювання всього пакета за один прохід — «Copy-Once». Цей метод мінімізує запити на передачу через шину ЦП-пям'ять та оптимізує використання оптимізації DMA.

Метод «Copy-on-Demand» є деяким компромісом перших двох, його особливість полягає у попередній ідентифікації пакета за деякими критерієм, після чого для цього пакета відбувається обробка за методом «Copy-Once». Для практичного застосування цей метод має найважливіше значення, оскільки він дозволяє досягти найбільшого росту продуктивності роботи системи, так як тільки пакети, що відповідають умові, копіюються в локальну область пам'яті. В залежності від шаблонів доступу до пакета, образ пам'яті може бути відображений за допомогою метода «Zero-Copy», використовуючи процес перевизначення (remap) пам'яті через системний виклик *mmap*.

В результаті аналізу матеріалів з оптимізації мережевої взаємодії на високих швидкостях, використання механізму Polling є доцільним, так як кількість операцій та накладні витрати у разі використання переривання для кожного пакета на високих швидкостях є значними факторами, що обмежують продуктивність системи. Суть механізму Polling полягає у налаштуванні періодичного опитування мережевої карти ЦП сервера для отримання декількох пакетів, які знаходяться в буфері мережевої карти. Таким чином, обробка пакетів відбувається за детермінований часовий проміжок, а не подією надходження пакета. На практиці цей процес організовано за допомогою використання додаткового забезпечення NAPI для LINUX-ядра. Порівняння кожного з механізмів для пакетів однокового розміру (300 байт) наведено на рис. 3.

Тобто, з досягненням швидкості надходження(відправлення) ~100 тис. пакетів/с до деякого граничного рівня, обробка пакетів за перериваннями ЦП стає неефективною, що доведено подальшою інтеграцією метода NAPI в ядро операційної системи для досягнення більшої швидкості в роботі за високих значень pps (пакетів/с).

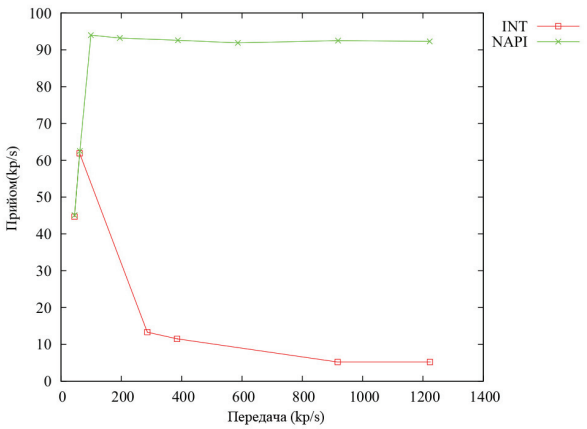

Рис. 3. Ефективність механізму Polling, FreeBSD (NAPI, Linux)

2.4. Мінімізація процесів копіювання та зміни контексту

Метод «Zero-Copy» у більшості випадків прикладного застосування є менш продуктивним, ніж «Copy-Once», зокрема для високошвидкісних з'єднань. Часті переключення контексту обробки пакета з рівня ядра (kernel-space) до додатку ко-

ристувача (user-space) не найкращим чином відображаються на кінцевій швидкодії системи. Більшість цих процедур можуть бути оптимізовані під виконання на рівні ядра, альтернативним рішенням може бути копіювання пакета мережевою картою у оперативну пам'ять напряму так, що ядро ніколи не дізнається про цей пакет [8]. Ця процедура може бути не завжди ефективною, але в найгіршому випадку вона позбавляє систему від частих переключень контексту.

2.5. Організація апаратної нейтральності

Необхідність підтримки комплексних архітектур та різноманіття програмного забезпечення ускладнює процес обробки пакетів та підтримку різних мов програмування для доступу до API. Розглянемо бібліотеку PCAP для BPF, яка є найпопулярнішою серед систем моніторингу. Фактором, що зумовив широке розповсюдження є відкритість та простота використання API. Але у нього є тенденція непродуктивного використання ресурсів через інтерпретовану природу його коду. Також відсутність переходів назад та збереження попереднього стану обмежують використання цього інструменту. Написання додаткових функцій, таких, як розпізнавання по шаблону було реалізовано поверх давно сформованої конструкції, а не всередині, що має недоліки в плані раніше згадуваного перемикання контекстів. На сьогодні не існує програмного інструменту, який би дозволяв проводити будь-які маніпуляції з мережевими IP-пакетами, саме тому необхідно використовувати компонування різних застосувань.

Netgraph — модульна мережева підсистема ядра системи FreeBSD, заснована на принципі графів. У netgraph будується граф з вузлів різних типів, вузол кожного типу має деяку кількість входів/виходів (англ. hooks). Вузол netgraph дозволяє виробляти певні дії над пакетом, що проходить через нього. Деякі netgraph вузли надають підтримку різних протоколів, інкапсуляцій, таких, як Ethernet, PPPoE та інших, вони служать для зв'язки модулів і сортування/маршрутизації між вузлами netgraph, наприклад BPF. Деякі дослідники зацікавленні в отриманні відбитків ідентичних взаємодій в мережі для цілей виявлення аномалій та прогнозування росту пропускної спроможності [9]. Рух в напрямку розширення можливостей стандартного функціоналу API PCAP було представлено в реалізації MAPI (дослідницький проект SCAMPI [10]).

2.6. Підтримка комплексних графів обробки

В сучасних комплексах аналізу мережевих пакетів функції обробки, наприклад фільтри, оголошуються, як вирази, або як набір скомпільованих правил, які застосовуються до потоку даних та дозволяють проводити перенаправлення виводу одного каналу на вхід іншого (концепція UNIX-PIPE). Але представлення мережевих протоколів зручніше робити у вигляді дерева, оскільки в ідею передачі по мережі закладено принцип інкапсуляції. Тому, запис правил у вигляді виразу є ненайзручнішим, оскільки застосування його повторно для дещо зміненого параметра важко модифікувати для апаратного програмування регістрів. Логічно встановити лічильники одразу після де-інкапсуляції протоколів TCP та UDP. Тобто дві гілки будуть сходитися в один функціональний потік.

Інший підхід, використаний у популярних обробниках пакетних даних, як OpenBSD pf або LINUX IPTables [11] — представлення фільтрів, як набір регулярних правил. В таких системах пакети опрацьовуються за принципом FIFO черг. Елементи черг проходять почергово через всі впорядковані набори правил, доки не відбувається збіг хоча б в одному з них. Після цього відбувається спрацювання тригера та подальша обробка пакета. Стандартний набір процедур включає: відкидання (drop), перенаправлення (forward), погодження (accept) та ін. Очевидним недоліком є залежність часу спрацювання від довжини списку правил.

2.7. Розділення площини управління та передачі даних

Створення комплексної ієрархії обробки на розподілених апаратних засобах є складним процесом з таких причин. По перше, з точки зору управління такою системою. По друге, процес передачі пакета від одного модуля до іншого буде вносити затримку в обробці пакета, яка є не бажаною. Тоді, як перше обмеження є умовним, боротьба із другим приводить до необхідності відділення шляху проходження пакета для аналізу.

Процес виділення оперативної пам'яті під потреби збереження пакетів доцільно контролювати централізовано. Структури пам'яті мають оголошуватись статично або виділятись динамічно з напередвизначених пулів системних ресурсів [12].

Розглянуті методи оптимізації можуть бути застосовані під час побудови комплексних систем

аналізу мережевого трафіку. Використання на практиці розглянутих принципів може бути узагальнене у вигляді табл. Під характерною оптимізацією мається на увазі схильність застосування того чи іншого методу, виходячи з такого показника, як мінімальній час обробки одиничного пакета, показники {1—7} табл. відповідає саме цій метриці, де {1} є найкращим результатом. Кожен з розглянутих методів має переваги, що відмічається, як «+»; наявність «++» показує суттєве підвищення метрик відповідної програмної оптимізації.

Реалізація	Характерна оптимізація принципу побудови						
Метод	1	2	3	4	5	6	7
1. BPF+	+	—	—	—	—	+	—
2. pf	+	—	—	—	+	+	—
3. Netfilter	++	+	+	+	+	+	—
4. MAPI	+	+	+	+	++	+	+
5. VR (Vyatta)	++	++	+	+	—	+	+
6. FFPF	+	+	++	++	++	—	—

Механізми, використання яких має місце в пакетному обробнику MPF(MAPI), є розширенням віртуальної машини BPF+ шляхом додавання розширених інструкцій обробки потоків даних через оптимізацію внутрішнього представлення фільтрів, що співпадають за декількома ознаками, наприклад, довжиною префіксу чи формою виразу.

Частина реалізацій (наприклад, 1—4) використовує розглянуті в статті оптимізації; окремі методи (6) вже не використовуються через їх архітектурну обмеженість та складності в реалізації.

Висновки

У ході дослідження обґрунтовано доцільність та розглянуто напрямки проведення оптимізації пакетної фільтрації для глибокої обробки мережевих пакетів з метою досягнення максимальної пропускної здатності. Розглянуто підходи, охарактеризовано раніше відомі та сучасні новітні методи, що використовуються в системах класифікації пакетів.

В рамках проведення дослідження було встановлено:

1. Програмна реалізація системи Netfilter (системи сімейства Linux) на базі методу «Copy-on-Demand» має найвдалішу архітектурну реалізацію. Завдяки використанню механізмів «гачків», маніпуляції з пакетом можливі на будь якому рівні під час проходження його через систему. Взаємодія з апаратними можливостями мережевої карти проявляється у можливості проведення змін у заголовках багато разів без необхідності повторного завантаження пакета з буферного елементу.

2. Реалізація в підсистемі Netgraph (операційна система FreeBSD) побудована за схожим підходом, але використовує поняття «вузлів» та «зв'язків». Специфічна обробка пакета відбувається в окремому модулі, кількість послідовних сполучень модулів необмежена, тому система є гнучкішою. Однак, використання спеціалізованого синтаксису опису взаємодії ускладнює створення комплексних застосувань з великою кількістю «зв'язків».

3. Вибір остаточної програмної частини значною мірою залежить від доступної адміністратору системи апаратної бази та кола задач, для яких планується використання комплексу аналізу. Розглянуті механізми можуть застосовуватись у побудові масштабованих систем мережевої класифікації пакетів.

4. Надані технічні рекомендації ґрунтуються на побудові засобів пакетної фільтрації з акцентом на використання відкритого програмного забезпечення та апаратних можливостей мережевих адаптерів.

В подальшому може бути розглянута процедура проведення інтеграції з системою Snort для можливості роботи системи в режимі IDS та з набором утиліт класифікатора OpenDPI (набір бібліотек для класифікації мережевих пакетів в режимі реального часу).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. OpenDPI — carrier-grade DPI bandwidth management solutions, 2011 / [Електронний ресурс]. — Режим доступу : http://www.ipoque.com/en/news-events/press-center/press-releases/2010/ipoque-open-kick-off-ipoque%E2%80%99s-open-source-dpi.

2. Hummel S. Network Performance and Optimization Guide: The Essential Network Performance / S. Hummel // CreateSpace Independent Publishing. — Toronto, October 26, 2013. — P. 111—113.

3. Network Processor Design / [H. Haldun, A. Mark, Z. Peter, P. Crowley] // Issues and Practices, Volume. — October 18, 2002. — P. 21—25.

4. Burin W. A General Approach for Efficiently Accelerating Software-based Dynamic Filters / W. Burin // Proc. 19th Network and Distributed System Security Symposium (NDSS). — San Diego, CA, USA, February 2012. — P. 37—38.
5. Jeremy C. The Openbsd Pf Packet Filter / C. Jeremy // Reed Media Services — New–York, August 18, 2006. — P. 12—16.
6. O'Neill T. Use Packet Filtering to Monitor 10 Gigabit Networks with Gigabit Tools / T. O'Neill // Alastair Hartrup, July 28, 2008. — P. 55—59.
7. Serpanos D. Architecture of Network Systems Computer / D. Serpanos, W. Tilman // Security Magazine, February 2, 2011. — P. 44—45.
8. Varghese G. Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices / G. Varghese // Security Magazine, December 29, 2004. — P. 91—96.
9. Yang B. Packet classification algorithms: From theory to practice, Proc. / B. Yang, Y. Xue, J. Li // INFOCOM, 2009. — P. 11—15.
10. KVM : The linux virtual machine monitor / [A. Kivity, Y. Kamay, D. Laor, U. Lublin] // Linux Symposium, Magazine, 2007. — P. 56—61.
11. Martin L. Programming with libpcap — sniffing the network from our own application, Hakin / L. Martin // Computer Security Magazine, 2008. — P. 66—71.
12. Moore A. Architecture of a network monitor / A. Moore, J. Hall // PAM'03, 2003. — P. 9—11.

Рекомендована кафедрою обчислювальної техніки ВНТУ

Стаття надійшла до редакції 6.06.2014

Дєєв Костянтин Сергійович — аспірант кафедри комп'ютерної інженерії, e-mail: kostic@dinet.co;
Бойко Юрій Володимирович — канд. фіз.-мат. наук, доцент кафедри комп'ютерної інженерії, e-mail: boyko@univ.kiev.ua.
Київський національний університет ім. Тараса Шевченка, Київ

K. S. Dieev[1]
Yu. V. Boyko[1]

Analysis of methods and means of realization of packet filtering for deep analyzing of network packets

[1]Taras Shevchenko National University of Kyiv

The samples of usage of packet filters for tasks of prioritizing certain types of network traffic in order to maintain service layer arguments on high level are described in the paper. The possibility of application of open software for implementation of the same aims is estimated.

Keywords: network monitoring, traffic analysis, packet classifying, intrusion detection.

Dieev Kostiantyn S. — Post-Graduate Student of the Chair of Computer Engineering, e-mail: kostic@dinet.co;
Boyko Yurii V. – Cand. Sc. (Ph.-Math.), Assistant Professor of the Chair of Computer Engineering, e-mail: boyko@univ.kiev.ua

К. С. Деев[1]
Ю. В. Бойко[1]

Анализ методов и средств реализации пакетной фильтрации для глубокого анализа сетевых пакетов

[1]Киевский национальный университет имени Тараса Шевченко

Приведены примеры использования пакетных фильтров для задач приоритезации отдельных типов сетевого трафика с целью поддержания высоких показателей качества обслуживания. Оценена возможность применения открытого программного обеспечения для выполнения этих задач.

Ключевые слова: сетевой мониторинг, анализ трафика, классификация пакетов, обнаружение вторжений.

Деев Константин Сергеевич — аспирант кафедры компьютерной инженерии, e-mail: kostic@dinet.co;
Бойко Юрий Владимирович — канд. физ.-мат. наук, доцент кафедры компьютерной инженерии, e-mail: boyko@univ.kiev.ua