УДК 004.7

Р.Б. АНДРУЩЕНКО[*], А.Д. БЕСКОСТИЙ[*], С.В. ЗАЙЦЕВ[*], Я.Ю. УСОВ[*], М.А. ПИСЬМЕНЮК[*]

ДОСЛІДЖЕННЯ МЕТОДІВ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ПЕРЕДАЧІ ДАНИХ У СТЕКАХ ПРОТОКОЛІВ TCP/IP СИСТЕМ ПУБЛІЧНОГО АДМІНІСТРУВАННЯ

[*]Чернігівський національний технологічний університет, м. Чернігів, Україна

Анотація. *Стек протоколів TCP/IP та модель OSI зарекомендували себе як надійну, відносно просту, технологію передачі даних у комп'ютерних мережах. Проте, за наявності надлишкового шуму у каналі передачі даних виникає необхідність у застосуванні додаткових методів підвищення достовірності передачі інформації. В роботі проведено аналіз існуючих методів підвищення цілісності передачі даних у комп'ютерних мережах, побудованих на базі сімейства протоколів TCP/IP за рахунок застосування завадостійкого кодування. Виявлено їх особливості функціонування, переваги і недоліки та розглянуто основні напрями досліджень у цій області.*
Ключові слова: *канал передачі даних, TCP/IP, комп'ютерна мережа, кодування.*

Аннотация. *Стек протоколов TCP/IP и модель OSI зарекомендовали себя как надежную, относительно простую, технологию передачи данных в компьютерных сетях. Однако, при наличии избыточного шума в канале передачи данных возникает необходимость в применении дополнительных методов повышения достоверности передачи информации. В работе проведён анализ существующих методов повышения целостности передачи данных в компьютерных сетях, построенных на базе семейства протоколов TCP/IP за счет использования помехоустойчивого кодирования. Изучены их особенности функционирования, преимущества и недостатки, рассмотрены основные направления исследований в данной области.*
Ключевые слова: *канал передачи данных, TCP/IP, компьютерная сеть, кодирование.*

Abstract. *The TCP/IP protocol stack and the OSI model have already proved their reliability and relatively simplicity in data transmitting over the computer networks. However in case of presence of the excessive noise in the data transmitting channel, the additional methods of increasing of data reliability are required. The paper analyzes the existing methods of increasing the reliability of data transmission in computer networks based on TCP/IP protocol family using anti-jamming coding techniques. It discovers the characteristics of their functionality, advantages and disadvantages; considers the main directions of research in this area.*
Keywords: *data transmission channel, TCP/IP, computer network, coding.*

1. Вступ. Актуальність теми дослідження

Аналіз тенденцій розвитку комп'ютерних мереж показує, що стек протоколів передачі даних TCP/IP використовується у величезній кількості корпоративних мереж, а також забезпечує зв'язок вузлів всесвітньої інформаційної мережі Інтернет. Масове поширення безпровідних мереж та зростання кількості пристроїв, які взаємодіють між собою за допомогою мережевих протоколів сімейства TCP/IP та які потребують доступу до глобальної мережі Інтернет, приводить до необхідності приділяти більшу увагу до реалізації протоколів зв'язку, їх безпеки та пропускної здатності.

2. Постановка проблеми

Починаючи ще з 90-х років минулого століття комп'ютерні мережі з кожним роком стають все складнішими та поширенішими. З того часу змінилось декілька поколінь мобільного зв'язку, а кількість пристроїв, що мають доступ до всесвітньої мережі Інтернет, зросла в тисячі разів. Незважаючи на це, найпоширеніші протоколи передачі даних транспортного рівня – Transmission Control Protocol (TCP) та User Datagram Protocol (UDP), майже не за-

знали змін. У більшості випадків вони добре справляються з поставленими перед ними завданнями. Однак в останній час все більшої популярності набувають безпровідні мережі, які характеризуються певними особливостями, зокрема, значно більшим впливом завад на пакети даних, що, у свою чергу, знижує ефективність каналу передачі даних. Тому постає важливе питання, як саме можна знизити вплив завад та збільшити пропускну здатність і достовірність передачі даних протоколів транспортного рівня. Саме аналізу існуючих методів і присвячена дана стаття.

3. Аналіз останніх досліджень і публікацій

Загалом, усі роботи в даному напрямі можна розділити на три види [1–6]. Перші [1–3] – це використання оптимізованих апаратних рішень у пристроях прийому/передачі даних, другі [4, 5] – підвищення ефективності протоколів зв'язку, треті [6] – комбіновані. До других та третіх відносять різноманітні методи підвищення цілісності пакетів TCP і UDP за рахунок застосування корекційних кодів Forward Error Correction (FEC), методів Automatic Repeat Request (ARQ – автоматичні запити на повторну передачу), а також гібридних методів (FEC-TCP та ARQ) в умовах підвищеного шуму в каналі, який призводить до викривлення переданої інформації. Аналіз досліджень щодо порушення цілісності протоколів міжмережевої взаємодії свідчить, що в теперішній час протоколи TCP/IP широко використовуються в безпровідних каналах передачі даних, які характеризуються підвищеним рівнем шуму. Це призводить за певних обставин до виникнення багатьох помилок у прийнятих пакетах даних, а, отже, до повторних передач цих пакетів. Крім того, на зниження цілісності протоколів TCP/IP впливають віддалені мережеві атаки, наприклад, атаки типу «зашумлення». Існуючі методи підвищення цілісності передачі даних протоколів TCP/IP за певних умов завадової обстановки не забезпечують гарантовану доставку повідомлень.

4. Мета статті

Головною метою даної роботи є визначення основних методів підвищення цілісності передачі даних у сучасних комп'ютерних мережах, аналіз основних напрямів досліджень у цій галузі, а також визначення подальших шляхів підвищення цілісності протоколів міжмережевої взаємодії TCP/IP.

5. Виклад основного матеріалу

Для більш повної картини спочатку розглянемо стандартний протокол гарантованої доставки повідомлень транспортного рівня – TCP-протокол.

Протокол TCP реалізує взаємодію в режимі встановлення логічного (віртуального) з'єднання, забезпечує двосторонній дуплексний зв'язок. Він організовує потоковий (з точки зору користувача) тип передачі даних та надає можливість пересилання частини даних як екстрених. Для ідентифікації вузлів на транспортному рівні протокол TCP використовує 16-бітові номери портів. Також протокол TCP реалізує принцип sliding window для підвищення швидкості передачі й підтримує ряд механізмів для забезпечення надійної передачі даних. Незважаючи на те, що для користувача передача даних з використанням протоколу TCP виглядає як потокова, насправді ж обмін між вузлами здійснюється за допомогою пакетів даних [7].

У більшості випадків документ, електронна пошта або інша інформація не надсилається вся і відразу. Замість цього, вона розбивається на невеликі пакети даних, кожен з яких містить частину даних та заголовок, що визначає правильну послідовність даних. Коли пакети даних надсилаються по мережі, вони можуть надходити до адресата різними маршрутами та в різний момент часу. Але це не має значення. Приймач організовує всі прийняті пакети даних повторно в належному порядку. Після того, як усі пакети отримані, ге-

нерується повідомлення до вихідної мережі. Якщо ж деякий пакет не надходить до адресата, то відправляється повідомлення до вихідної мережі про необхідність повторної відправки пакета. Таким чином, TCP забезпечує гарантовану доставку даних до адресата [8, 9].

У протоколі TCP використовується контрольна сума для перевірки цілісності пакетів даних – 16-бітне доповнення суми всіх 16-бітних слів заголовка й даних. Якщо сегмент містить у заголовку й тексті непарну кількість октетів, що підлягають обліку в контрольній сумі, то останній октет буде доповнений нулями праворуч для того, щоб утворити для надання контрольній сумі 16-бітне слово. Утворений при такому вирівнюванні октет не передається разом із сегментом по мережі. Перед обчисленням контрольної суми поле цієї суми заповнюється нулями.

Контрольна сума, крім усього іншого, враховує 96 бітів псевдозаголовка, який для внутрішнього використання розміщується перед TCP-заголовком. Цей псевдозаголовок містить адреси відправника, одержувача, протокол та довжину TCP-сегмента [7].

Коли TCP-пакет надходить до місця призначення, приймаюче програмне забезпечення виконує той же самий розрахунок контрольної суми. Він також виконує дії по створенню псевдозаголовка, який потім так само приєднується перед фактичним TCP-сегментом, а потім розраховує контрольну суму (встановлюючи значення поля контрольної суми в 0 для розрахунку, як і вузол, що відправив дані). Якщо існує невідповідність між його розрахунком та значенням у полі контрольної суми, це означає, що виникла помилка, і пакет зазвичай відхиляється [7].

Легко бачити, що використовувана контрольна сума надто примітивна. У випадку надлишкового шуму в каналі передачі даних маємо велику кількість запитів на повторну передачу, що значно знижує пропускну здатність каналу.

Альтернативою є використання, наприклад, FEC-кодів. Особливо це стосується передачі мультимедійних даних великого об’єму, оскільки найбільше навантаження на мережу спричиняється саме мультимедійними даними.

Forward Error Correction (FEC) – кодування/декодування сигналу з можливістю виявлення помилок і корекції інформації. Таким чином, приймач може виявляти і виправляти помилки, що виникають у каналі передачі без запитів на повторну передачу фрагментів даних, що були втрачені або пошкоджені. Отже, FEC-кодування доцільно застосовувати у випадках, якщо ретрансляція пакетів є дорогою операцією. Існує кілька FEC-алгоритмів кодування, які розрізняються за складністю та продуктивністю. Одним із найбільш поширених кодів першого покоління FEC є, наприклад, код Ріда-Соломона. Коди Ріда-Соломона – це циклічні коди, що дозволяють виправляти помилки у блоках даних. Елементами кодового вектора є не біти, а групи бітів (блоки). Дуже поширені коди Ріда-Соломона, що працюють з байтами (октетами).

Ця особливість робить коди Ріда-Соломона особливо корисними для протидії груповим помилкам: шість послідовних бітових помилок, наприклад, можуть пошкодити максимум два байти. Тому навіть версія коду Ріда-Соломона з корекцією подвійної помилки може забезпечити достатній запас міцності. Математично коди Ріда-Соломона засновані на арифметиці кінцевих полів [10].

Код Ріда-Соломона має мінімальну відстань:

$$d = 2t + 1 = n - k - 1. \qquad (1)$$

Це код з максимально досяжною кодовою відстанню, тобто при фіксованих n і k не існує коду, у якого мінімальна відстань більша, ніж у коду Ріда-Соломона. Код Ріда-Соломона будується на довжинах $N = q - 1$ у полі GF(q) за породжувальним многочленом:

$$g(x) = (x - \alpha^{j_0}) \cdot (x - \alpha^{j_0+1}) \ldots (x - \alpha^{j_0+2t-1}), \qquad (2)$$

де α – примітивні елементи поля $GF(q)$, $j_0 = \overline{(1,n)}$ – довільні елементи поля, t – виправляюча здатність коду. Зміна будь-якого з параметрів (n, α, j_0, t) породжувального многочлена коду Ріда-Соломона призводить до утворення нового суміжного класу коду. Варто зазначити, що якщо на приймальній стороні не відомий закон зміни параметрів $g(x)$, то декодування є складним обчислювальним завданням. Крім того, коди Ріда-Соломона мають структурну властивість: змінюючи основу алфавіту q, можна виправляти як одиночні помилки, так і пакети помилок [10, 11].

Існує метод покращання медіа-протоколів за допомогою використання FEC. У цьому методі разом із медіа-пакетами транспортного протоколу Real-time Transport Protocol (RTP) в режимі реального часу надсилаються резервні пакети для перевірки наявної пропускної здатності. Резервні пакети – це закодовані RTP-пакети з FEC-кодами [12].

Основна ідея використання FEC для управління швидкістю полягає в тому, щоб передавати надлишкові FEC-пакети разом із потоком даних та використовувати їх для перевірки доступної пропускної здатності. Якщо умови в середовищі забезпечують стабільну та надійну передачу, то потік FEC-пакетів зменшується, тим самим швидкість кодування даних стає більшою. Потік FEC-пакетів може навіть взагалі припинятись у разі дійсно стабільного каналу.

Якщо бути більш точним, то суть цього методу – в підрахунку кількості втрат надлишкових FEC-пакетів. Якщо в мережі відсутні втрати надлишкових пакетів, відправник може збільшити швидкість передачі за рахунок зменшення кількості надлишкових FEC-пакетів. У випадку ж втрат через конфлікти в середовищі кількість резервних пакетів, навпаки, збільшується. Більша кількість резервних пакетів дозволяє відновити більше втрачених пакетів.

На стороні передавача модуль керування швидкістю обчислює новий бітрейт. Якщо нова швидкість передачі даних вище, ніж попередня, то модуль FEC, залежно від його внутрішнього стану, додає FEC-дані. Після цього він вказує модулю керування поточний бітрейт для потоку даних, який може бути меншим, ніж розрахункова швидкість передачі.

На стороні приймача модуль FEC реконструює втрачені пакети даних. Якщо пакети успішно відновлюються, то модуль створює звіт про втрати після ремонту для відповідних пакетів даних. Також модуль FEC отримує додаткові дані Real-time Transport Control Protocol (RTCP), пов'язані з основним потоком даних, та звіт про виправлені помилки. Він використовує RTCP-дані для розрахунку ефективності FEC.

Типова схема з модулем FEC зображена на рис. 1 нижче.

Основна перевага даного підходу полягає в тому, що алгоритм керування швидкістю в ньому може бути легко масштабовано та змінено в залежності від наявної потужності каналу, оскільки підвищена надійність компенсує можливі помилки, які виникають у зв'язку з надмірним використанням каналу зв'язку [12].

TCP with dynamic FEC (TCP-dFEC). Основна відмінність цього методу полягає в тому, що в залежності від втрат пакетів у мережі регулюється не кількість надлишкових пакетів, а складність виправляючого коду.

Протокол TCP-dFEC може змінювати рівень складності коду. Тобто, він є адаптивним. Адаптивність базується на підрахунку значення залишкових втрат. Залишкові втрати показують не саму кількість втрат, а її зміну в часі. Тобто, значення залишкових втрат може показати, наскільки надійною стала мережа передачі даних у даний момент часу у порівнянні з попереднім моментом часу. Якщо бути більш точним, то у протоколі TCP-dFEC значення залишкових втрат розраховується протягом заданого періоду T. Воно розраховується як співвідношення кількості повторно переданих пакетів за даний та попередній періоди. Далі береться середнє значення залишкових втрат за декілька періодів (це зроблено для того, щоб згладити значення), яке потім і порівнюється з зарані заданою константою –

цільовим рівнем втрат. Якщо отримане середнє значення вище, ніж цільове, то складність коду FEC зменшується, інакше – збільшується.

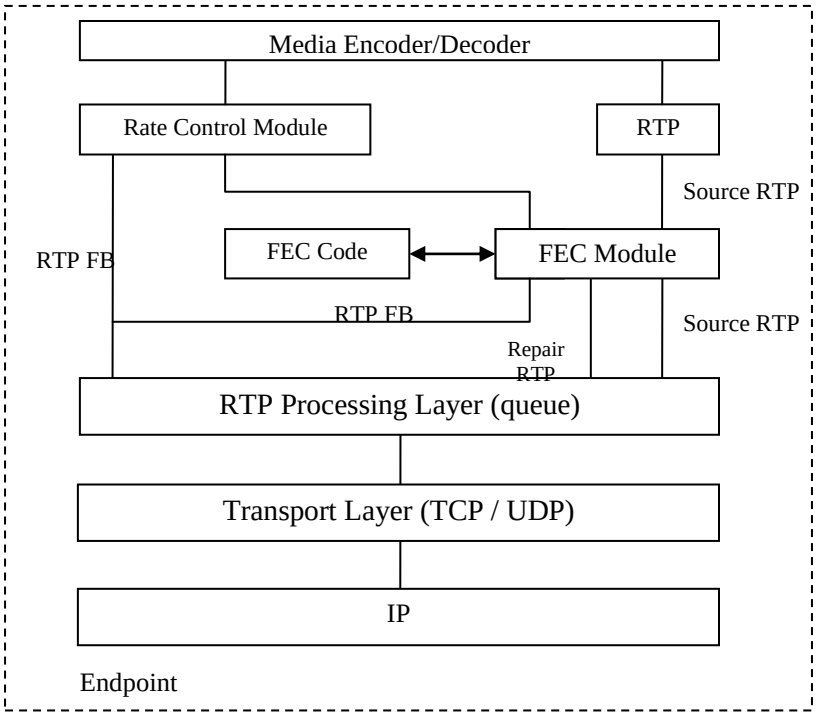

Рис. 1. Схема використання модуля FEC-кодування для контролю навантаження

TCP-dFEC забезпечує таку ж швидкість передачі даних, як і звичайний TCP при низьких втратах у каналі передачі даних. Але в той же час він забезпечує в середньому на 40% більшу швидкість передачі даних при високих втратах у каналі [13].

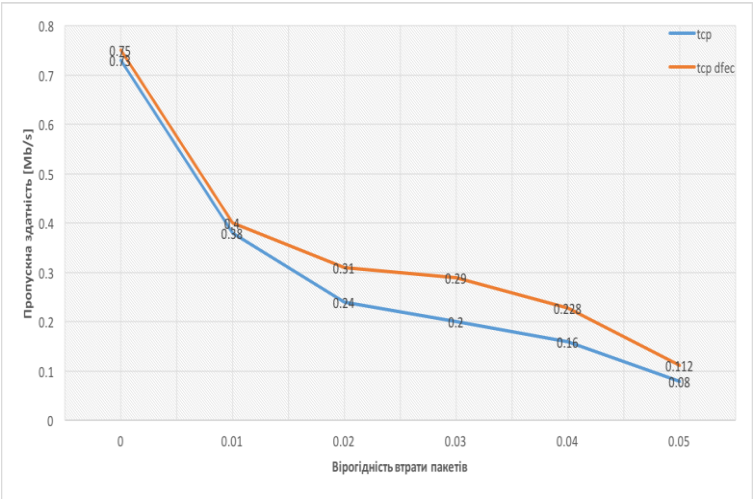

Рис. 2. Порівняння TCP та TCP-dFEC

Також різновидом ідеї використання кодів корекції помилок є залучення Fountain-кодів. Їх особливість полягає у тому, що для передачі M пакетів використовується $\min(M+F)$ закодованих пакетів, тобто додається додаткова інформація. Причому для успішного прийому даних необхідно успішно прийняти будь-які N пакетів з набору [14]:

$$M < N < \min(M+F). \qquad (3)$$

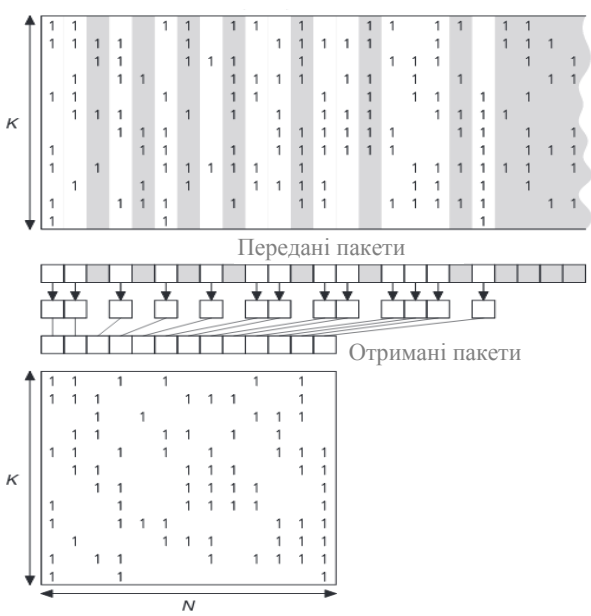

Рис. 3. У результаті втрати пакетів немає необхідності використовувати запити на повторну передачу. Достатньо зібрати будь-які *N* пакетів

На стороні передавача можна не визначати, скільки пакетів необхідно передати. Нема необхідності застосовувати запити на повторну передачу. Пакети генеруються до тих пір, поки приймач не прийме будь-які *N* пакетів.

У результаті кодування на виході отримуємо матрицю даних для передачі. Приймач у разі втрати пакетів отримує розріджену матрицю. Проте, маючи достатню кількість стовпчиків, він може повністю декодувати передану інформацію [14].

Одним із найпотужніших методів, що у своїй роботі використовують коди корекції помилок, є AprilFEC. Його застосування доречне більше для передачі мультимедійних файлів. AprilFEC побудований на UDP і забезпечує надійну доставку шляхом використання fountain-кодів. Ці коди ефективно адаптуються до рівня втрат у каналі, тому AprilFEC може підтримувати достовірну та швидку передачу даних при мінімізації накладних витрат на мережу.

Після того, як користувач передасть дані в AprilFEC, система розбиває їх на менші фрагменти, придатні для передачі в корисних навантаженнях пакетів UDP. Базуючись на оцінці миттєвої швидкості втрати пакетів, AprilFEC обчислює та передає партію закодованих фрагментів. Система миттєво призупиняється і чекає отримання підтвердження від кінцевого вузла, після чого він припиняє процес передачі. Якщо ACK не отримано, AprilFEC повторить процес із більш високим ступенем втрати пакетів. Після декількох повторень AprilFEC завершує процес передачі навіть за відсутності будь-якого ACK [15].

У кінцевому вузлі AprilFEC збирає закодовані фрагменти і намагається відновити оригінальне повідомлення від них. Після того, як воно зможе відновити вихідні дані, AprilFEC передає дані користувачеві та надсилає ACK назад до передавального вузла.

Частка успішних передач під час моделювання потокового зображення, керованого реальними робочими мережевими даними, зображена на рис. 4.

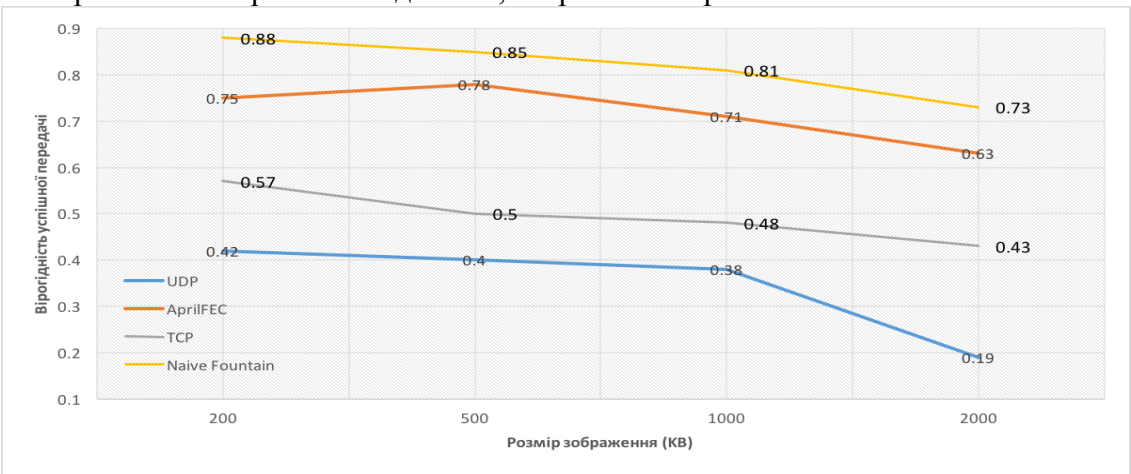

Рис. 4. Вірогідність успішної передачі зображення в залежності від розміру даних при використанні AprilFEC та без нього

AprilFEC значно перевершує TCP, традиційний надійний протокол доставки. Коли частота втрати пакетів висока, частота появи повідомлень ACK збільшується через те, що TCP припиняє очікування повторної передачі. Для порівняння AprilFEC використовує адаптивну схему виправлення помилок для передачі даних з високою ймовірністю навіть в умовах поганої мережі.

Не зважаючи на збільшення втрат пакетів, AprilFEC продовжує передавати зображення. Причому, чим більша кількість втрачених пакетів, тим більш ефективно працює AprilFEC по відношенню до класичного TCP (рис. 5).

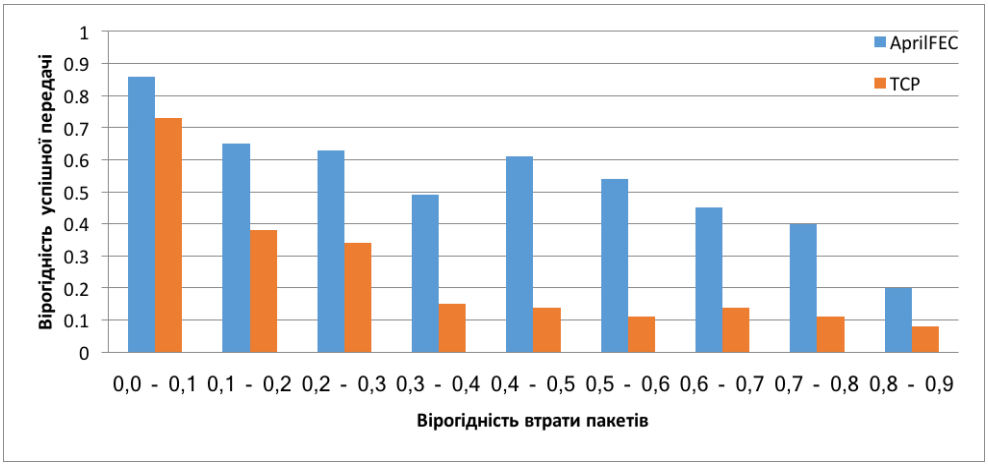

Рис. 5. Вірогідність успішної передачі зображення 1 Мб залежно від вірогідності втрати пакетів

З рис. 4 можна зробити висновок, що Naive Fountain значно краще справляється з поставленими перед ним завданнями. Однак це не так.

У більшості випадків використання Naive Fountain призводить до набагато більших накладних витрат порівняно з AprilFEC. Так як Naive не здійснює оцінку кількості фрагментів, які необхідно закодувати та передати, тому зменшує пропускну здатність каналу передачі інформації. Коефіцієнт накладних витрат зображено на рис. 6 [15].

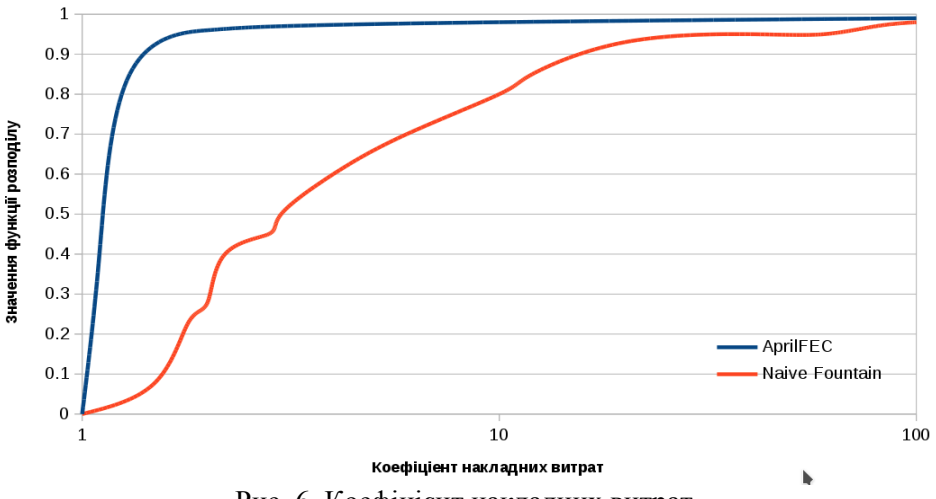

Рис. 6. Коефіцієнт накладних витрат

Метод End-to-end FEC-TCP. Реалізується або як частина транспортного рівня, або безпосередньо на рівень нижче, як на стороні відправника даних, так і на стороні приймача. Це найбільш підходящий сценарій для впровадження FEC-FTP. Кодер FEC може легко

отримати інформацію про поточне значення вікна TCP і виконати миттєве сповіщення у випадку втрат, пов'язаних з перевантаженням, безпосередньо з TCP-рівня.

Як видно на рис. 7, пропускна здатність FEC-TCP більш ніж у двічі більша від класичного TCP для вірогідності втрати пакетів між 0,01 та 0,02. Пропускну здатність можна підвищити шляхом застосування більш складних алгоритмів кодування та збільшенням надлишкової інформації на один блок даних.

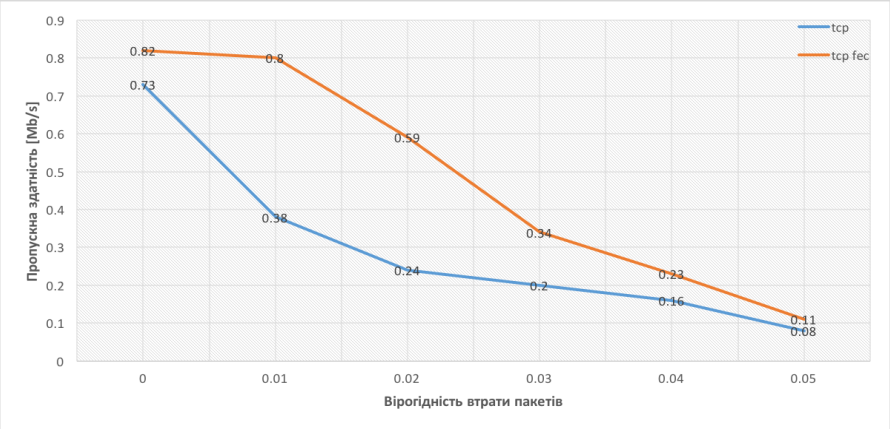

Рис. 7. Порівняння TCP та End-to-end TCP-FEC (при розмірі пакета 1000 байтів)

End-to-end FEC-TCP також можна використовувати не лише для підвищення пропускної здатності каналу зв'язку, а і для оцінки ймовірності втрати пакетів у мережі. Тобто End-to-end FEC-TCP також можна застосувати як альтернативне рішення для попереджуючого виявлення максимальної пропускної здатності каналу зв'язку [16].

6. Висновки і пропозиції

1. Стек протоколів TCP/IP та модель OSI зарекомендували себе як надійну, відносно просту технологію передачі даних у комп'ютерних мережах. Проте за наявності надлишкового шуму у каналі передачі даних виникає необхідність у застосуванні додаткових методів підвищення достовірності передачі інформації.
2. Протоколи TCP і UDP, що широко застосовуються у комп'ютерних мережах, майже не зазнали змін. У більшості випадків вони добре справляються з поставленими для них завданнями, проте виникають ситуації, за яких стандартних рішень не достатньо для забезпечення необхідного рівня достовірності передачі даних.
3. Розглянуто методи модифікації стеку протоколів TCP/IP на транспортному та вищих рівнях: RTP-FEC для управління потоком даних, адаптивний dFEC-TCP, Fountain-кодування, AprilFEC та End-to-end FEC-TCP, що дозволило визначити напрями подальших досліджень у цій галузі за рахунок застосування завадостійких кодів.
4. Недоліком методів, які використовують FEC-коди, є низька адаптивність до зміни характеристик каналу передачі даних. Крім того, для FEC-методів характерна деяка затримка у часі, що витрачається на кодування, декодування та використання надлишкових ресурсів каналу для передачі закодованої інформації. AprilFEC показує хороші результати за умови вчасного прогнозування втрати пакетів. Однак втрати в ненадійних каналах передачі даних можуть залежати від чинників, які важко спрогнозувати. Використання Fountain-кодів може призводити до великої кількості надлишкових пакетів у каналі у випадку, якщо передавач не отримає за якихось причин повідомлення від приймача про успішний прийом даних.
5. Для підвищення характеристик достовірності інформації у сучасних комп'ютерних мережах пропонується додаткове застосування методів завадостійкого кодування, зокрема, турбокодів та кодів Ріда-Соломона.

СПИСОК ДЖЕРЕЛ

1. Nair R. A Symbol Based Algorithm for Hardware Implementation of Cyclic Redundancy Check (CRC) / R. Nair, G. Ryan, F. Farzaneh // VHDL International Users' Forum. – 1997. – P. 82 – 87.
2. Liu Q. Implementation of hardware TCP/IP stack for DAQ systems with flexible data channel / Q. Liu, X. Zhiqiang, L. Zhengying // Electronics Letters. – 2017. – Vol. 53, Issue 8. – P. 530 – 532.
3. Assar A. A hardware implementation of the TCP protocol applying TCP-BIC and TCP-CUBIC standards / A. Assar, K. Hofmann // Microelectronics (ICM), 28th International Conference. – Giza, Egypt, 2016. – P. 37 – 40.
4. Dalessandro D. iWarp protocol kernel space software implementation / D. Dalessandro, A. Devulapalli, P. Wyckoff // Parallel and Distributed Processing Symposium, IPDPS 2006. – Rhodes Island, Greece, 2006. – P. 274.
5. Thanh V.T. Mobile TCP socket for secure applications / V.T. Thanh, Y. Urano // Advanced Communication Technology (ICACT), The 12th International Conference. – Nangang, China, 2010. – P. 971 – 974.
6. Hong R.L. Research and application of TCP/IP protocol in embedded system / R.L. Hong // 2011 IEEE 3rd International Conference on Communication Software and Networks. – 2011. – P. 584 – 587.
7. Аничкин С.А. Протоколы информационно-вычислительных сетей / С.А. Аничкин, С.А. Белов, А.В. Бернштейн; под ред. И.А. Мизина и А.П. Кулешова. – М.: Радио и связь, 1990. – 504 с.
8. Winkelman R. An educator's Guide to School Networks / Winkelman R. – Florida Center for Instructional Technology, College of Education, University of South Florida, 2009. – P. 1 – 15.
9. Core TCP/IP protocols / L. Parziale, N. Rosselot, W. Liu [et al.] // International Technical Support Organization, IBM. – 2016. – P. 4 – 9.
10. Сидельников В.М. Криптография и теория кодирования / В.М. Сидельников // Материалы конф. «Московский университет и развитие криптографии в России». – М.: МГУ, 2002. – 22 с.
11. Мельник А.П. Динамічні схеми захисту інформації на кодах Ріда-Соломона / А.П. Мельник, В.І. Грабчак // Тези доповідей II міжнар. НПК «Безпека та захист інформації в інформаційних системах». – 2009. – Вип. 7. – С. 139 – 140.
12. Congestion Control using FEC for Conversational Multimedia Communication / M. Nagy, V. Singh, J. Ott [et al.] // MMSys '14 Proceedings of the 5th ACM Multimedia Systems Conference. – 2014. – P. 191 – 202.
13. Ferlin S. TCP with dynamic FEC For High Delay and Lossy Networks / S. Ferlin, O. Alay // International Conference on emerging Networking Experiments and Technologies. – 2016. – P. 1 – 3.
14. MacKay D.J.C. Fountain codes / D.J.C. MacKay // Capacity approaching codes design and implementation, EE Proc.-Commun. – 2005. – Vol. 152, N 6. – P. 1062 – 1068.
15. Marcotte R.J. AprilFEC: Real-Time Channel Estimation and Adaptive Forward Error Correction / R.J. Marcotte, W. Xipeng, E. Olson // Robot Communication in the Wild 2017. – 2017. – P. 68.
16. Lundqvist H. TCP with End-to-End FEC / H. Lundqvist, G. Karlsson // Communications, International Zurich Seminar. – 2004. – P. 152 – 155.

Стаття надійшла до редакції 10.04.2018