

Гніденко М.П., к.т.н.; Прокопов С.В., к.т.н.;  
Кароян Р.Р.; Карпик К.О.; Петренко В.В.

## ЗАБЕЗПЕЧЕННЯ QoS ДЛЯ ВИСОКОПРОДУКТИВНИХ РОЗПОДІЛЕНИХ ПРОГРАМ У ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖАХ (SDNs)

**Hnidenko N.P., Prokopov S.V., Karoian R.R., Karpik K.O., Petrenko I.O. Providing QoS to High Performance Distributed Applications in Software-Defined Networks (SDNs).** Nowadays, there is an ever-increasing number of applications that have strict quality of service (QoS) requirements. The ability to define QoS requirements in traditional networks is limited due to the complex control of these environments. Modification of the existing infrastructure has high administrative costs, since the subsystem of each service component must have its resources managed by a centralized organization to guarantee users the level of QoS that each of them requires.

The specification of quality of service (QoS) requirements in traditional networks is limited by the high administrative cost of these environments. Nevertheless, newer network paradigms, as Software-Defined Networks (SDNs), simplify and relaxes the management of networks. In this sense, SDN can provide a simple/effective way to develop QoS provisioning. This approach allows the specification of classes of service and also negotiates the QoS requirements between applications and the SDN network controller. The SDN controller, in turn, monitors the network and adjusts its performance through resource reservation and traffic prioritization. Have developed a QoS provision architecture for high performance distributed applications exploiting the capabilities of SDN. In this architecture, simple software components enable a process of message exchanging that includes reservation request and reservation confirmation. This procedure allows a network controller to configure QoS queues to individual packet flows. The developed a proof-of-concept of proposal and experimental results show that the additional routines present low overhead, whereas – for a given test application — observes a reduction of up to 47% in transfer times. Despite the specific scenario evaluated, results indicate our solution feasibility. Future works include the development of renegotiation and adaptation functions

**Keywords:** Software-defined networks (SDNs), QoS, OpenFlow.

**Гніденко М.П., Прокопов С.В., Кароян Р.Р., Карпик К.О., Петренко В.В. Забезпечення QoS для високопродуктивних розподілених програм у програмно-визначених мережах (SDNs).** Специфікація вимог щодо якості обслуговування (QoS) у традиційних мережах обмежена високою адміністративною вартістю цих середовищ. Тим не менш, новітні мережеві парадигми, такі як програмно-визначені мережі (SDNs), спрощують і полегшують керування мережами. У цьому сенсі SDN може забезпечити простий та ефективний спосіб розробки забезпечення QoS. Цей підхід дозволяє специфікувати класи обслуговування, а також погоджувати вимоги щодо QoS між програмами та мережевим контролером SDN. Контролер SDN, у свою чергу, контролює мережу та регулює її продуктивність за допомогою резервування ресурсів і пріоритизації трафіку.

**Ключові слова:** Software-defined networks (SDNs), QoS, OpenFlow.

### Вступ

**Постановка задачі.** Зараз спостерігається постійно зростаючу кількість програм, які пред'являють суворі вимоги до якості обслуговування (QoS). Можливості визначення вимог QoS у традиційних мережах обмежені через складний контроль цих середовищ. Модифікація існуючої інфраструктури має високі адміністративні витрати, оскільки підсистема кожного компонента служби повинна мати свої ресурси, керовані централізованою організацією, щоб гарантувати користувачам рівень QoS, який вимагає кожен із них. Навпаки, програмно-визначені мережі (SDN) являють собою нову парадигму, яка полегшує створення та впровадження нових абстракцій у мережі, спрощуючи керування та сприяючи її розвитку.

Однією з фундаментальних характеристик концепції SDN є відокремлення площин даних і керування мережевими пристроями (комутаторами та маршрутизаторами). Мережевий інтелект передається логічно централізованому елементу контролера, який через захищені протоколи зв'язку, наприклад OpenFlow, керує пристроями пересилання даних.

Таким чином, модель SDN забезпечує глобальний огляд і підвищений рівень програмованості мережі. Завдяки цим гнучким можливостям мережевий контролер може опціонально забезпечувати прості, але ефективні механізми надання QoS. Наприклад, контролер може виконувати формування трафіку, гарантуючи чутливій програмі максимальну затримку, джиттер або пропускну здатність.

У цій статті буде представлена архітектура надання QoS, використовуючи можливості SDN. Запропонована модель забезпечує ефективну комунікаційну платформу, резервуючи ресурси та пріоритезуючи трафік даних. Підхід полягає у забезпеченні програмних компонентів, за допомогою яких розподілені програми виконують виклики функцій. Вони узгоджують параметри конкретних вимог QoS між компонентами архітектури (об'єктами зв'язку та контролерами SDN). Класи обслуговування та їхні специфікації трафіку попередньо визначені в структурах даних, створених адміністратором мережі та доступними для деяких елементів середовища. Процес обміну інформацією про запит на бронювання та підтвердження бронювання заснований на протоколі RSVP (Resource ReSerVation Protocol), який використовується в архітектурі IntServ (Integrated Services). Ця процедура контролюється контролером SDN, який остаточно модифікує мережу, розподіляючи черги QoS для окремих потоків пакетів на інтерфейсах пристроїв пересилання.

Щоб продемонструвати здійсненність цієї пропозиції, були розроблені компоненти архітектури SDN для конкретного сценарію (тобто передачі файлів даних). Було налаштовано середовище SDN в емуляторі Mininet1, використовуючи контролер POX2 SDN. Оцінки показали, що пропозиція має незначні накладні витрати. Крім того цей підхід покращив продуктивність мережі.

### **Підходи до забезпечення QoS на основі архітектури SDN.**

Набір вимог до QoS є функцією, яка різниться в залежності від послуги і, як правило, залежить від типу програми. Можна перерахувати два основні принципи, пов'язані із забезпечення QoS: (i) специфікація вимог користувача; та (ii) забезпечення механізмів, які забезпечують вимоги користувача. Вимоги, як правило, визначаються з точки зору максимальної затримки, статистичних змін затримки (тремтіння), пропускну здатності, частоти помилок і частоти втрат. Забезпечення механізмів, які гарантують такі вимоги, означає, що зазначені параметри гарантуються наскрізно для додатків під час їх виконання.

На основі цих принципів можна визначити чотири загальні фази забезпечення QoS: (i) запит на послугу; (ii) укладення угод про надання послуг; (iii) обслуговування угод про надання послуг; та (iv) закриття угод про надання послуг. На етапі запиту на послугу можуть бути використані статичні механізми, які накладають дорогий адміністративний тягар, або динамічні, такі як ті, що використовують протоколи сигналізації. Після того, як запит зроблено, між програмою та мережею встановлюється неявна угода. Потім мережа забезпечує транспортування потоків програми, доки програма підтримує генерацію трафіку відповідно до того, що було зазначено в запиті. Ця угода підтримується відповідними механізмами до її закриття, яке також може бути зроблено статичним або динамічним способом, коли звільняються зарезервовані ресурси.

Враховуючи ці принципи та етапи, необхідними функціональними можливостями для загальної структури забезпечення QoS є наступні: (i) параметризація послуг визначає параметри та категорії послуг; (ii) спільне використання ресурсів стосується механізмів розподілу, класифікації та планування; (iii) оркестровка ресурсів враховує функції узгодження, контролю допуску та налаштування; і (iv) адаптація послуг змінює поведінку вищезазначених механізмів і функцій.

Конфігурація функцій і механізмів забезпечення QoS у кожному елементі мережі в звичайній мережі передачі даних накладає високі адміністративні витрати. SDN, навпаки, спрощує управління цими мережами. Коротко кажучи, SDN відноситься до мережевої архітектури, яка базується на чотирьох стовпах: (i) відокремлення площини даних і управління; (ii) пересилання рішень на основі потоків; (iii) передача логіки керування до

логічно централізованого зовнішнього контролера; та (iv) мережеве програмування за допомогою програмних додатків, що працюють у верхній частині моделі. На рис.1 зображено типову архітектуру SDN.

Інтерфейс між контролером SDN і пристроями мережевої інфраструктури (південний API) забезпечується захищеними протоколами, такими як OpenFlow. На відміну від північного API, який ще не стандартизований, OpenFlow вважається стандартом де-факто. Таким чином, у типовому середовищі SDN елементи пересилання даних (комутатори та маршрутизатори) підтримують OpenFlow, тобто ними можна керувати контролером, сумісним з OpenFlow.

API OpenFlow, починаючи з версії 1.0, надає власні функції, які дозволяють розробляти механізми забезпечення QoS. Можна легко реалізувати розподіл пакетів, класифікацію та планування. Деталізація обробки окремих потоків і глобальне уявлення про SDN також дозволяють забезпечити узгодження QoS, контроль доступу та механізми налаштування. Ці можливості використовуються в рішеннях QoS, які використовують перемаршрутизацію пакетів і балансування трафіку чутливих потоків.

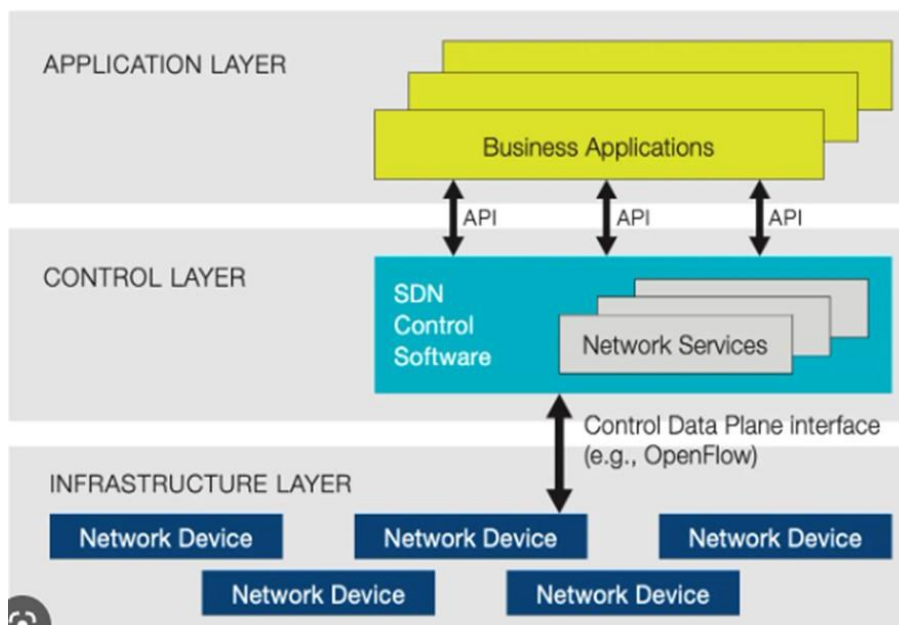


Рис. 1. Типова архітектуру SDN

Однак ці методи не завжди здійсненні. У багатьох випадках політика маршрутизації, яка застосована до мережі, перешкоджає встановленню альтернативних шляхів. В інших випадках доступних маршрутів може не бути, або ці маршрути можуть бути просто перевантаженими. У цьому сенсі методи формування трафіку шляхом наскрізного резервування смуги пропускання та обмеження швидкості є простими та ефективними рішеннями надання QoS, які здатні гарантувати вимоги QoS додатків.

#### **Пропозиції щодо нової архітектури забезпечення QoS.**

Запропонована архітектура враховує принципи, етапи та функціональні можливості, розглянуті вище, які безпосередньо керують проектуванням компонентів і розробкою прототипів, що використовуються в оцінці. Тим не менш, важливо підкреслити, що розробка цієї архітектури обмежується забезпеченням рівнів якості в підсистемі комунікаційної мережі, не турбуючись про забезпечення QoS на кінцевих хостах (буфери введення/виведення, процеси ОС, стек протоколів, серед інші), як запропоновано. На Рис.2 показано високорівневе представлення запропонованої архітектури.

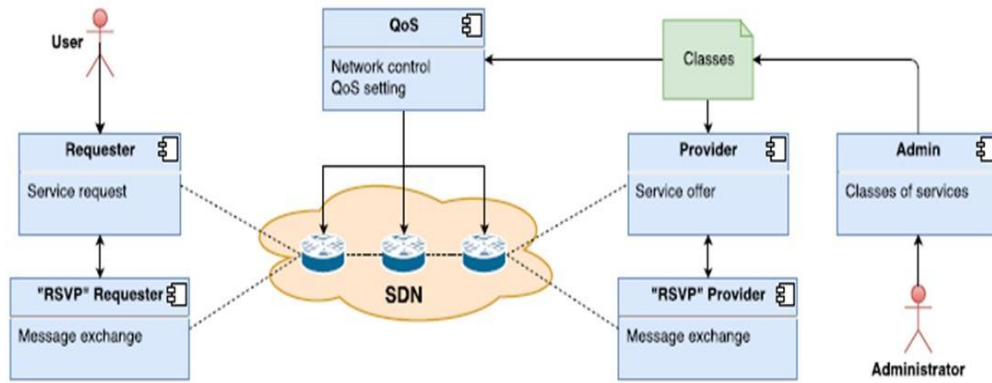


Рис. 2. Нова архітектури забезпечення QoS

Архітектура складається з модулів, пов'язаних із комунікуючими об'єктами (запитувач і постачальник, які представляють самі розподілені програми), обмін повідомленнями (запитувач «RSVP» і постачальник «RSVP»), контролер SDN (QoS) і адміністрування мережі (адміністратор). разом зі структурою даних (класи). Модуль запитувача відповідає за запит послуг від розподіленої програми до модуля постачальника. Останній, окрім зберігання даних, запитуваних першим, виконує функцію перевірки можливості надання послуг із необхідними гарантіями якості. Для цього постачальник перевіряє структуру даних, яка містить ідентифікацію категорії послуги (наприклад, VoIP, відео, найкращі зусилля) та її параметри (наприклад, середня та максимальна пропускна здатність). Модуль адміністрування мережі, доступний для управління мережею, має завдання генерувати цю базу даних класів обслуговування, тобто реалізує функціональні можливості параметризації обслуговування. Він також виконує початкову конфігурацію мережевих політик QoS, оскільки OpenFlow не має цієї конкретної можливості. Модулі обміну повідомленнями відповідають за процедури запиту та підтвердження резервування ресурсів. Модуль контролера відповідає за контроль доступу резервування та налаштування в мережі для забезпечення QoS.

У запропонованій архітектурі розподілені програми мінімально модифіковані. Спочатку, після запиту до структури даних, модуль провайдера інформує модуль запитувача про очікуване надання QoS. Таким чином, обидва своєчасно ініціюють звернення до відповідних функцій модулів запиту на бронювання та підтвердження бронювання. Модуль провайдера також взаємодіє з відповідним модулем «RSVP», вказуючи категорію послуг, пов'язану з додатком. Завдяки цим змінам створюється простий і ефективний інтерфейс, завдяки якому розподілені програми можуть запитувати певний QoS у мережі, в той же час, що це вимагає невеликого втручання розробників цих програм, для цього достатньо реалізувати запити до класів і виклики функцій модулів обміну повідомленнями.

Процес запиту резервування та підтвердження резервування інспірований протоколом RSVP, як показано на Рис. 3.

Процедура контролюється контролером SDN, який перевіряє пакети, що містять відповідні повідомлення, і реалізує механізми надання QoS. Спочатку (i) процес на стороні постачальника надсилає повідомлення «PATH» процесу на стороні запитувача разом із числовим ідентифікатором, що відповідає класу послуги, що надається. Під час обробки кожного пакета, що містить це повідомлення, підпрограми модуля контролера встановлюють правила пересилання на комутатори/маршрутизатори, а потім встановлюють наскрізний шлях. Потім (ii) процес запитувача, отримавши повідомлення «PATH», розпізнає ідентифікатор класу та генерує повідомлення «RESV» (запит на резервування), вставляючи відносний номер до ідентифікатора в поле ToS заголовка пакета. Під час аналізу цього пакету в кожному елементі пересилання даних на маршруті між взаємодіючими об'єктами підпрограми QoS модуля контролера виконують контроль допуску резервувань, обчислюючи в кожному порту з'єднання доступність пропускної здатності, необхідної для цього класу

обслуговування, який отримано шляхом вивчення поля ToS. Надходження повідомлення «RSVP» на стороні постачальника означає, що бронювання можливе протягом усього маршруту. Таким чином (iii) процес постачальника генерує новий пакет із повідомленням про підтвердження бронювання «RESVCONF» з ідентичним позначенням поля ToS. Таким чином, коли модуль контролера аналізує такий пакет, він застосовує політики QoS на інтерфейсах пристроїв, через які мають передаватися чутливі потоки, а також налаштовує їхні черги з правилами, що відповідають цим потокам (через директиву ENQUEUE OpenFlow). Таким чином розподіл пакетів, класифікація та планування гарантуються. Нарешті, (iv) надходження повідомлення «RESVCONF» до процесу запитувача підтверджує резервування ресурсу на шляху, що дозволяє відправити повідомлення «FIN», закриваючи процедуру. З цього моменту постачальник починає надсилати потік даних, який матиме гарантований QoS.

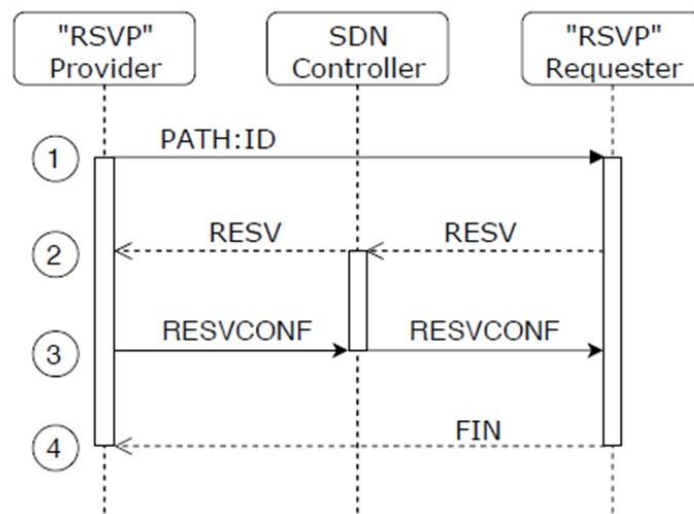


Рис. 3. Діаграма послідовності повідомлень про резервування

У логіці, описаній вище, не показано винятки, такі як недоступність пропускну здатності під час резервування. Винятки обробляються повідомленнями про помилки та вичерпанням резерву/шляху, як-от наявні в RSVP. Ми також пропускаємо більш складні функції надання QoS, такі як механізми налаштування та адаптація послуг.

#### Оцінка ефективності нової архітектури забезпечення QoS.

Щоб оцінити здійсненність пропозиції, був розроблений прототип архітектури та її компонентів. Нижче наведено деталі оцінки ефективності нової архітектури.

На Рис.4 представлений сценарій оцінки, розглянутий у цій роботі. Він базується на загальному середовищі SDN, пропонуючи послугу передачі даних. Цей спрощений сценарій служить будівельним блоком для більш складних сценаріїв. Він складається з двох хостів (h1 і h2), двох серверів (srv1 і srv2), трьох маршрутизаторів (r1 до r3) і мережевого контролера (c0). Маршрутизатори r1 і r2 діють як шлюзи для двох різних мереж.

Цей сценарій оцінювання складається зі скороченого, але репрезентативного мережевого середовища даних, яке, згідно з попередніми аналізами, охоплює всі важливі аспекти нашої оцінки. У цьому сценарії програми вимагають певного рівня QoS від постачальника послуг, який забезпечує необхідний QoS під контролем адміністративної сутності. Функції постачальника послуг і запитувача послуг можуть виконуватися будь-яким із хостів/серверів так само, як і в архітектурі без QoS. Топологія singlepath між станціями представляє мережу, де методи надання QoS шляхом перемаршрутизації або балансування трафіку не застосовуються. Кількість маршрутизаторів виправдовується середньою відстанню в стрибках між будь-якими хостами/серверами в менших мережах, таких як ті, що

знаходяться в університетах або невеликих центрах обробки даних, що робить сценарій реалістичним.

Мережне середовище, описане вище, було реалізовано в емуляторі Mininet. Хости та сервери були включені з однаковими характеристиками, хоча вони представлені графічно різними способами. Подібним чином маршрутизатори були додані до топології як комутатори під керуванням Open vSwitch 2.0.2. POX з використанням OpenFlow 1.0 використовувався як контролер SDN. Незважаючи на наявність нових версій OpenFlow, версія 1.0 все ще є найпоширенішою. Все середовище було змонтовано на 64-розрядному процесорі Intel та операційною системою Ubuntu 14.04. Використані обчислювальні ресурси були достатніми для оцінки та не вплинули на результати, оскільки споживання пам'яті та процесора під час усіх тестів залишалося приблизно на рівні 30% та 25% відповідно.

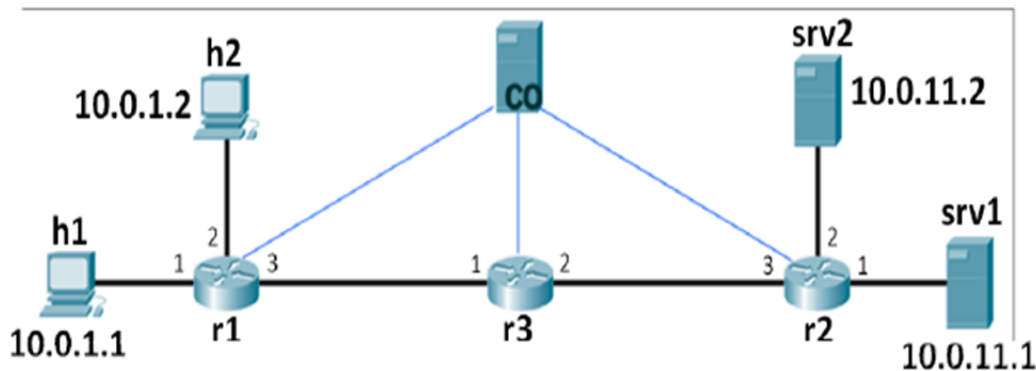


Рис. 4. Сценарій оцінки ефективності нової архітектури забезпечення QoS

Була підготовлена наступна тестова модель для відтворення середовища, здатного висвітлити вплив надання QoS на одночасні потоки: (i) сервер srv1 відповідає за передачу даних; (ii) хост h1 (зонд) відіграє роль запитувача, виконуючи запити даних до srv1; (iii) між srv2 і h2 генерується односпрямований постійний трафік; і (iv) пропускна здатність каналів обмежена 1 Гбіт/с. У цьому контексті трафік між srv2 і h2 імітує невідстримувані потік даних QoS. Цей потік генерується інструментом iPerf4 із завантаженням за замовчуванням (TCP із максимальною швидкістю). У свою чергу, потік даних між srv1 і h1 розглядається як конфіденційний. Отже, із запиту h1 трафік відповіді srv1 класифікується відповідно до попередньо встановленого рівня якості. Таким чином, під час передачі між srv1 і h1 обидва потоки стають одночасними на маршруті r2-r3-r1, так що пропускна здатність трафіку від srv2 зменшується до мінімального рівня, тоді як пропускна здатність чутливого потоку гарантується. У симуляції без QoS, як буде видно пізніше, обидва потоки спільно використовують максимальну пропускну здатність каналів.

Для реалізації тестової моделі було розроблено дві програми (запитувач і провайдер) для передачі мережових даних по протоколах TCP/IP. Використання TCP виправдано тим фактом, що це переважний транспортний протокол Інтернету. Зараз навіть програми, чутливі до затримки/третиння та стійкі до втрат, такі як поточні потокові мультимедійні рішення, розгортаються на цьому протоколі, особливо на адаптивних поточкових платформах, що надаються такими провайдерами, як Netflix. Також реалізовано програмні компоненти, які дозволяють специфікувати параметри та категорії послуг у мережі. Незважаючи на те, що архітектура надає велику свободу у визначенні класів QoS, у контексті тестової моделі були потрібні лише два рівні, які змінювалися в залежності від гарантованої пропускної здатності. Їх можна порівняти, наприклад, із типовими рівнями контрольованого навантаження та гарантованого обслуговування. Для обміну повідомленнями параметрів якості між комунікуючими об'єктами також були розроблені відповідні модулі. Нарешті, модуль пересилання пакетів POX було модифіковано з включенням процедур забезпечення QoS. Все програмування виконано на мові Python.

Для розробки моделювання було визначено дві змінні відповіді, обидві виміряні з точки зору запитувача: (i) накладні витрати, додані до часу передачі, викликані обробкою підпрограм надання QoS; та (ii) загальний час передачі даних. Для симуляції передачі ми використовували відеофайли, розмір яких був встановлений як параметр експериментів, у цьому випадку змінюючись у трьох рівнях розміру файлу: (i) малий (44 МБ); (ii) середній (128 МБ); і (iii) великий (435 МБ). Ці значення відповідають розмірам відносно трьох різних роздільних здатностей одного відео. З цього планування було виконано три тестові сесії, де для кожної сесії використовувався лише один розмір файлу. У кожному сеансі було виконано три раунди запитів, у яких мережа поводить за трьох різних умов: (1) контролер SDN з оригінальним модулем (без надання QoS); (2) Контролер SDN з модифікованим модулем і 40% резервуванням пропускної здатності для чутливого потоку; і (3) контролер SDN з модифікованим модулем і 60% резервуванням пропускної здатності для чутливого потоку. У тестах із наданням QoS (випадки 2 і 3) трафік без QoS було встановлено на рівні 10% пропускної здатності, так що потік даних між *srv2* і *h2* потрапив у цю категорію. Щоб визначити вплив експериментальної помилки, у кожному раунді робили 30 запитів того самого файлу за допомогою розроблених додатків передачі з інтервалом 120 с між кожним запитом..

Експериментальні результати показують, що пропозиція має низькі накладні витрати. Дійсно, середня накладна затримка становить від 1,05 до 1,06 с для 95% рівня довіри. На Рис.5 показано CDF (Cumulative Distribution Function) цих накладних витрат.

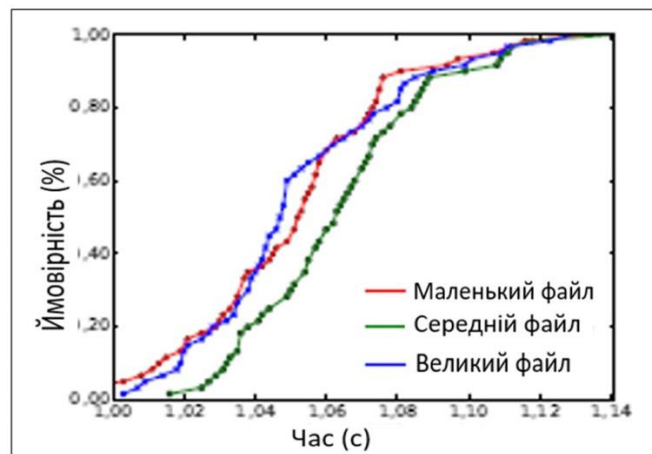


Рис. 5. Розподіл накладних витрат для кожного розміру файлу.

Окрім самого процесу обміну повідомленнями на етапі запиту на резервування, цей показник пов'язаний із кількістю мережевих переходів. Іншими словами, чим більша кількість переходів, тим вищими будуть накладні витрати, оскільки процедури резервування, що виконуються контролером, вимагають конфігурації кожного елемента пересилання мережі. Природно, що для сценарію симуляції, який використовується, це середнє значення має великий вплив, головним чином на коротші передачі даних, і мінімізується в ситуаціях, коли передачі займають більше часу. У цьому відношенні можна оцінити, що в середовищах, де є великі файли або великі навантаження, такі як потокове відео або масова передача даних (чия тривалість може досягати десятків хвилин), накладні витрати компенсуються гарантією QoS, наданою для користувачів.

Як показано на Рис.6, спостерігається значне скорочення тривалості передачі в залежності від параметрів і встановлених категорій обслуговування. Наприклад, для передачі невеликих файлів (Рис.6а) система без будь-якого механізму QoS працює краще (1,82 с) порівняно з системою, яка використовує запропонований нами механізм (2,71 с і 2,40 с для 40% і 60% резервування пропускної здатності відповідно). Як показано на Рис.6b, механізм QoS перевершує традиційну систему (для 60% резервування QoS), скорочуючи середній час

передачі практично на 20%. Нарешті, використання механізму QoS під час передачі великих файлів (Рисунок 6с) демонструє найкращу продуктивність, збільшуючи час передачі приблизно на 23% і 47%, для 40% і 60% резервування відповідно. У цьому випадку, незважаючи на додаткові накладні витрати, архітектура надання QoS демонструє значно кращу продуктивність, що підтверджує її здійсненність.

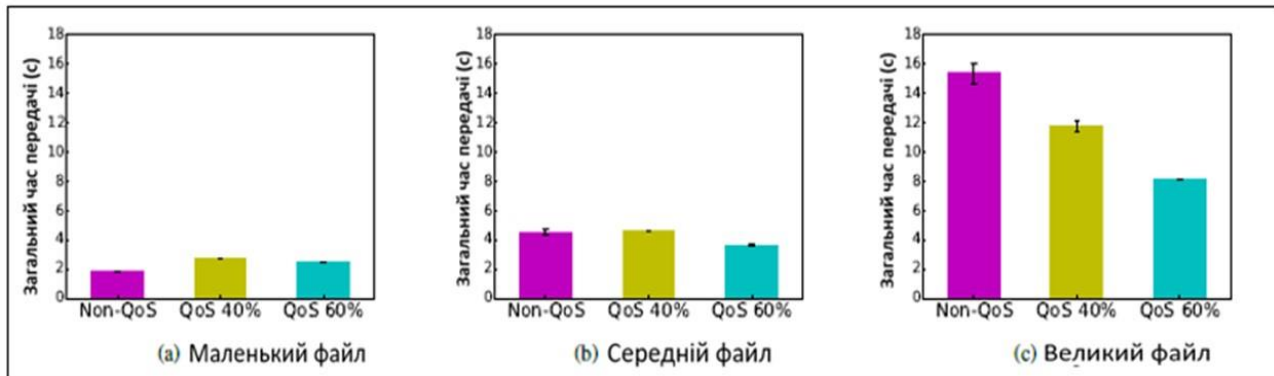


Рис. 6. Середній загальний час передачі даних

Під час додаткової оцінки було проведено порівняльний аналіз парних спостережень між підходом до виконання додатків без QoS і виконанням із резервуванням ресурсів. Були обчислені довірчі інтервали та середні значення для 95% рівня довіри, як показано в Таблиці I. Негативні показники довірчих інтервалів у порівнянні «Не-QoS/QoS 40%» та у «Non-QoS/QoS 60 %» з використанням невеликого файлу вказує на те, що традиційний підхід кращий. У сеансі моделювання із середнім файлом значення 0,0, включене в інтервал порівняння «Non-QoS/QoS 40%», вказує на те, що для цієї конфігурації альтернативи еквівалентні. При збільшенні резервування до 60% застосування архітектури з QoS вже є перевагою, про що свідчить інтервал позитивних значень. Нарешті, порівняння вимірювань із великим файлом показують позитивні інтервали в обох випадках, що свідчить про перевагу підходу до забезпечення QoS, особливо для великих файлів, де вииграш у продуктивності компенсує введені накладні витрати. Ця методологія оцінки підтверджує попередній аналіз.

Таблиця 1 - Порівняльний аналіз за попарними спостереженнями

	Non-QoS/QoS 40%	Non-QoS/QoS 60%
Маленький файл	(-0,92 , -0,87)	(-0,61 , -0,55)
Середній файл	(-0,25 , 0,10)	(0,68 , 1,02)
Великий файл	(2,89 , 4,27)	(6,54 , 7,90)

### Висновки

У роботі була представлена архітектура забезпечення QoS для високопродуктивних розподілених програм, які використовують можливості SDN. У цій архітектурі прості програмні компоненти забезпечують процес обміну повідомленнями, що включає запит на бронювання та підтвердження бронювання. Ця процедура дозволяє мережевому контролеру налаштовувати черги QoS для окремих потоків пакетів. Були розроблені прототипи компонентів програмного забезпечення та проведена їх оцінка у скороченому сценарії як доказ концепції. Архітектура надання QoS забезпечує низькі накладні витрати та скорочує час передачі до 47%. Незважаючи на конкретний оцінений сценарій, результати вказують на здійсненність цього рішення.



**Список використаної літератури**

1. D. Kreutz, F. Ramos, P. Ver'issimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, pp. 14–76, 2015.
2. M. Karakus and A. Duresi, "Quality of service (qos) in software defined networking (sdn): A survey," Journal of Network and Computer Applications, vol. 80, pp. 200–218, 2017.
3. Гніденко М.П., Вишнівський В.В., Ільїн О.О. Побудова SDN мереж. – Навчальний посібник. – Київ: ДУТ, 2019. – 190 с.
4. Гніденко М.П., Кароян Р.Р. Розробка архітектури SD-Branch на основі концепції SD-WAN та обладнання Aruba. Міжнародна науково-практична конференція «Сучасні досягнення компанії Hewlett Packard Enterprise в галузі ІТ та нові можливості їх вивчення і застосування» /грудень/ Київ: ДУТ, - 2020 р.

***Автори статті***

**Гніденко Микола Петрович** – кандидат технічних наук, доцент кафедри комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

**Прокопов Сергій Васильович** – кандидат технічних наук, доцент кафедри комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

**Кароян Роман Романович** – студент, Державний університет телекомунікацій, Київ, Україна.

**Карпик Катерина Олегівна** - студент, Державний університет телекомунікацій, Київ, Україна.

**Петренко Владислав Валентинович** - студент, Державний університет телекомунікацій, Київ, Україна.

***Authors of the article***

**Hnidenko Mykola Petrovych** - Candidate of Science (technic), associate professor of computer science department, State University of Telecommunications, Kyiv, Ukraine.

**Prokopov Serhii Vasylovych** - Candidate of Science (technic), associate professor of computer science department, State University of Telecommunications, Kyiv, Ukraine.

**Karoyan Roman Romanovych** – student, State University of Telecommunications, Kyiv, Ukraine.

**Karpik Kateryna Olehivna** – student, State University of Telecommunications, Kyiv, Ukraine.

**Petrenko Vladyslav Valentynovych** - student, State University of Telecommunications, Kyiv, Ukraine.

Дата надходження в редакцію: 27.01.2022 р.

Рецензент: д.т.н., проф. А.П. Бондарчук