УДК 004.78

С.О. Сілін, І.В. Шостак

Національний аерокосмічний університет імені М.Є. Жуковського «ХАІ», Україна

ПІДХІД ДО СТВОРЕННЯ ВІРТУАЛЬНОЇ МЕРЕЖІ МІЖ IoT-ПРИСТРОЯМИ, ЩО ПОЄДНАНІ ТРАНСЛЯЦІЄЮ МЕРЕЖЕВИХ АДРЕСІВ

Викладено підхід до створення віртуальної мережі, яка буде поєднувати велику кількість IoT-користувачів шляхом створення perr-to-peer з'єднань між пристроями, що знаходяться в мережах різного рангу. Проаналізовані переваги та недоліки уже існуючих рішень зазначеної проблеми, таких як застосування торрент-клієнтів та менеджера Tox. Запропоноване рішення у вигляді такої віртуальної мережі, яка б об'єднувала пристрої незалежно від того, в якій фізичній мережі вони знаходяться. Описана структура такої віртуальної мережі, та принципи, за якими ця мережа може функціонувати.

Ключові слова:*віртуальна мережа, perr-to-peer з'єднання, IoT-користувачі, торрент-клієнти, менеджер Tox.*

Вступ

Актуальність теми. З розвитком сучасних телекомунікаційних технологій доступ до глобальної мережі став можливий з будь якої точки планети. Як результат - кількість пристроїв, які обмінюються інформацією з її допомогою збільшується дуже швидко. В минулому це були персональні комп'ютери (ПК), потім з'явилися смартфони, поява котрих призвела до різкого збільшення кількості активних користувачів глобальної мережі. Зараз набуває популярності новий клас малопотужних пристроїв, які отримали назву «інтернет речі»(Internet of things) [1]. В перспективі, кількість цих пристроїв може стати більшою, ніж ПК і смартфони разом взяті. Тому удосконалення і розробка методів комунікації між ними є актуальною проблемою.

Аналіз існуючих рішень. Оскільки глобальна мережа Інтернет по своїй суті являє собою сукупність різнорангових мереж, то пристрої, що знаходяться в різних мережах одного рангу, не зможуть встановити з'єднання між собою без сторонньої допомоги, оскільки цьому заважає трансляція мережених адресів, або англійською Network Address Translation (NAT) [1]. Варіантом вирішення цієї проблеми може бути створення центрального веб-сервера, який має публічний IP адрес [2], та який є доступним для всіх користувачів, котрі знаходяться в низько рангових мережах. Але зі збільшенням загальної кількості активних пристроїв зростає і навантаження на веб-інфраструктуру. З цього виходить, що для підтримки ефективної роботи системи необхідні постійні інвестиції в обладнання. Якщо цього не робити, то спочатку це може призвести до зменшення відгуку системи в цілому, а потім і до повної її відмови. Іншим варіантом побудови розподілених застосунків може бути створення peer-to-peer [3] з'єднань між пристроями з прямим обміном інформацією. Як результат, всі пристрої повинні бути в одній мережі одного рангу, або мати статичні IP адреси, якщо обумовлює необхідність керування ними через глобальну мережу Інтернет. Зазначений процес реалізувати часом не можливо, особливо, якщо пристрої у складі мережі розподілені по всьому світу. Рішенням цієї проблеми може бути створення віртуальної мережі, яка б об'єднувала всі пристрої в єдине середовище, та надавала б можливість створювати peer-to-peer з'єднання між ними., незалежно від того, в яких фізичних мережах знаходиться кожен з цих пристроїв.

Сама можливість створення peer-to-peer з'єднань між пристроями, що сховані за NAT не нова. Яскравими прикладами можуть бути різноманітні реалізації torrent-клієнтівр [3] та захищений меседжер Tox [4]. Кожен з них використовує свою власну реалізацію яку важко виділити як окремий модуль, тому розробка подібного рішення коштує дорого та займає багато часу. Крім цього їх функціональності недостатньо для створення повноцінної віртуальної мережі.

Мета статі полягає в описі вимог до захищеної віртуальної мережі, яка б надала змогу організовувати peer-to-peer з'єднання між пристроями, що знаходяться за NAT.

Основна частина

Розглянемо переваги підходу до побудови віртуальної мережі на такому прикладі. У нас є будинок із встановленими декількома IoT пристроями, якими можливо керувати за допомогою мобільного застосунка. Цей мобільний застосунок має доступ до центрального серверу з публічним адресом, який, в свою чергу, з'єднаний із всіма IoT пристроями. Коли нам потрібно відправити команду для девайса в будинку, то мобільний застосунок спочатку зв'язується з центральним сервером, котрий в свою чергу опрацьовує команду, розуміє її, та пересилає далі. Щоб це було можливо, пристрій, для якого призначена команда, повинен постійно підтримувати зв'язок із центральним сервером. Після виконання команди він може відправити якусь відповідь для мобільного застосунка. Тут повторюється уже відома нам послідовність, але вже навпаки. Щоб оптимізувати витрати, центральний сервер буде працювати одночасно з декількома будинками, що у випадку його відмови вплине нас всіх користувачів.

Тепер розглянемо той самий приклад, але реалізований за допомогою віртуальної мережі. Вона буде

складатися з декількох IoTдевайсів і смартфона. При необхідності відправити команду, за стосунок отримає в однієї із нод адрес необхідного пристрою, запам'ятає його та відправить команду безпосередньо на пристрій, якій її опрацює. Після того він отримає адрес мобільного девайса, запам'ятає його і відправить відповідь. Якщо знадобиться відправити нову команду, то мобільний застосунок вже буде знати адресу і відразу відправить її, те ж саме справедливо дляIoTдевайсу. Але слід відмітити, що так буде відбуватися не завжди, бо існує ймовірність того, що NAT може змінити публічний адрес пристрою; в цьому випадку, не отримавши підтвердження того, що команда дійшла до адресата, мобільний застосунок оновить адресу за допомогою ноди і знов відправить команду. Подібна ситуація буде траплятися не часто, оскільки кожен раз при її виникненні нода буде змінювати налаштування holepunching-a, щоб уникнути її повторення в майбутньому.

Розглянемо обидва приклади більш детально. У першому випадку центральний сервер бере активну участь в комунікації між мобільним застосунком та IoT пристроєм і весь трафік йде через нього. В результаті може наступити момент, коли ширини каналу передачі даних може не вистачити. Це може навіть трапитися до повної утилізації центрального серверу, оскільки він не виконує важких розрахунків.

При реалізації подібного застосунку через віртуальну мережу старт ноди буде викликати різкий сплеск навантаження, але коли всі мережі сформуються, настане період відносно рідкого коригування з'єднань між окремими пристроями. При цьому буде спостерігатися отримання періодичних пакетів для підтримки hole punching-a, а весь трафік буде йти прямо від відправника до адресата. Таким чином, трафік буде рівномірно розподілений між всіма мережами. Це означає, що одна нода зможе підтримувати роботу набагато більшої кількості пристроїв одночасно в порівняні з централізованим сервером.

На практиці існує багато застосунків, які створюють peer-to-peer з'єднання між своїми клієнтами. Наприклад, торрент клієнти використовують публічні сервери-трекери для отримання та підтримки публічних адресів. Можливостей цих трекерів навіть достатньо для створення віртуальної мережі між декількома пристроями. В такій мережі хешторрент файл буде її ідентифікатором. Всі, кому він відомий, зможуть приєднайтеся до неї, та дізнатися поточні адреси всіх клієнтів. Але трекери не мають інтерфейсу для ідентифікації кожного окремого пристрою та обмеження доступу для тих, що не пройшли авторизацію. Крім того, відсутня можливість прямого з'єднання між окремими пристроями, які не знаходяться в одній мережі. В свою чергу,менеджер Tox має таку можливість. Кожен активний застосунок генерує публічний та приватний ключ при першій активації. Публічний ключ стає унікальним публічним ідентифікатором пристрою, знаючи його, достатньо приєднатися до tox-ноди для того щоб дізнатися поточний адрес пристрою. За допомогою приватного ключа шифруються всі пакети, що меседжерTox відправляє іншим пристроям. Але тут існує проблема, tox-ноди не мають можливості об'єднати окремі пристрої в групу, що необхідно для організації повноцінної віртуальної мережі.

Структура та протокол віртуальної мережі. Інфраструктура для підтримки подібних віртуальних мереж може складатися з необмеженої кількості нод. У кожної мережі повинний бути свій адміністратор. Тільки він має право на підключення чи відключення пристроїв до неї. Створення та наступне управління мережею буде виконуватися на сервері-ноді.

Авторизація адміністратора та будь якого пристрою буде проходити з використанням публічного та приватного ключа, аналогічно з меседжером Tox. Кожна мережа повинна мати унікальне ім'я, яке буде слугувати для об'єднання пристроїв, аналогічно з хешем в торренттрекерах. Процедуру створення віртуальної мережі можливо показати у вигляді діаграми послідовності (рис. 1).

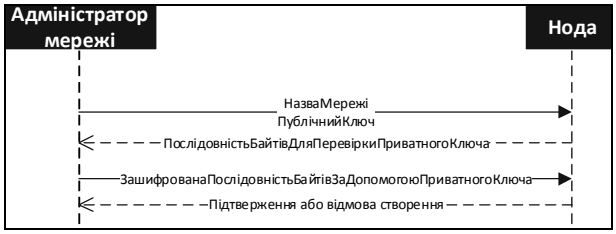

Рис. 1. Діаграма послідовності для створення віртуальної мережі

На цій і наступних діаграмах інформація, яка потрібна для ідентифікації поточного з'єднання відсутня, оскільки вона залежить від типу з'єднання. Як видно із діаграми, адміністратор повинний надати ім'я мережі та свій публічний ключ. У відповідь нода надішле послідовність байтів, яку необхідно буде зашифрувати за допомогою приватного ключа та відправити назад. Якщо після розшифровки надіслані байти співпадуть із отриманими, то мережа буде створена, а її адміністратором буде вважатися будь хто, якщо має відповідний приватний ключ. Підключення нового пристрою можливо показати діаграмою (рис. 2).

Пристрій, що необхідно підключити до мережі, спочатку необхідно авторизувати, ця операція реалізується в перших трьох повідомленнях, аналогічно до авторизації адміністратора. Після чого нода відправить повідомлення до адміністратора для підтвердження підключення і, залежно від його рішення, підключить, або відмовить у підключенні. Якщо адміністратор на даний час відсутній в мережі,нода помістить запит в список, котрий буде доступний адміністратору, колі той з'явиться в мережі. Відправлення повідомлення від пристрою до пристрою показано на рис. 3. Як видно із діаграми (рис. 3), спочатку перший пристрій дізнається про поточну адресу іншого пристрою у ноди, а вже потім між ними виникає повноцінна взаємодія.

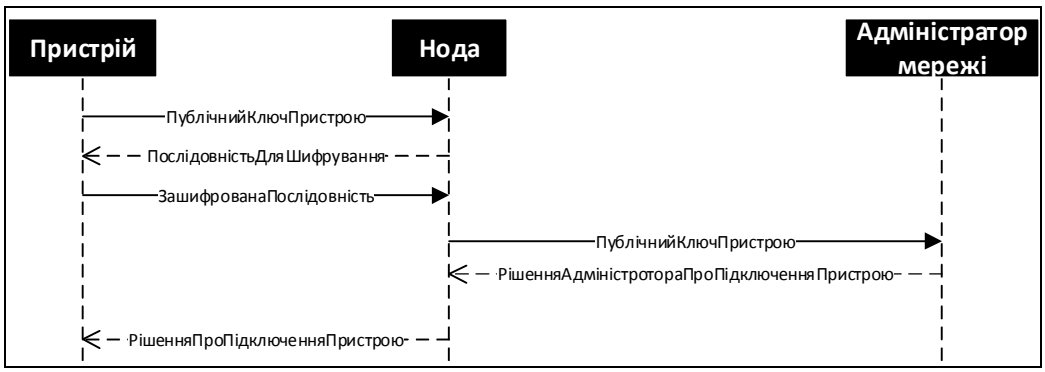

Рис. 2. Діаграма послідовності для підключення нового пристрою до віртуальної мережі

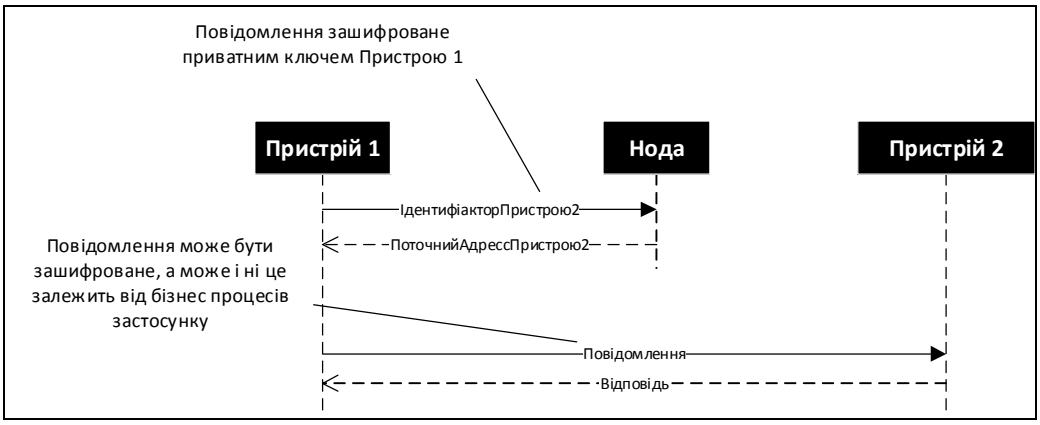

Рис. 3. Діаграма послідовності для відправки повідомлення іншому пристрою віртуальної мережі

Висновки

В статі розглянута проблема створення perr-to-peer з'єднань між пристроями що знаходяться в мережах різного рангу.

Проаналізовані переваги та недоліки уже існуючих рішень, таких як застосування торрент-кліентів та менеджера Tox.

Запропоноване рішення у вигляді такої віртуальної мережі, яка б об'єднувала пристрої незалежно від того, в якій фізичній мережі вони знаходяться.

Описана структура такої віртуальної мережі та принципи,за якими ця мережа може функціонувати.

Список літератури

1. InternetofThings [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Internet_of_things

2. Netrwork address translation [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/NAT

3. BitTorrent [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/BitTorrent

4. ToxMessager [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Tox_(protocol).

Надійшла до редколегії 2.04.2017

Рецензент: д-р техн. наук, проф., О.Є. Федорович, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.

ПОДХОД К СОЗДАНИЮ ВИРТУАЛЬНОЙ СЕТИ МЕЖДУ IoT-УСТРОЙСТВАМИ, СОЕДИНЕННЫМИ ТРАНСЛЯЦИЕЙ СЕТЕВЫХ АДРЕСОВ

С.О. Силин, И.В. Шостак

Изложены подход к созданию виртуальной сети, которая будет сочетать большое количество IoT-пользователей путем создания perr-to-peer соединений между устройствами, находящимися в сетях различного ранга. Проанализированы преимущества и недостатки уже существующих решений данной проблемы, как применение торрент-клиентов и менеджера Tox. Предложенное решение в виде такой виртуальной сети, которая объединяла устройства независимо от того, в какой физической сети они находятся. Описана структура такой виртуальной сети, и принципы, по которым эта сеть может функционировать.

Ключевые слова: *виртуальная сеть, perr-to-peer соединения, IoT-пользователи, торрент-клиенты, менеджер Tox.*

APPROACH CREATE A VIRTUAL NETWORK BETWEEN IoT-DEVICES COUPLED NETWORK ADDRESS TRANSLATION

S.O. Silin, I.V. Shostak

The approach to create a virtual network that will combine a large number of IoT-friendly by creating perr-to-peer connections between devices within the networks of different rank. Advantages and disadvantages of existing solutions to this problem, such as the use of torrent clients and manager Tox. The proposed solution such as virtual network device which would unite regardless of where they are physical network. We describe the structure of this virtual network, and the principles on which the network can function.

Keywords: *virtual network, perr-to-peer connection, IoT-users torrent clients, manager Tox.*