

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет комп'ютерної інженерії та управління
Кафедра ЕОМ

Магістерська кваліфікаційна робота

Методи підвищення пропускної здатності каналів
комп'ютерних мереж

Виконав: ст. гр. КСМм-23-1 Сокирко М.А.

Керівник: доц. каф. ЕОМ Янковський О.А.

ВСТУП

Сучасні додатки, такі як віртуальна реальність, розподілена обробка даних або хмарні технології, потребують зв'язку з високою пропускною здатністю і малою затримкою. Ці додатки повинні обмінюватися даними через безліч мереж з різноманітними пропускними здатностями та кількістю переданого трафіку. Контроль за перевантаженням мережі, контроль швидкості і балансування навантаження, ось деякі з основних підходів до зменшення затримки та збільшення пропускної здатності різних мереж.

TCP – це протокол транспортного рівня, який зараз використовується в комп'ютерних мережах. Це надійний і впорядкований мережвий протокол, орієнтований на підключення. Згідно зі статистикою, більшість трафіку даних у сучасному Інтернеті передається саме через TCP.

Таким чином, якість протоколу передачі TCP має великий вплив на загальну продуктивність мережі. Дослідження показують, що перевантаження мережі є однією з основних причин, які впливають на ефективність передачі TCP.

МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ

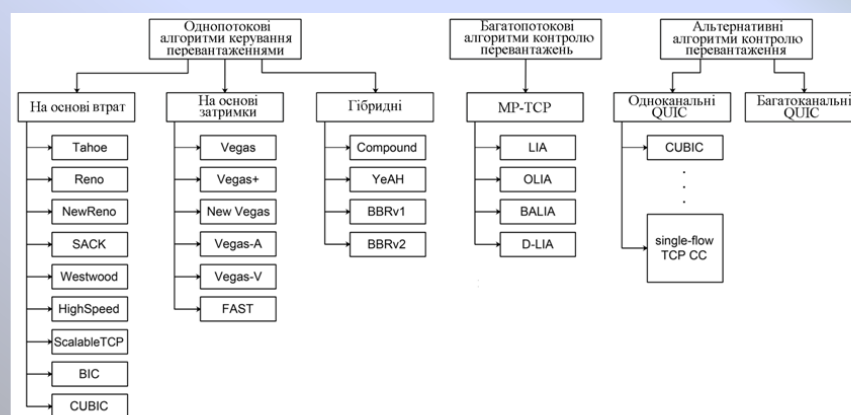
Протокол управління передачею ТСР відіграє вирішальну роль в інфраструктурі сучасних мереж. Надаючи орієнтовану на з'єднання, надійну службу потоку, ТСР дозволив Інтернету процвітати в багатьох типах мереж від високошвидкісних, високонадійних волоконно-оптичних з'єднань до безпроводних каналів, схильних до помилок.

Незважаючи на свою довгу історію, контроль перевантажень ТСР залишається гарячою темою донині, оскільки ТСР досі залишається єдиним стандартним мережевим протоколом, здатним забезпечити наскрізну надійну передачу та контролювати перевантаження.

Метою магістерської кваліфікаційної роботи є розробка новітніх методів контролю перевантаження алгоритмами протоколу ТСР та поліпшення характеристик функціонування комп'ютерних мереж у сенсі зниження ймовірності виникнення перевантажень в умовах інтенсивного трафіку. Необхідно розглянути поточні та запропонувати нові методи, засновані на затримці, з метою вдосконалення існуючих наскрізних алгоритмів контролю перевантажень на основі втрат пакетів.

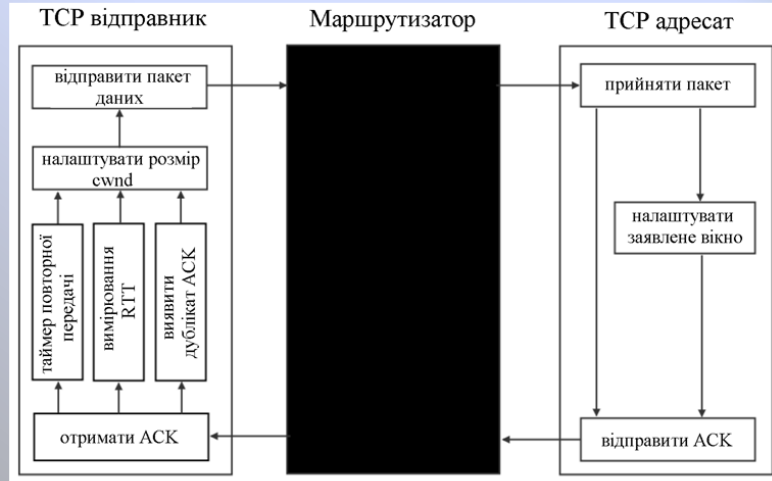
3

ОСНОВНІ КАТЕГОРІЇ ТА ТИПИ АЛГОРИТМІВ ТСР



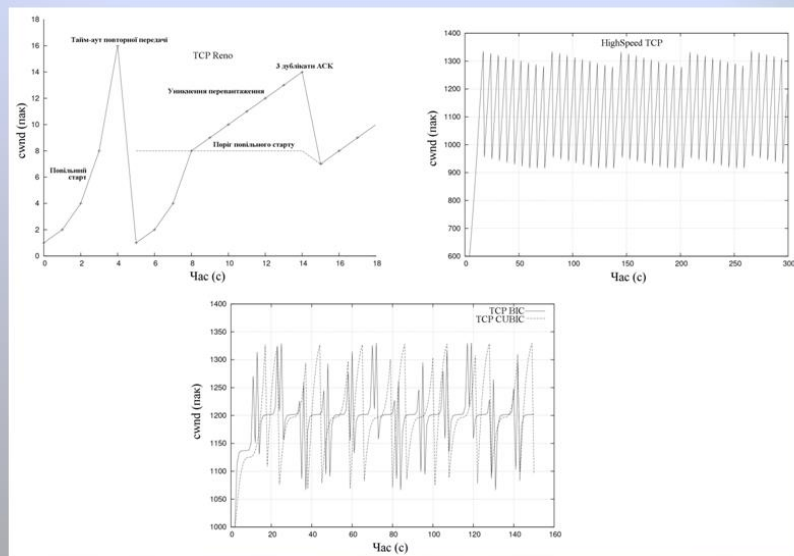
4

АРХІТЕКТУРА НАСКРІЗНОГО КОНТРОЛЮ ПЕРЕВАНТАЖЕНЬ TCP



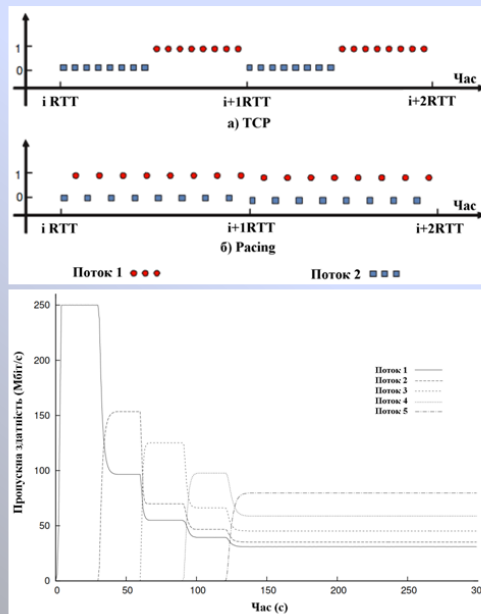
10

АЛГОРИТМИ КОНТРОЛЮ ПЕРЕВАНТАЖЕННЯ НА ОСНОВІ ВТРАТ



11

ПРОБЛЕМИ АЛГОРИТМІВ НА ОСНОВІ ЗАТРИМКИ



7

ПРОПОНОВАНІ МЕТОДИ

Підводячи підсумок, пропонується використати наступні методи:

- механізм пейсингу для покращення синхронізації потоку
- простий механізм синхронізації на основі затримки для покращення справедливості потоків
- Min CUBIC метод з використанням мінімальної затримки для оцінки затримки передачі
- Max CUBIC метод з використанням максимальної затримки, де використовується максимальне значення RTT, усуваючи необхідність оцінювати затримку передачі
- зменшення вікна перевантаження на основі порогового значення для покращення масштабованості

8

АЛГОРИТМ ВИЯВЛЕННЯ ПОТОКІВ НА ОСНОВІ ВТРАТ

Лістинг 6.1 – Псевдокод алгоритму виявлення потоків на основі втрат

```

Ініціалізація:
increasedCwnd ← true stablePhase ← false

Для кожного АСК:
if cwnd > prevMaxCwnd and stablePhase = false then
  /* досягнення стаціонарного стану */
  stablePhase ← true
end if

При збільшенні cwnd:
increasedCwnd ← true

При зменшенні cwnd:
if increasedCwnd = false and stablePhase = true then
  /* виявлено виснаження - не зменшувати cwnd і перехід у режим
на основі втрат */
  mode ← lossBased
  stablePhase ← false
else
  /* зменшити cwnd */
  increasedCwnd ← false
end if

```

9

MAX CUBIC НА ОСНОВІ МАКСИМАЛЬНОЇ ЗАТРИМКИ

Лістинг 6.2 – Псевдокод алгоритму Max CUBIC

```

Ініціалізація:
δ ← 500 /* період очікування в мілісекундах */
reduce ← false /* Якщо істина, то зменшити cwnd */
threshold ← 100 /* Попіг затримки в мілісекундах */
Для кожного АСК:
if now > waitTime then
  if reduce then
    cwnd ← β · cwnd /* β - параметр зменшення */
    ssthresh ← cwnd
    waitTime = now + δ
    reduce ← false
  else if rtt > maxRTT - threshold then
    reduce ← true
    waitTime ← now + δ
  else
    /* збільшити вікно перевантаження за правилами CUBIC */
  end if
else
  /* залишити cwnd без змін */
end if

При втраті пакета:
cwnd ← β · cwnd /* β - параметр зменшення */
ssthresh ← cwnd

```

10

ГІБРИДНИЙ АЛГОРИТМ MAX CUBIC/NEWRENO

Лістинг 6.3 – Псевдокод алгоритму MAX CUBIC/NEWRENO

```

Для кожного ACK:
if rtt > maxRTT - threshold then
  cwnd ← cwnd + 1/cwnd /* NewReno правило */
else
  /* збільшити вікно перевантаження за правилами CUBIC */
end if

При втраті пакету:
cwnd ← β*cwnd /* β є параметром зменшення */
ssthresh ← cwnd

```

MAX CUBIC ІЗ ПОКРАЩЕНОЮ КОРОТКОСТРОКОВОЮ СПРАВЕДЛИВІСТЮ

Лістинг 6.4 – Псевдокод алгоритму Max CUBIC із покращеною короткостроковою справедливістю

```

Ініціалізація:
sprobe ← 10 /* Час очікування зондування мережі в секундах */
probeChangeTimer ← 0 /* Таймер переходу стану зондування */
probeEndTimer ← 0 /* Таймер відключення зондування */

Для кожного ACK:
/* Зміна на орижованого на втрату, якщо максимальної RTT */
if rtt > maxRTT then
  maxRTT ← rtt
  if probe ≠ on or rtt > prevMaxRTT then
    probe ← on
    probeChangeTimer ← now + sprobe /* оцінити стан мережі за sprobe
    секунда */
    prevMaxRTT ← rtt
  end if
end if

/* В режимі втрати, поки не закінчиться таймер зондування */
if probe = on then
  if now < probeChangeTimer then
    cwnd ← cwnd + 1/cwnd /* NewReno increase rule */
  else
    probe ← freezing
    probeEndTimer ← now + sprobe
  end if
end if

/* Виявити фактичну оцінку стану мережі та перевірити, чи можна
перейти в повільний режим на основі затримки */
if probe = freezing or probe = decreasing then
  probe ← off
  if rtt < maxRTT - threshold then
    /* на цьому моменту алгоритм CUBIC з максимальною затримкою */
    else if now > probeChangeTimer then
      if probe = freezing then
        probe ← decreasing
        probeChangeTimer ← now + sprobe
      else /* probe = decreasing */
        cwnd ← cwnd * decreaseFactor
        probeChangeTimer ← now + sprobe
      end if
    else
      /* зменшити cwnd незмінно */
    end if
  end if
else
  /* дозволити алгоритму максимальної затримки CUBIC */
end if

```

MIN CUBIC НА ОСНОВІ МІНІМАЛЬНОЇ ЗАТРИМКИ

Лістинг 6.5 – Псевдокод алгоритму Min CUBIC з мінімальною затримкою

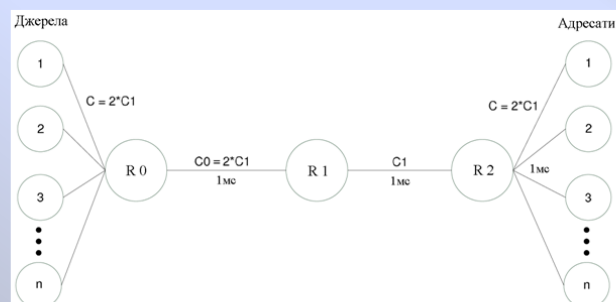
```

Ініціалізація:
δ - 500 /* Період очікування, у мілісекундах */
reduce - false /* Якщо істина, то зменшити cwnd */
threshold - 100 /* Попіг затримки в мілісекундах */
Для кожного ACK:
baseRTT - min(baseRTT, rtt)
β - 0.8*baseRTT/rtt /* β, розраховане для того, щоб очистити чергу */
if now > waitTime then
  if reduce then
    cwnd - β*cwnd /* β - параметр зменшення */
    ssthresh - cwnd
    waitTime = now + δ
    reduce - false
  else if rtt - baseRTT > threshold then
    reduce - true
    waitTime - now + δ
  else
    /* збільшити вікно перевантаження за допомогою правила CUBIC */
  end if
else
  /* залишити cwnd без змін */
end if

При втраті пакету:
cwnd - β*cwnd /* β - параметр зменшення */
ssthresh - cwnd
  
```

13

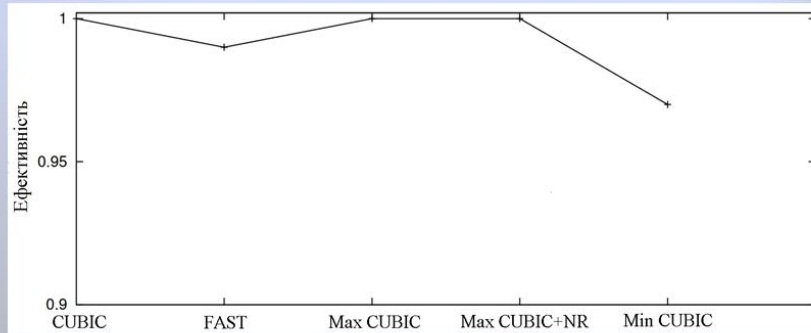
ІМІТАЦІЙНА МОДЕЛЬ ТА ПОЧАТКОВІ ПАРАМЕТРИ



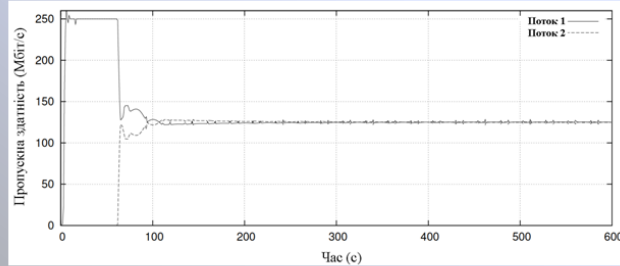
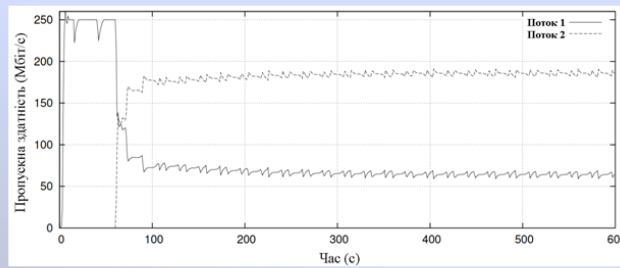
Параметри		Події
Джерел	1-50	t = 0 с старт 1 джерела
C0 (Мбіт/с)	500-2000	t = 300-3600 с час симуляції
C1 (Мбіт/с)	250-1000	
RTT (мс)	25, 160	
Довжина черги	100% BDP	
MSS	1460 Байт	

14

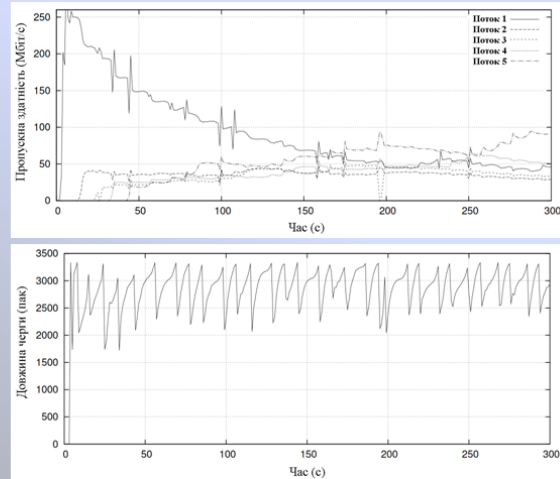
РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ



РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

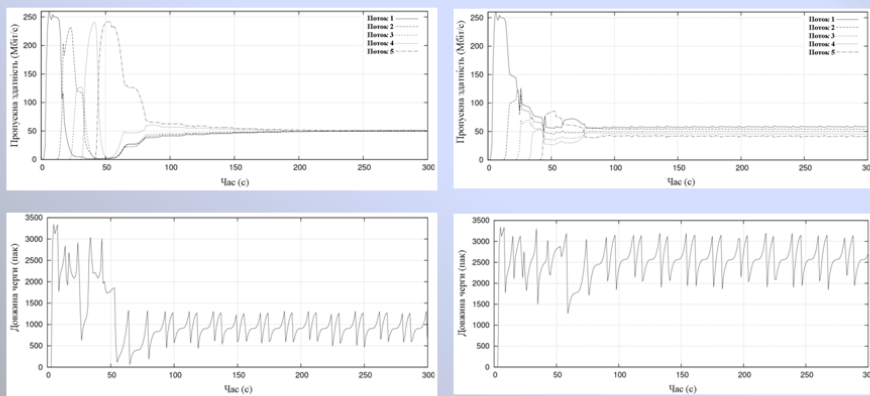


РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ



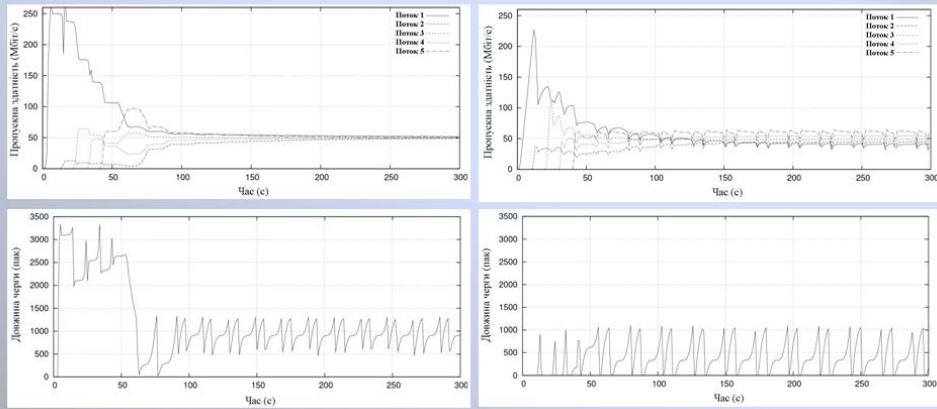
17

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ



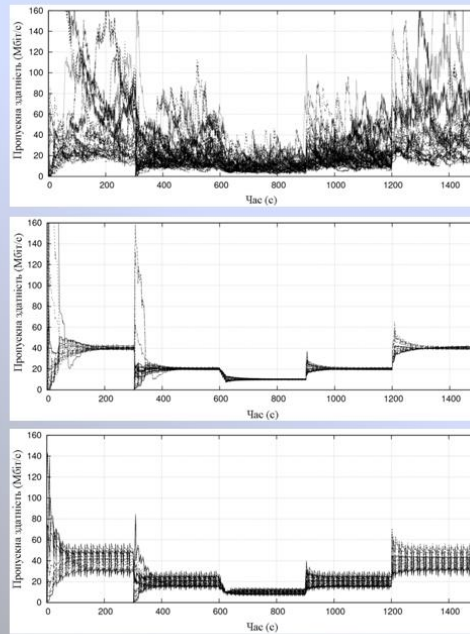
18

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ



19

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ



20

АПРОБАЦІЯ РЕЗУЛЬТАТІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ОБОРОНИ
АЗЕРБАЙДЖАНСЬКОЇ РЕСПУБЛІКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНИКИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ М. С. ЖУКОВСЬКОГО
"ХАРКІВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ"
УНІВЕРСИТЕТ МІСТА ЖИЛІНА

СУЧАСНІ НАПРЯМИ РОЗВИТКУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЗАСОБІВ УПРАВЛІННЯ

Тези доповідей чотирнадцятої міжнародної
науково-технічної конференції
25 – 26 квітня 2024 року
Том 1: секції 1, 2

Баку – Харків – Жиліна – 2024

Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління

УПРАВЛІННЯ МЕРЕЖЕВИМ ТРАФІКОМ

Созирає М. А., Чернов Д. Д., Порошенко А. І.
Харківський національний університет радіоелектроніки, Харків, Україна

Управління мережевим трафіком – це керування трафіком для досягнення заданих встановленої мети, наприклад, балансування навантаження на канали зв'язку, мінімізація наскрипної мережевої затримки тощо [1].

Сучасні комп'ютерні мережі та моделі трафіку є динамічними. Мережі можуть змінюватися з часом, тому, що можуть з'являтися нові канали зв'язку або елементи, як частина цієї системи. Деякі проблеми можуть виникати через зміни окремих її частин, наприклад швидкого підключення маршрутизатора [2].

Попит на трафік у сучасних системах є великим та швидко змінюється, тому через таку високу динамічність, методи управління трафіком мають бути гнучкими та швидко адаптуватися заради досягнення кращої продуктивності [3].

Якщо буде надіряне навантаження, то це може призвести до розриву і спливу, різних помилок при передачі даних та збільшення затримок в мережі.

Метою доповіді є розгляд проблем розробки методів управління мережевим трафіком, які є оптимальними з точки зору різних рівнів затримки та раціонального використання мережевих ресурсів.

В доповіді розкрито важливість ефективного управління трафіком у сучасних комп'ютерних мережах.

Запропоновано модифікований алгоритм управління трафіком, який має на меті забезпечити виконання наступних задач: забезпечення високої продуктивності, низької затримки та швидкої реакції на різноманітні зміни стану мережі.

Показано, що використання методів управління трафіком значно знижує навантаження на вузли та канали зв'язку, що в кінцевому підсумку підвищує загальну пропускну здатність та стабільність, а також значно знижує мережеву затримку.

Список літератури

1. Prasad, R. S. and Dovrolis, C., "Measuring the Congestion Responsiveness of Internet Traffic," in the Proceedings of Passive and Active Measurement (PAM), 2007.
2. Lee, B. P., Bala, R. K., Jacob, L., Seth, W. K. G., and Kulkarni, A. L., "Avoiding Congestion Collapse on the Internet using TCP Tunneling," Computer Net., vol. 39, no. 5, pp. 207–219, 2002.
3. Mohammed I. Sakman and Bin Wang, "Boosting performance for software defined networks from traffic engineering perspective," Computer Communications, 167:55 – 62, 2021.

118

21

АПРОБАЦІЯ РЕЗУЛЬТАТІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Інститут систем управління
МНО Азербайджанської республіки
Національний технічний університет
"Харківський політехнічний інститут"

Харківський національний
університет радіоелектроніки
Національний аерокосмічний університет
імені М. С. Жуковського
"Харківський авіаційний інститут"
Університет технологій і гуманітарних наук
(М. Бельсько-Бяла, Польща)

ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ

Тези доповідей дванадцятої міжнародної
науково-технічної конференції
21 – 22 листопада 2024 року
Том 1: секції 1, 2, 3

Баку – Харків – Бельсько-Бяла – 2024

Problems of informatization, the twelfth international scientific and technical conference

МЕТОДИ АДАПТАЦІЇ ПРОТОКОЛУ TCP ДО ІНТЕРНЕТНОГО МЕРЕЖЕВОГО СТАНУ

Колпа М. В., Созирає М. А., Яковський О. А.
Харківський національний університет радіоелектроніки, Харків, Україна

Обсяг даних, що передається в мережі, постійно зростає і може викликати багато проблем, одні із яких виникають при передаванні мережі (Network Congestion). Перевантаження мережі – це стан мережі, при якому вузол або канал передають так багато даних, що це може погіршити якість мережевих послуг та призводити до затримки в чергах пристроїв, втрати пакетів даних і блокування нових існуючих [1].

Протокол керування передачею (TCP) розроблено для надійної передачі даних через Інтернет. На продуктивність TCP дуже сильно впливають його алгоритми контролю перевантаження, які обмежують обсяг трафіку, який відправник може передати на основі наскрипних оцінок доступності окремих мережі [2].

Протокол TCP, незважаючи на свою ефективність для забезпечення стабільності трафіку, має суттєві обмеження при роботі з програмами, які генерують величезну кількість пакетів трафіку [3]. Стандартні алгоритми TCP не дозволяють таким програмам ефективно використовувати пропускну здатність мережі, особливо на довгих маршрутах з великими RTT, що може призводити до неефективного використання мережевих ресурсів та знижує загальну продуктивність мережі.

Метою доповіді є аналіз різноманітних причин неефективної роботи TCP з пакетним трафіком та виявлення методів, які дозволять програмам бути ефективно використовувати мережеві ресурси. Проведено детальний аналіз роботи стандартних алгоритмів TCP з пакетним трафіком. Було виявлено, що основною причиною неефективності є негнучкість алгоритмів швидко адаптуватися до змін у поточному мережевому стані.

В доповіді розглянуті алгоритми, які дозволять таким програмам з пакетним трафіком швидко адаптуватися до змін мережевих умов та ефективно використовувати доступну пропускну здатність. Доповідь спрямована на вирішення проблеми та вивчення сучасних алгоритмів, що надає змогу визначити найбільш перспективні напрями їх розвитку.

Список літератури

1. Bivwak, Md Idrisli, "Internet congestion control for variable-rate TCP traffic," Thesis, University of Aberdeen, 2011.
2. Anil Agarwal, Stefan Savage, and Thomas Anderson, "Understanding the Performance of TCP Raising," March 30, 2000, IEEE InfoCom 2000.
3. Mohd Muzaffar Mohamad, Mubashir Ahmad, Md Anis Nizam, "Empirical evaluation of TCP congestion control mechanisms in short and long distance networks," Journal of Theoretical and Applied Information Technology 71(2):153-166, 2015.

72

22

ВИСНОВКИ

Загальновідомо, що алгоритми управління перевантаженням TCP погано працюють у мережах наступного покоління. Джерела TCP адаптують своє вікно перевантаження, використовуючи мережеві сигнали, наприклад, втрати пакетів і зміни затримки. Однак поява мереж наступного покоління показала обмеження консервативного підходу до контролю перевантажень TCP, створюючи кілька проблем, таких як зниження ефективності з'єднання в гетерогенних мережах.

У ході виконання кваліфікаційної роботи були представлені наскрізні методи управління перевантаженням в протоколі TCP, які можуть значно покращити високошвидкісний наскрізний алгоритм на основі втрат пакетів CUBIC.

Під час проведення імітаційного моделювання оцінювалася продуктивність різних алгоритмів, головним чином за показниками ефективності, справедливості, зайнятості черги і масштабованості. Отримані результати показують, що в змодельованих мережевих сценаріях алгоритми Max CUBIC з використанням максимальної затримки і Min CUBIC з використанням мінімальної затримки можуть значно підвищити продуктивність широко застосованого алгоритму CUBIC.