ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «**Методи управління потоком в мережах наступного покоління**»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Електронні комунікації та мережі
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Вадим ТЕРЕЩЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти
групи ТСД-63
Вадим ТЕРЕЩЕНКО

Керівник: Оксана ВАРФОЛОМЄЄВА

Рецензент: _____

Київ – 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій

Кафедра	Телекомунікаційних систем та мереж
Ступінь вищої освіти	Магістр
Спеціальність	172 Електронні комунікації та мережі
Освітньо-професійна програма	Телекомунікаційні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедри ТСМ
_____ Віктор ЗАЇКА

« » _____ 2025року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Терещенку Вадиму Олеговичу

1. Тема кваліфікаційної роботи: «Методи управління потоком в мережах наступного покоління», керівник кваліфікаційної роботи Оксана ВАРФОЛОМЄЄВА, к.т.н., доцент
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «_30_» _10_ 2025 р. № 467
2. Строк подання кваліфікаційної роботи «___» _____ 2025 р.
3. Вихідні дані до роботи:

1. Алгоритми управління потоком в мережах NGN за протоколом TCP/IP;
2. Методи керування “ ковзаючим вікном”.
3. Ефективність передачі інформації при різних алгоритмах управління перевантаженням.
4. Науково-технічна література.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Алгоритми управління потоком в мережах NGN за протоколом TCP/IP.
2. Дослідження впливу вибору алгоритмів за протоколом TCP/IP на ефективність передачі інформації в мережах NGN.
3. Вплив вибору алгоритму за протоколом TCP/IP на ефективність роботи мережі NGN.

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання « 09 » жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	10.10.25	Виконано
2	Архітектура мережі NGN	22.10.25	Виконано
3	Дослідження впливу вибору алгоритмів за протоколом TCP/IP на ефективність передачі інформації в мережах NGN.	08.11.25	Виконано
4	Вплив вибору алгоритма за протоколом TCP/IP на ефективність роботи мережі NGN.	28.11.25	Виконано
5	Висновки, вступ, реферат	08.12.25	Виконано
6	Розробка презентації	15.12.25	Виконано

Здобувач вищої освіти Вадим ТЕРЕЩЕНКО

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи Оксана ВАРФОЛОМЕЄВА

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Терещенко В.О. до захисту кваліфікаційної роботи
(прізвище та ініціали)

за спеціальністю 172 Електронні комунікації та мережі

освітньо-професійної програми Телекомунікаційні системи та мережі

на тему: « Методи управління потоком в мережах наступного покоління »

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Терещенко Вадим Олегович під час виконання магістерської роботи продемонстрував уміння самостійно вирішувати поставлені завдання, належний рівень теоретичної та практичної підготовки, а також навички роботи з сучасними комп'ютерними технологіями та науково-технічною літературою. У роботі досліджено методи управління потоком і перевантаженням за протоколом TCP/IP та їх застосування в мережах NGN, проаналізовано стек протоколів TCP/IP і його параметри, що впливають на пропускну здатність мережі. Розглянуто поведінку алгоритмів управління перевантаженням, математичну модель встановлення TCP-з'єднання та запропоновано метод оптимізації TCP-з'єднань. Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Терещенко Вадима Олеговича на оцінку "відмінно" та присвоїти йому кваліфікацію «магістр з електронних комунікацій та радіотехніки».

Керівник кваліфікаційної роботи _____ Оксана ВАРФОЛОМЕЄВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

" " _____ 2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Терещенко В.О.
(прізвище та ініціали)

допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою
Телекомунікаційних систем та мереж _____ Віктор ЗАЇКА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

" __ " ____ 20__ року

ВІДГУК РЕЦЕНЗЕНТА

На кваліфікаційну роботу

На здобуття освітнього ступеня магістр

здобувача вищої освіти Терещенка Вадима Олеговича

на тему: «Методи управління потоком в мережах наступного покоління»

Актуальність:

Одним із головних сучасних пріоритетів України є формування та розвиток національного сегмента інформаційного суспільства і входження нашої країни до світової інформаційної спільноти. Сучасні конвергентні мережі є надскладною інфраструктурою, конче потребують нових методів дослідження. Необхідно розвивати загальну теорію оптимальних методів обробки та передачі інформації, котрі сприяють поліпшенню показників якості передачі інформації в цифровій формі каналами зв'язку мереж NGN. Серед напрямків розвитку науки та техніки особливе значення мають інноваційні технології, які окреслюють напрями науково-технічного прогресу. Тому тема магістерської роботи, яка присвячена дослідженню методів управління потоком в мережах NGN за протоколом TCP/IP, є актуальною.

Позитивні сторони:

1. Зміст магістерської роботи відповідає завданню. Терещенко В.О. показав добрий рівень знань і ступінь підготовленості до майбутньої роботи за фахом.

2. Поставлені завдання в роботі з огляду, аналізу, дослідження виконано в повному обсязі. Достатньо успішно викладено технічні питання. Досліджено алгоритми, які використовуються в протоколі TCP та його різноманітні модифікації. Проведені розрахунки алгоритму запобігання перевантаженню , а саме параметрів, які впливають на пропускну здатність каналу і проведені між ними залежності.

3. Текст викладено грамотно, послідовно. Графічний матеріал оформлено якісно. Список науково-технічної літератури свідчить про вміння користуватись матеріалами за темою магістерської роботи.

Недоліки:

1. Доцільно було б конкретніше сформулювати яким чином вибираються різні модифікації алгоритмів для каналів різних типів.

2. Не сформульовано, який вплив на завадостійкість мережі NGN впливають окремі модифікації TCP/IP.

Висновки: Вказані недоліки не впливають на якість виконаної роботи, яка заслуговує оцінки «відмінно», а її автор Терещенко Вадим Олегович заслуговує на присвоєння кваліфікації магістр з електронних комунікацій та радіотехніки .

Рецензент:

_____	_____	_____
(науковий ступінь, вчене звання)	*(підпис)*	*(ім'я, прізвище)*

РЕФЕРАТ

Текстова частина кваліфікаційної магістерської роботи на здобуття освітного ступеня магістр : 113 сторінок, 33 рисунка, 2 таблиці, 18 джерел.

Мета роботи – Дослідження алгоритмів управління потоком в мережах NGN за протоколом TCP/IP, встановлення проблемних аспектів в управлінні потоком, визначення можливих методів покращення управлінням потоку в мережах NGN.

Об’єкт дослідження – Методи управління потоком інформації в мережах NGN.

Предмет дослідження – Алгоритми запобігання перевантаження по протоколу TCP/IP.

Короткий зміст роботи *:* В роботі приведено основні відомості про мережі NGN та виявлено тенденції їх сучасного розвитку. Проаналізований стек протоколів TCP/IP для мережі NGN, також проаналізовані самі ці протоколи та їх реалізація. Дослідженні механізми управління потоком та перевантаженням за протоколом TCP/IP. Проаналізовані параметри TCP/IP які впливають на пропускну здатність мережі NGN. Досліджено залежності часу кругового обходу (RTT) та частоти втрат від розміру вікна в базовому алгоритмі TCP в режимі виключення перевантаження. Було досліджено поведінку різних алгоритмів управління перевантаженням при різних значеннях затримки інформації в мережі та втратах. Проаналізована математична модель встановлення TCP-з’єднання. Та представлений метод оптимізації TCP-з’єднаннь.

КЛЮЧОВІ СЛОВА: NGN, TCP, IP, CC, CAA, RTT, ПОВІЛЬНИЙ СТАРТ, УПРАВЛІННЯ ПОТОКОМ, УПРАВЛІННЯ ПЕРЕВАНТАЖЕННЯМ.

ABSTRACT

The text part of the qualifying master's thesis for obtaining
The Master's degree: 113 pages, 33 figures, 2 tables, 18 sources

Purpose of the work – Study of flow-control algorithms in NGN networks using the TCP/IP protocol, identification of problematic aspects in flow control, determination of possible methods for improving flow control in NGN networks.

Object of research – Methods of information flow control in NGN networks. NGN.

Subject of research – Algorithms for preventing congestion under the TCP/IP protocol.

Brief summary of the work: The work presents basic information about NGN networks and identifies trends in their modern development. The TCP/IP protocol stack for an NGN network is analyzed; the protocols themselves are also analyzed, as well as their implementation. The mechanisms of flow control and congestion control under the TCP/IP protocol are studied. TCP/IP parameters that affect the throughput of the NGN network are analyzed. The dependencies of round-trip time (RTT) and loss rate on the window size in the basic TCP algorithm in the congestion-avoidance mode are investigated. The behavior of various congestion-control algorithms under different values of network delay and losses is studied. A mathematical model of establishing a TCP connection is analyzed. A method for optimizing a TCP connection is presented.

KEY WORDS: NGN, TCP, IP, CC, CAA, RTT, SLOW START, FLOW CONTROL, CONGESTION CONTROL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ (за наявності).....
ВСТУП……...10
1. ОСНОВНІ ПРИНЦИПИ ОРГАНІЗАЦІЇ МЕРЕЖ NGN…….....12
1.1. Передумови переходу до мереж NGN…….....12
1.2. Архітектура NGN……...……16
1.2.1. Архітектура за чотирьох рівнявою моделлю……16
1.2.2.Архітектура за концепцією ITU-T…….……17
1.2.3.Архітектура за концепцією 3GPP……...20
2. СТЕК ТА ХАРАКТЕРИСТИКА БАЗОВИХ ПРОТОКОЛІВ NGN……23
2.1. Визначення стека TCP/IP……...23
2.2. Структура стека TCP/IP……23
2.3. Протокол IP……28
2.3.1.Формат протоколу IP……....29
2.3.2.IP маршрутизація…….……33
2.4. Протокол TCP……35
2.4.1. Формат протоколу TCP……...37
2.4.2.Встановлення сесії TCP……...41
3. ДОСЛІДЖЕННЯ БАЗОВИХ МЕТОДІВ УПРАВЛІННЯ ПОТОКОМ ЗА ПРОТОКОЛОМ TCP/IP В МЕРЕЖАХ NGN……....47
3.1. Механізм управління потоком – метод ковзаючого вікна……47
3.2. Механізм управління потоком – повільний старт……...49
3.3. Алгоритм запобігання перевантаження AIMD……...52
3.4. Параметри які впливають на пропускну здатність протокола TCP……..55
3.5. “Вузькі місця” в мережі……59
3.6. Розрахунок часу затримки інформації в мережі за протоколом TCP/IP з урахуванням алгоритмів запобігання перевантаження……....64
3.6. Розрахунок залежності часу кругового обходу (RTT) та частоти втрат від вікна в базовому алгоритмі TCP в режимі виключення перевантаження……66
3.7 Математична модель встановлення TCP-з’єднання……67
4. ДОСЛІДЖЕННЯ МЕТОДІВ В АЛГОРИТМАХ УПРАВЛІННЯ ПОТОКОМ ЗА ПРОТОКОЛОМ TCP……..76
4.1. Алгоритм TCP Reno……...……76
4.2. Алгоритм TCP NewReno……78
4.3. Алгоритм TCP Vegas……..82
4.4. Алгоритм TCP Compound……..83
4.5. Алгоритм TCP Tahoe……..84
4.6. Алгоритм TCP Westwood……..88
4.7. Алгоритм TCP Hybla……...……...90
4.8. Алгоритми BIC-TCP та CUBIC-TCP……93
4.9. Алгоритми TCP RED\WRED……98
4.10. Метод оптимізації TCP-з’єднань……...106
ВИСНОВКИ…..……110
ПЕРЕЛІК ПОСИЛАНЬ……...……112
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) ……

ВСТУП

Сучасний розвиток інфокомунікаційних послуг відбувається в умовах стрімкого зростання обсягів передавання даних та підвищення вимог користувачів до якості й надійності телекомунікаційних сервісів. Переважна більшість таких послуг реалізується на основі глобальної мережі Інтернет, доступ до якої здійснюється через існуючі мережі зв'язку. Проте у низці практичних випадків традиційна транспортна інфраструктура Інтернету не забезпечує необхідних характеристик швидкодії, масштабованості та якості обслуговування, що обмежує можливості подальшого розвитку інформаційного суспільства.

За таких умов актуальним стає завдання підвищення ефективності управління можливостей телекомунікаційних мереж. Це, у свою чергу, стимулює процес інтеграції пакетних мереж передачі даних із класичними телекомунікаційними системами та сприяє впровадженню концепції мереж наступного покоління (Next Generation Network, NGN).

Мережі NGN являють собою мультисервісну телекомунікаційну інфраструктуру, побудовану на основі IP-технологій, яка забезпечує спільне передавання голосових, інформаційних та мультимедійних потоків. На відміну від традиційних мереж, у NGN чітко розмежовуються сигнальні процеси, що відповідають за встановлення та управління з'єднаннями, і передавання корисних даних користувача. Такий підхід дозволяє оптимізувати використання мережних ресурсів та підвищити гнучкість впровадження нових послуг.

Подальший розвиток концепції NGN зумовив появу різних архітектурних підходів, які в межах єдиної інфраструктури поєднують мережі загального користування, мобільні системи зв'язку та ресурси Інтернету. Перехід від класичних телекомунікаційних мереж до NGN супроводжується зміною функціональної моделі, відповідно до якої функції передавання даних, управління сигналізацією та надання сервісів розділяються між окремими рівнями. Це забезпечує можливість їх незалежного розвитку та стандартизації інтерфейсів взаємодії.

Ключову роль у функціонуванні мереж наступного покоління відіграє стек протоколів TCP/IP, який визначає принципи надійного передавання даних між кінцевими вузлами. Протокол TCP виконує завдання управління потоком та перевантаженням, адаптуючи швидкість передавання інформації до поточного стану мережі. Завдяки використанню механізмів підтвердження, повторної передачі та контролю порядку доставки пакетів TCP забезпечує цілісність і достовірність переданих даних.

Одним із базових механізмів TCP є управління потоком на основі змінного вікна, що дозволяє оптимізувати процес передавання пакетів без необхідності

підтвердження кожного з них окремо. Ефективність роботи мережі NGN значною мірою залежить від алгоритмів управління перевантаженням TCP, які враховують параметри часу кругового обходу, втрати пакетів та стан черг у вузлах мережі. Оскільки кожен алгоритм має власні особливості поведінки в різних умовах, універсального рішення для всіх сценаріїв експлуатації не існує.

У зв’язку з наведеним дослідження методів управління потоком у мережах наступного покоління за протоколом TCP/IP є актуальним та практично значущим завданням. Результати такого дослідження можуть бути використані для підвищення ефективності роботи телекомунікаційних мереж, оптимізації використання мережних ресурсів та покращення якості обслуговування користувачів.

1 ОСНОВНІ ПРИНЦИПИ ОРГАНІЗАЦІЇ МЕРЕЖ NGN

1.1 Передумови переходу до мереж NGN

Розвиток телекомунікаційної галузі протягом останніх десятиліть супроводжується переходом від класичних мереж електрозв'язку до глобальної інформаційної інфраструктури. Така трансформація зумовлена зростанням ролі інформації в сучасному суспільстві та потребою забезпечення швидкого, надійного й універсального доступу до інформаційних ресурсів незалежно від місця перебування користувача. У межах цієї концепції традиційні телекомунікаційні послуги поступово еволюціонували в інфокомунікаційні, що поєднують функції передачі, обробки, зберігання та надання інформації.

Глобальна інформаційна інфраструктура (Global Information Infrastructure, GII) розглядається як технологічна основа інформаційного суспільства та передбачає можливість недискримінаційного доступу до інформаційних ресурсів для широкого кола користувачів. До її складу входять мережі зв'язку, обчислювальні засоби, бази даних, а також абонентські термінали. Надання доступу до таких ресурсів здійснюється за допомогою інфокомунікаційних послуг, які, на відміну від традиційних, орієнтовані не лише на передачу сигналів, а й на підтримку складних інформаційних сервісів.

Сучасні користувачі отримали доступ до широкого спектра послуг, які ще відносно недавно вважалися інноваційними. Електронна пошта, доступ до мережі Інтернет, мобільний зв'язок і мультимедійні сервіси стали невід'ємною частиною повсякденного життя. При цьому зростають вимоги до мобільності, універсальності та персоналізації послуг. Користувачі прагнуть мати єдиний ідентифікатор, що дозволяє залишатися доступними незалежно від типу мережі або використовуваного терміналу.

Водночас розвиток телекомунікаційних мереж визначається не лише технічними можливостями, а й економічною доцільністю. Оператори зв'язку зацікавлені у зменшенні витрат на експлуатацію мереж, підвищенні гнучкості впровадження нових послуг і зниженні залежності від конкретних постачальників обладнання. Посилення конкуренції на ринку телекомунікаційних послуг,

зумовлене процесами дерегуляції та появою нових сервіс-провайдерів, додатково стимулює пошук ефективніших мережних рішень.

Інфокомунікаційні послуги мають низку характерних особливостей, які відрізняють їх від послуг традиційних мереж зв'язку. Вони, як правило, реалізуються на верхніх рівнях еталонної моделі взаємодії відкритих систем, передбачають використання клієнт-серверної архітектури та орієнтовані на передавання мультимедійної інформації. Такі послуги часто потребують високої пропускної здатності, асиметрії інформаційних потоків і підтримки складних конфігурацій з'єднань, зокрема багатоточкових.

Додатковою особливістю інфокомунікаційних сервісів є різноманітність прикладних протоколів і можливість управління послугами безпосередньо з боку користувача. Для ідентифікації абонентів можуть застосовуватися додаткові механізми адресації, специфічні для окремих сервісів. При цьому функціональність таких послуг розподіляється між обладнанням постачальника та кінцевими пристроями користувача, що потребує врахування ролі абонентського обладнання під час проектування та регламентації послуг.

Зміна характеру телекомунікаційних послуг призвела до трансформації бізнес-моделі їх надання. Окрім традиційних учасників — оператора та абонента — з'явилися постачальники послуг, інформаційні провайдери, брокери та посередники, які забезпечують адаптацію сервісів до потреб конкретних користувачів. У такій моделі постачальники послуг використовують транспортні ресурси операторів мереж загального користування, що потребує забезпечення відкритого та недискримінаційного доступу до мережної інфраструктури.

З огляду на зазначене до перспективних мереж зв'язку висуваються підвищені вимоги, серед яких мультисервісність, широкосмуговість, підтримка мультимедійних застосувань, інтелектуальні механізми управління та інваріантність доступу. Крім того, важливим є забезпечення багатооператорського середовища, у межах якого декілька операторів можуть брати участь у наданні однієї послуги з чітким розмежуванням відповідальності.

Існуючі мережі з комутацією каналів і пакетів не повною мірою відповідають зазначеним вимогам, що обмежує можливості впровадження нових інфокомунікаційних сервісів і негативно впливає на показники якості обслуговування. Це зумовлює необхідність пошуку нових підходів до розвитку телекомунікаційної інфраструктури, одним із яких стала концепція мереж наступного покоління NGN. Основним принципом цієї концепції є розділення функцій передачі, управління викликами та надання послуг, що створює умови для гнучкого та масштабованого розвитку сучасних телекомунікаційних мереж.

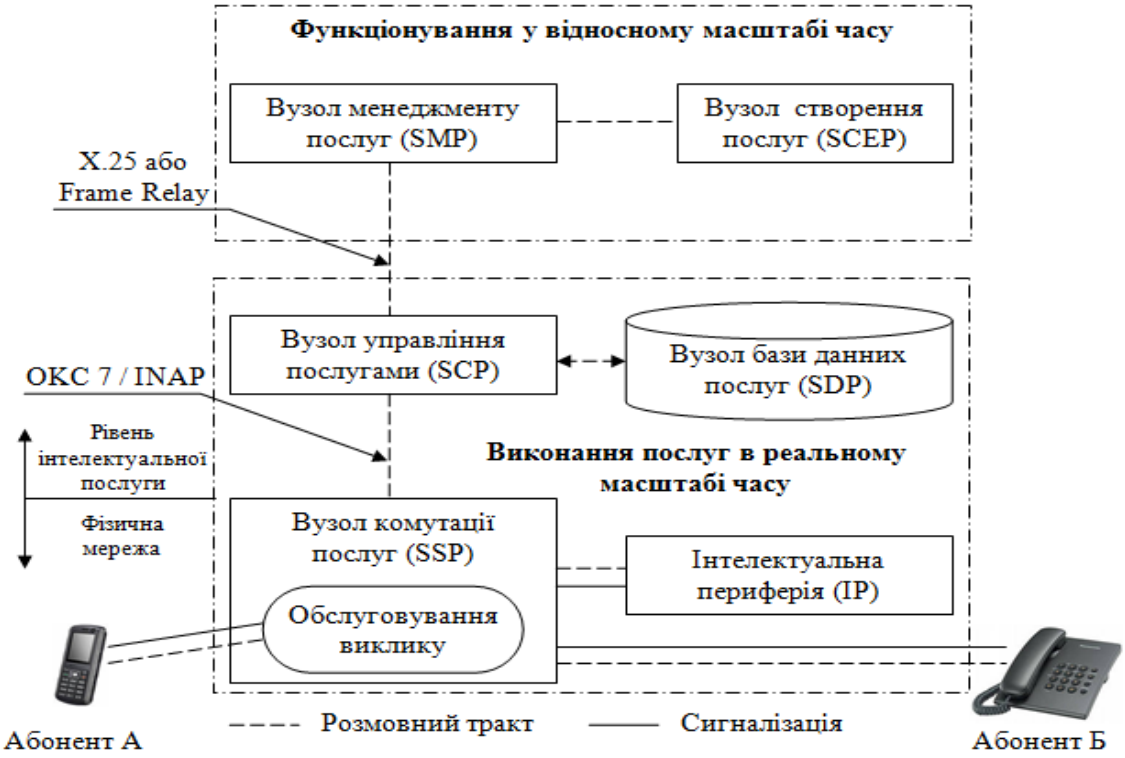


Рис.1.1 Платформа Інтелектуальної мережі

Схематично процедура обробки виклику в інтелектуальній мережі (IN) реалізується таким чином: комутаційний вузол (SSP) приймає виклик і виконує його первинну обробку. У разі виникнення події, що потребує застосування додаткової логіки послуги (тригера), SSP призупиняє стандартний сценарій обробки та надсилає запит до вузла керування послугами (SCP). SCP, використовуючи правила та дані сервісу, формує відповідь з інструкціями

(наприклад, щодо маршрутизації або обмеження з’єднання), після чого SSP продовжує виконання виклику відповідно до отриманих вказівок.

Загальний процес встановлення та завершення з’єднання можна представити як послідовність станів (ініціація виклику, аналіз номера, вибір маршруту, встановлення з’єднання, завершення). Запити до сервісної логіки можуть бути інтегровані в окремі моменти цього процесу, що дозволяє реалізовувати додаткові послуги без повної зміни базових процедур комутації.

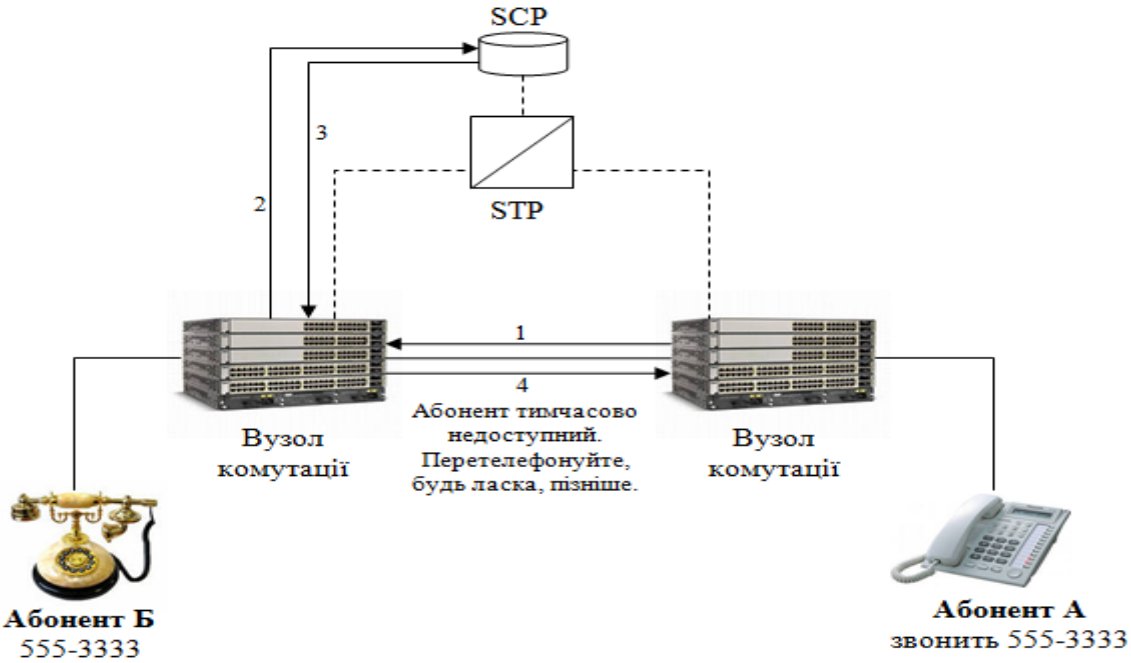


Рис. 1.2 Приклад реалізації сервісу "Не турбувати".

Наведений приклад демонструє підхід інтелектуальної мережі (IN) до реалізації додаткових сервісів шляхом винесення логіки послуги на спеціалізовані керуючі вузли. Водночас практика експлуатації IN виявила низку обмежень: функціональність і перелік послуг значною мірою залежали від прикладних протоколів та реалізації інтерфейсів, а впровадження нових сервісів нерідко потребувало модернізації мережних елементів керування та комутації. Додатковими проблемами були складність забезпечення сумісності рішень різних

виробників, обмежені можливості залучення сторонніх постачальників послуг і висока вартість розгортання та супроводу інфраструктури.

Подальший розвиток інфокомунікаційних послуг і IP-технологій зумовив перехід до концепції мереж наступного покоління (NGN), у якій ключовим є розділення функцій транспортування, керування з'єднаннями та надання послуг. Такий підхід забезпечує гнучкість, масштабованість і можливість швидшого впровадження нових сервісів без суттєвого впливу на транспортну інфраструктуру, що є важливим для сучасних телекомунікаційних мереж.

1.2 Архітектура NGN

1.2.1 Архітектура за чотирьох рівнявою моделлю

В даний час найбільшого поширення набула чотирьохрівнева (площинна) архітектура NGN:

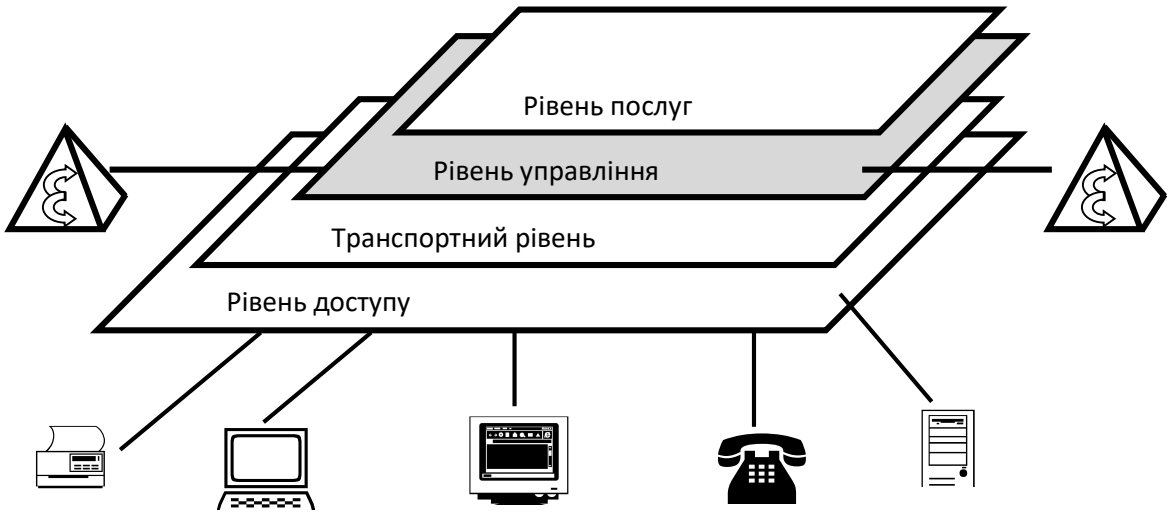


Рис.1.3 Архітектура мережі наступного покоління

Архітектура мереж наступного покоління (NGN) ґрунтується на функціональному розподілі компонентів, що дозволяє відокремити передачу даних, керування

з’єднаннями та надання послуг. Чотирирівнева модель є поширеним підходом до побудови NGN, оскільки спрощує модернізацію мережі та підтримує мультисервісність.

Рівень доступу забезпечує підключення абонентів і формування трафіку, транспортний рівень відповідає за перенесення інформаційних потоків, рівень керування реалізує встановлення/підтримку сесій та політики керування, а рівень послуг забезпечує роботу прикладних сервісів. Розмежування рівнів дає змогу незалежно розвивати транспорт і сервіси, що є важливим для ефективної роботи NGN у умовах зростання навантаження.

1.2.2 Архітектура за концепцією ITU-T

Відповідно до рекомендацій ITU-T архітектура NGN базується на принципі розмежування транспортних і сервісних функцій. Це дозволяє розвивати транспортну інфраструктуру незалежно від прикладних сервісів та спрощує впровадження нових послуг без суттєвих змін у транспортному середовищі.

Для опису взаємодії між функціями транспорту й послуг застосовується горизонтальне та вертикальне функціональне розшарування. Горизонтальне розшарування відображає поділ на транспортний рівень і рівень послуг, а вертикальне — поділ кожного рівня на площини:
– площина користувача (передача корисних даних);
– площина керування (встановлення та підтримка з’єднань/сесій);
– площина управління (експлуатаційне та адміністративне керування).

Загальний принцип розподілу функціональності послуг і транспорту в NGN за концепцією ITU-T наведено на рис. 1.4

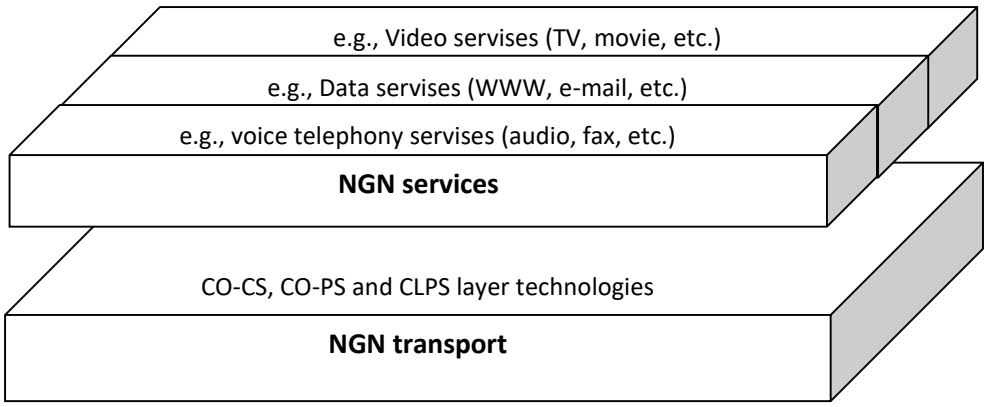


Рис. 1.4 Розподіл функціональності послуг і транспорту в NGN за концепцією ITU-T.

Як показано на рис. 1.4, в архітектурі мереж наступного покоління чітко розмежовуються функції транспорту та послуг, що забезпечує можливість їх незалежного розвитку. Такий підхід дозволяє реалізовувати прикладні сервіси без безпосереднього впливу на транспортну інфраструктуру, а також спрощує впровадження нових технологій передачі даних у межах єдиної мережі.

Транспортний рівень може включати набір різних мережних технологій, які відповідають нижнім рівням еталонної моделі взаємодії відкритих систем. Це забезпечує сумісність між мережами різної архітектури та створює умови для інтеграції існуючих і перспективних технологій у складі NGN. При цьому пакетна передача даних розглядається як основний спосіб транспортування інформаційних потоків.

Рівень послуг, у свою чергу, зосереджує функції реалізації прикладної логіки та управління сервісами. Він взаємодіє з транспортним рівнем через стандартизовані інтерфейси, що дозволяє розділити відповідальність між процесами передачі даних і управління послугами. Така взаємодія є ключовою умовою забезпечення гнучкості та масштабованості мереж наступного покоління.

Для більш детального опису внутрішньої організації кожного з рівнів у концепції ITU-T використовується багатошарова модель, яка передбачає наявність окремих

площин. Зокрема, виділяються площина користувача, площина керування та площина управління, що логічно присутні в межах як транспортного рівня, так і рівня послуг.

Площина користувача відповідає за безпосередню передачу інформаційних потоків між мережними вузлами. Площина керування забезпечує виконання процедур встановлення, підтримки та завершення з'єднань, а також обробку сигналізації. Площина управління реалізує функції експлуатаційного контролю, моніторингу стану мережі та адміністрування ресурсів.

Зазначений принцип багатоплощинної організації архітектури NGN детальніше ілюструється на рис. 1.5, де наведено базову еталонну модель мереж наступного покоління відповідно до рекомендацій ITU-T.

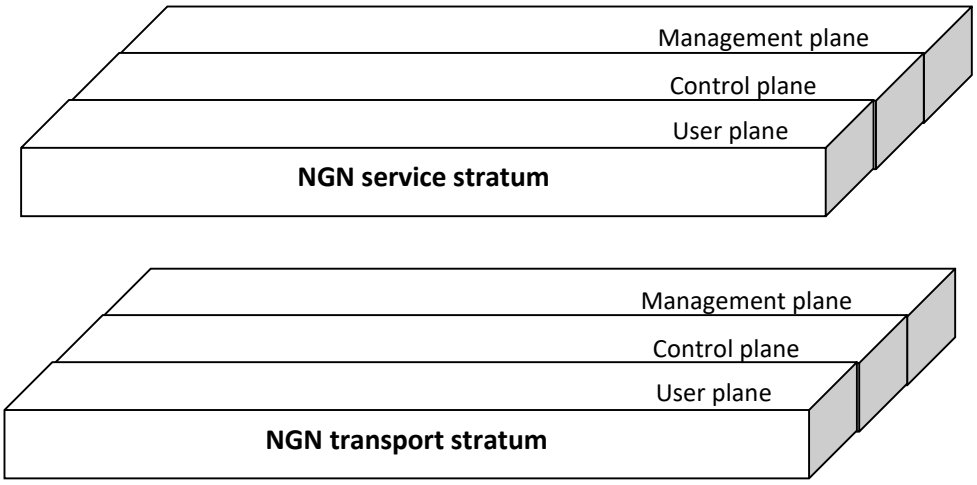


Рис. 1.5 Базова еталонна модель NGN

Базова еталонна модель NGN, наведена на рис. 1.5, відображає багаторівневу організацію мережі наступного покоління та взаємодію між основними функціональними площинами. У межах кожного рівня виділяються площина користувача, площина керування та площина управління, які відповідають за передачу даних, контроль з'єднань і експлуатаційне управління мережею відповідно.

Такий підхід забезпечує гнучкість архітектури NGN і дозволяє незалежно розвивати транспортні та сервісні функції, а також спрощує інтеграцію нових технологій і послуг у межах єдиної мережної інфраструктури.

1.2.3 Архітектура за концепцією 3GPP

Архітектура мереж наступного покоління за концепцією 3GPP базується на використанні підсистеми IMS (IP Multimedia Subsystem), яка призначена для організації мультимедійних сеансів (зокрема голосових) у пакетних мережах на основі IP. IMS формує уніфіковану сервісну платформу та забезпечує конвергенцію фіксованого й мобільного доступу, дозволяючи надавати широкий спектр інфокомунікаційних послуг у межах єдиної інфраструктури.

Сигналізаційна взаємодія в IMS, як правило, реалізується із застосуванням протоколу SIP, що забезпечує встановлення, модифікацію та завершення мультимедійних сеансів незалежно від технології доступу. Як транспортне середовище можуть використовуватись IP-мережі та IP/MPLS, що дає змогу ефективно передавати різнорідний трафік і масштабувати мережу під зростаючі навантаження.

Загальну структуру підсистеми IMS наведено на рис. 1.6

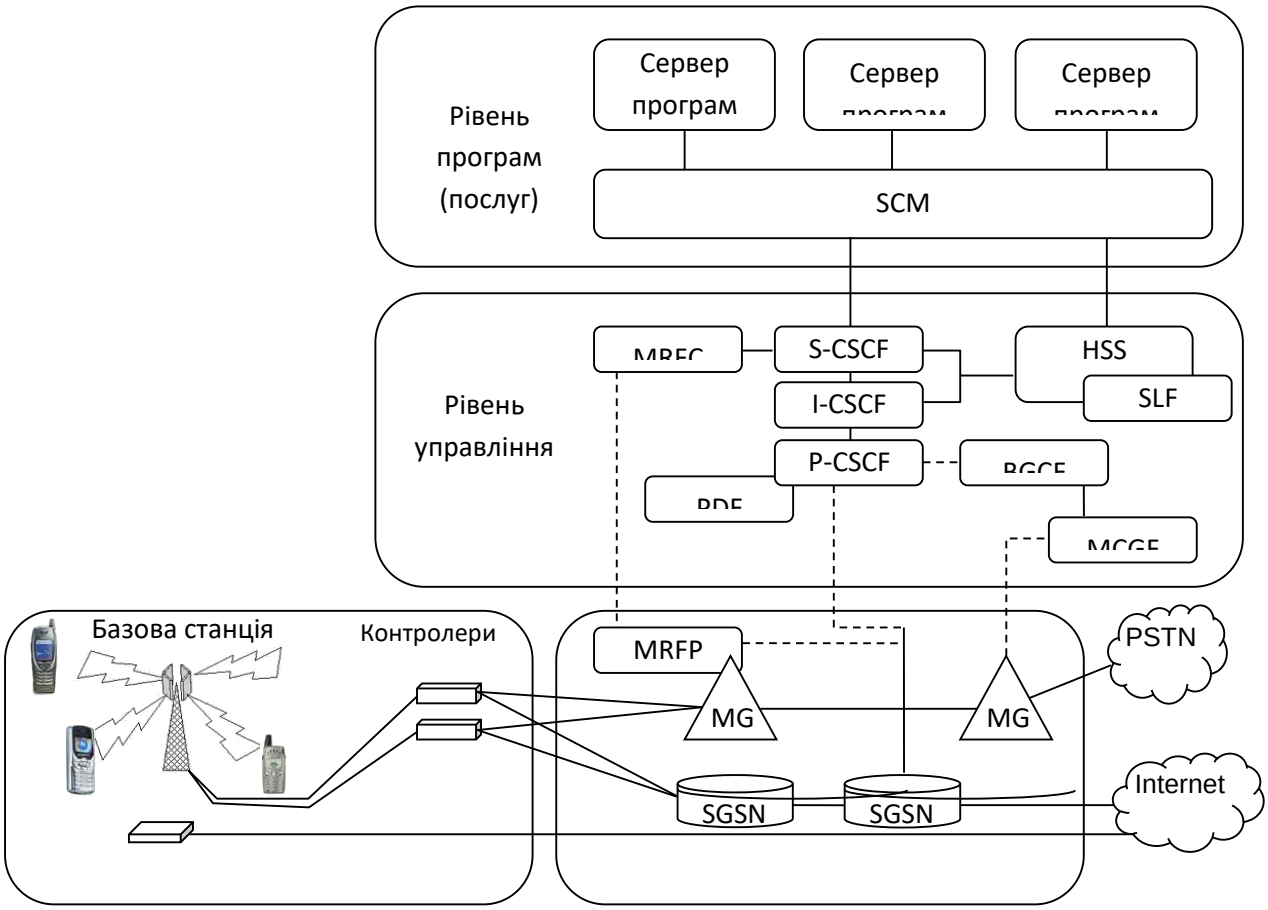


Рис. 1.6 Архітектура підсистеми IMS

Як показано на рис. 1.6, у межах IMS можна виділити три логічні групи функцій: рівень прикладних сервісів, рівень керування та транспортний рівень. Рівень прикладних сервісів представлений прикладними серверами (Application Servers), які реалізують логіку послуг та взаємодіють із сигналізаційними елементами IMS.

Рівень керування забезпечує обробку SIP-сигналізації, маршрутизацію сеансів і керування політиками обслуговування. До типових функціональних вузлів цього рівня належать CSCF-елементи (зокрема P-CSCF, I-CSCF, S-CSCF), а також абонентська база даних (HSS), що використовується для автентифікації та визначення профілю послуг користувача.

Транспортний рівень відповідає за передачу пакетного трафіку та медіапотоків у IP-мережі (зокрема на базі IP/MPLS), забезпечуючи масштабованість і ефективне

використання ресурсів. Для взаємодії з мережами загального користування (PSTN) та іншими зовнішніми системами застосовуються шлюзи (наприклад, медіашлюзи та сигнальні шлюзи), що дозволяє підтримувати сумісність і забезпечувати коректну маршрутизацію викликів між різними доменами.

Таким чином, IMS у рамках концепції 3GPP виступає універсальною платформою для побудови мультимедійних сервісів у NGN, забезпечуючи незалежний розвиток послуг і транспортної інфраструктури, а також інтеграцію різних типів мереж в єдиному сервісному середовищі.

2 СТЕК ТА ХАРАКТЕРИСТИКА БАЗОВИХ ПРОТОКОЛІВ NGN

2.1 Визначення стека TCP/IP

Стек протоколів TCP/IP є набором мережних протоколів, призначених для організації передавання даних у пакетних комп'ютерних мережах. Він забезпечує взаємодію між мережними вузлами незалежно від типу апаратного забезпечення та є базовою технологічною основою функціонування мережі Інтернет.

Стандартизація протоколів TCP/IP здійснюється організацією IETF, а їх специфікації публікуються у документах серії RFC. Багаторівнева структура стека дозволяє розділити функції маршрутизації, управління з'єднаннями та обробки даних, що забезпечує гнучкість, масштабованість і широке застосування TCP/IP у мережах наступного покоління.

2.2 Структура стека TCP/IP

Стек TCP/IP є практично орієнтованою багаторівневою моделлю протокольної взаємодії, яка використовується в пакетних мережах і лежить в основі роботи Інтернет та сучасних NGN. Поділ на рівні дозволяє розмежувати функції передавання даних, маршрутизації та керування з'єднаннями, що спрощує модернізацію мережі та забезпечує сумісність рішень.

Для порівняння принципів побудови протокольних моделей часто використовують еталонну модель OSI. Відповідність між рівнями OSI та рівнями TCP/IP є умовною, оскільки TCP/IP об'єднує частину функцій OSI у більш узагальнені рівні. Схематично таке зіставлення наведено на рис. 2.1

Модель OSI		Модель TCP/IP
Прикладна		Прикладна
Представницька		
Сеансова		
Транспортна		Транспортна
Мережева		Міжмережева
Канальна		Інтерфейсна
Фізична		

Рис. 2.1 Відповідність еталонних моделей OSI і TCP/IP.

Як видно з рис. 2.1, прикладний рівень TCP/IP охоплює функції, які в OSI рознесені між прикладним, представницьким і сеансовим рівнями. Транспортний рівень TCP/IP відповідає транспортному рівню OSI та забезпечує доставку даних між кінцевими вузлами з урахуванням вимог застосунку. Міжмережний рівень TCP/IP (IP) реалізує адресацію та маршрутизацію, а інтерфейсний рівень охоплює функції доступу до середовища передавання.

Для подальшого дослідження управління потоком ключовим є транспортний рівень, оскільки саме тут протокол TCP реалізує механізми керування вікном, контролю перевантаження та адаптації швидкості передавання до стану мережі.

Узагальнена структура стека TCP/IP та приклади протоколів, що належать до кожного рівня, наведені на рис. 2.2. Такий поділ є зручним для аналізу роботи мережі NGN та впливу параметрів транспортного рівня (зокрема TCP) на пропускну здатність і затримки.

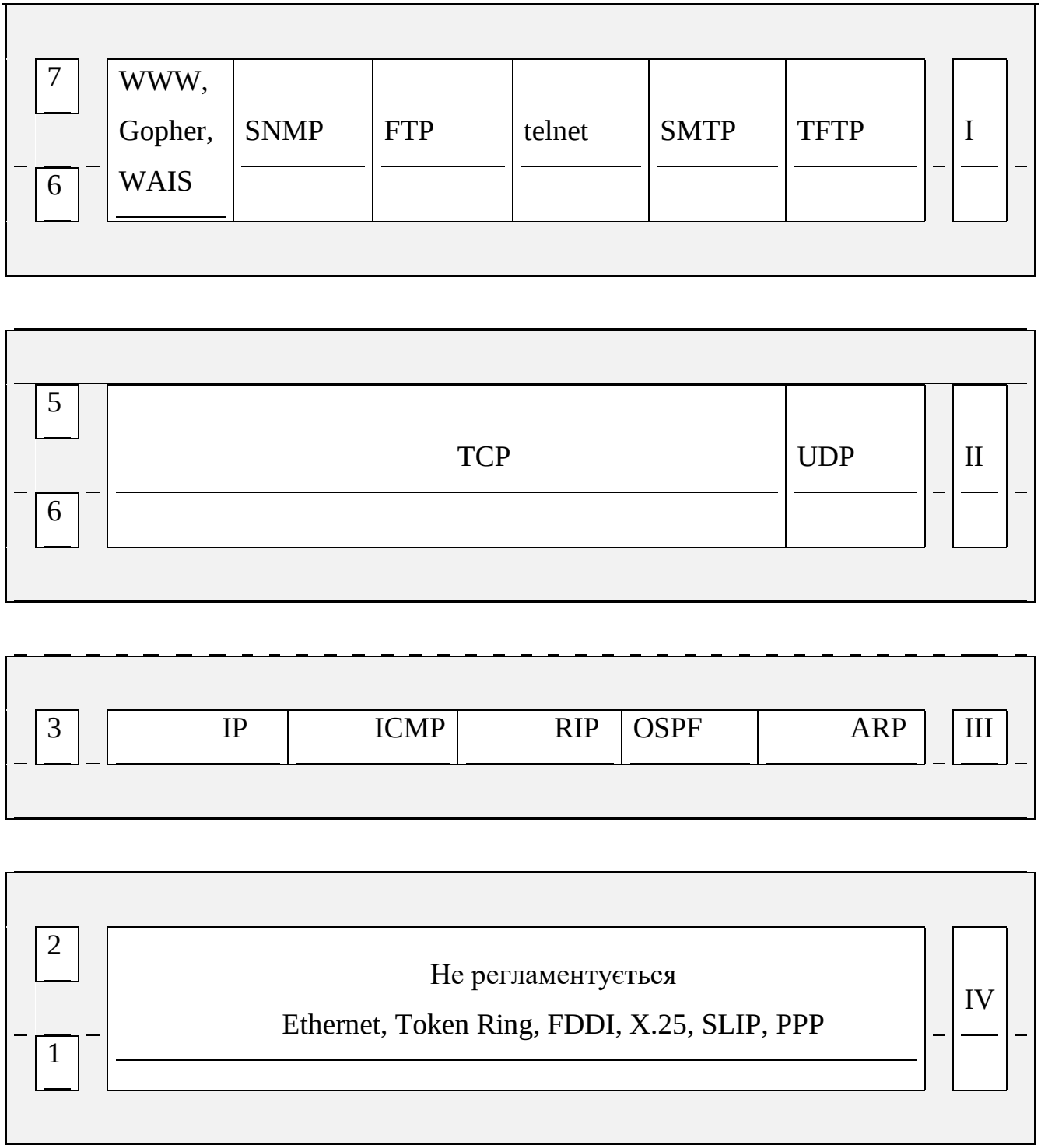


Рис. 2.2 Стек TCP/IP

Мережний рівень (network layer), який також називають рівнем міжмережної взаємодії, призначений для передавання пакетів у мережі та організації їх маршрутизації. Саме на цьому рівні визначається шлях проходження пакетів між вузлами мережі. У стеку протоколів TCP/IP функціонування мережного рівня забезпечують протоколи IP (Internet Protocol), ICMP (Internet Control Message Protocol) та IGMP (Internet Group Management Protocol). До цього рівня також належать протоколи, що використовуються для формування й оновлення таблиць маршрутизації, зокрема RIP (Routing Information Protocol) і OSPF (Open Shortest Path First).

Транспортний рівень (transport layer) забезпечує передавання потоку даних між двома кінцевими комп'ютерами та створює умови для роботи прикладного рівня. У межах сімейства протоколів TCP/IP на транспортному рівні використовуються два основні протоколи — TCP (Transmission Control Protocol) і UDP (User Datagram Protocol). Протокол TCP реалізує надійну передачу даних, забезпечуючи сегментацію інформації, підтвердження прийому пакетів, керування тайм-аутами та повторну передачу у разі втрат. Завдяки цьому механізми надійності не потребують реалізації на прикладному рівні.

На відміну від TCP, протокол UDP надає спрощений сервіс передачі даних, передаючи інформацію у вигляді дейтаграм без гарантії доставки або збереження порядку. У цьому випадку відповідальність за контроль надійності покладається на прикладний рівень. Кожен із транспортних протоколів використовується залежно від вимог конкретних застосувань і характеру передаваних даних.

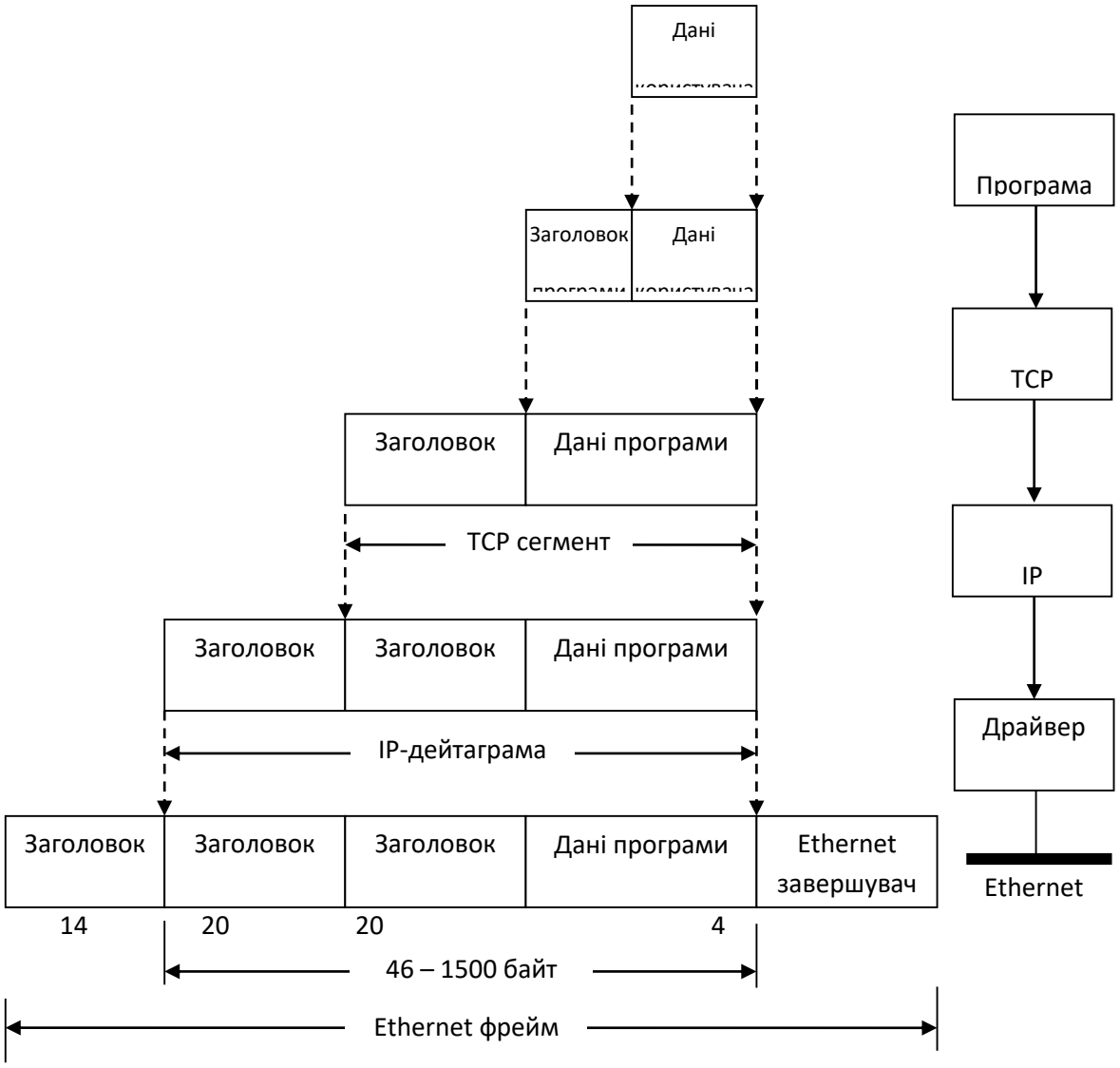


Рис. 2.3 Інкапсуляція даних по мірі того як вони проходять по стеку протоколів.

Прикладний рівень (application layer) визначає правила та механізми роботи конкретних мережних застосувань і забезпечує їх взаємодію з нижчими рівнями стека TCP/IP. Саме на цьому рівні реалізуються прикладні протоколи, які безпосередньо використовуються користувачами та програмними сервісами. До найбільш поширених прикладних протоколів TCP/IP, що присутні у більшості реалізацій, належать Telnet, який забезпечує віддалений термінальний доступ, FTP (File Transfer Protocol), призначений для передавання файлів, SMTP (Simple Mail Transfer Protocol), що використовується для пересилання електронної пошти,

а також SNMP (Simple Network Management Protocol), який застосовується для управління мережними пристроями.

Під час передавання даних прикладним застосуванням із використанням протоколу TCP інформація послідовно передається вниз по стеку протоколів, проходячи всі його рівні, доки не буде передана мережею у вигляді потоку бітів. На кожному рівні до даних додається службова інформація у вигляді відповідних заголовків, а в окремих випадках — і завершувальних полів. Такий процес поетапного додавання службової інформації називається інкапсуляцією та проілюстрований на рис. 2.3.

Блок даних, що формується на транспортному рівні та передається протоколом TCP до міжмережного рівня, називається TCP-сегментом. На рівні IP цей сегмент інкапсулюється в IP-дейтаграму, яка, у свою чергу, передається на канальний рівень. Під час передавання через мережі Ethernet IP-дейтаграми інкапсулюються в кадри (frames), які являють собою потік бітів, що безпосередньо передається фізичним середовищем.

2.3 Протокол IP

Протокол IP (Internet Protocol) належить до мережного рівня стека TCP/IP і призначений для забезпечення адресації та маршрутизації пакетів у мережі. Він містить службову інформацію, необхідну для передавання дейтаграм між вузлами об’єднаної мережі. Опис протоколу IP наведений у документі RFC 791, і він є базовим протоколом мережного рівня в сімействі Internet-протоколів. Разом із протоколом TCP протокол IP формує основу функціонування глобальної мережі Інтернет.

Основними функціями протоколу IP є організація передавання дейтаграм у режимі доставки без встановлення з’єднання, а також підтримка механізмів

фрагментації та повторної збірки пакетів. Фрагментація застосовується у випадках, коли розмір дейтаграми перевищує максимально допустимий розмір передаваного блока даних (MTU) для конкретного каналу зв’язку, що дозволяє забезпечити взаємодію мереж із різними характеристиками.

Протокол IP не гарантує надійної доставки даних і належить до класу протоколів без встановлення з’єднання. Це означає, що відсутні механізми підтвердження доставки, контролю порядку пакетів або повторної передачі у разі втрат. У разі виникнення помилкових ситуацій, зокрема перевантаження буферів маршрутизаторів, IP може відкинути дейтаграму та ініціювати надсилання повідомлення ICMP відправнику для інформування про помилку.

Характеристика «без з’єднання» також означає, що протокол IP не зберігає інформацію про стан передавання даних. Кожна дейтаграма обробляється незалежно від інших і може бути маршрутизована різними шляхами. Унаслідок цього можливі ситуації, коли дейтаграми, надіслані послідовно з одного джерела до одного пункту призначення, прибувають у зміненому порядку або з пошкодженнями. Забезпечення надійності передавання даних у таких випадках покладається на протоколи вищих рівнів, зокрема TCP.

2.3.1 Формат пакету IP

IP-пакет (IPv4) складається із заголовка та поля корисних даних. Заголовок містить службову інформацію, необхідну для адресації, маршрутизації та керування передаванням у мережі. Мінімальна довжина заголовка становить 20 байт, а за наявності опцій може збільшуватися. Загальну структуру заголовка IPv4 наведено на рис. 2.4.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															

Рис.2.4 Формат заголовку пакета IPv4

У заголовках TCP/IP багатобайтові числові поля передаються в мережному порядку байтів (big-endian), тобто від старшого байта до молодшого. Така домовленість є стандартною і забезпечує коректну інтерпретацію значень на різних обчислювальних системах.

Поле Version (версія), довжиною 4 біти, визначає версію протоколу IP, що використовується. У даному випадку розглядається протокол IPv4.

Поле IHL (Internet Header Length**)** також має довжину 4 біти та вказує розмір заголовка IP у 32-бітових словах. Стандартна довжина заголовка становить 20 байт, що відповідає п’яти 32-бітовим словам. За наявності додаткових параметрів у полі опцій довжина заголовка може збільшуватися, але не перевищує максимального значення 60 байт, тобто п’ятнадцяти 32-бітових слів. У заголовку IPv4 використовується 8-бітове поле, яке історично називалося Type of Service (ToS**)**, а в сучасному трактуванні відповідає Differentiated Services (DS) field. Основну частину DS field становить DSCP (Differentiated Services Code Point**)**, що

застосовується в механізмах DiffServ для класифікації та пріоритезації трафіку, зокрема сервісів реального часу (наприклад, VoIP). За наявності підтримки в мережі частина цього поля може використовуватися для сигналізації перевантаження (ECN). Структуру DS field наведено на рис. 2.5.

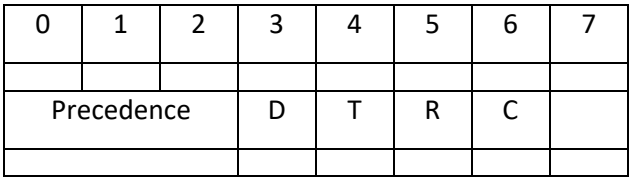


Рис. 2.5. Поле ToS/DSCP у заголовку IPv4.

Далі в заголовку IP розташовані чотири біти, які визначають параметри типу обслуговування. Із цих чотирьох бітів у кожному пакеті може бути встановлений лише один. Вони відповідають таким характеристикам трафіку: мінімальна затримка (Low Delay, D), висока пропускна спроможність (High Throughput, T), підвищена надійність (High Reliability, R) та низька вартість передавання (Low Cost, C). Останній із зазначених бітів був введений після публікації RFC 791 і на сьогодні практично не використовується.

Поле Type of Service (ToS) у заголовку IPv4 історично використовувалося для позначення вимог трафіку до мережі. У класичному поданні ToS містить біти, що відображають пріоритет та бажану характеристику обслуговування: мінімальна затримка (D), висока пропускна спроможність (T), підвищена надійність (R) і низька вартість (C) .

На практиці у кожному пакеті, як правило, активується не більше одного з цих бітів.

Приклади рекомендованого встановлення бітів ToS для типових мережних застосувань наведено в табл. 2.1

Таблиця 2.1

Рекомендовані значення поля ToS (D, T, R, C) для типових застосувань.

Застосування	**Мінімізація затримки**	**Максимізація продуктивності**	**Максимізація надійності**	**Мінімізація вартості**
Telnet/Rlogin	1	0	0	0
FTP				
управління	1	0	0	0
дані	0	1	0	0
SMTP				
фаза команд	1	0	0	0
фаза даних	0	1	0	0
DNS				
UDP запит	1	0	0	0
TCP запит	0	0	0	0
SNMP	0	0	1	0
NNTP	0	0	0	1

З наведених прикладів видно, що для інтерактивних застосувань (наприклад, Telnet/Rlogin) пріоритетною є мінімізація затримки, оскільки вони працюють у режимі реального часу та передають невеликі обсяги даних.

Для передавання великих масивів даних (FTP, фаза даних SMTP) доцільною є орієнтація на максимізацію пропускної спроможності.

Підвищена надійність є важливою для задач мережного керування (зокрема SNMP), де критичною є коректність доставки службової інформації.

Слід зазначити, що наведені значення мають переважно довідковий (історичний) характер і використовуються для ілюстрації підходів до класифікації трафіку.

2.3.2 IP маршрутизація

Розглянемо загальні принципи IP-маршрутизації. В IP-мережах кожен мережний інтерфейс має унікальну IP-адресу. При цьому IP-адреса ідентифікує не сам пристрій, а його підключення до мережі. Наприклад, вузол, що має декілька мережних інтерфейсів, буде мати відповідну кількість IP-адрес. Схема адресації протоколу IPv4 визначена в документах RFC 990 та RFC 997.

IP-адреса має довжину 32 біти і для зручності подається у вигляді десятково-точкової нотації, за якої кожен байт записується у вигляді десяткового числа в діапазоні від 0 до 255, а октети розділяються крапками (наприклад, 192.168.0.1).

Відповідно до принципів роботи стека TCP/IP, рівень IP може отримувати блоки даних від протоколів верхніх рівнів, зокрема TCP, UDP, ICMP та IGMP, які формують дейтаграми безпосередньо на даному вузлі. У такому випадку IP-дейтаграма створюється та передається до нижчого рівня — мережного інтерфейсу — для подальшого надсилання каналом зв’язку. Крім того, IP-дейтаграми можуть бути отримані з одного з мережних інтерфейсів вузла.

Якщо мережний вузол є джерелом інформації і пункт призначення безпосередньо підключений до нього, наприклад через канал типу «точка–точка», або якщо вузол підключений до кількох мереж, IP-дейтаграма передається безпосередньо адресату. В іншому випадку дейтаграма надсилається маршрутизатору за умовчанням, якому надається право визначати подальший шлях доставки пакета. Основна відмінність між хостом і маршрутизатором полягає в тому, що хост не

виконує пересилання дейтаграм між своїми інтерфейсами, тоді як маршрутизатор здійснює таке перенаправлення.

У разі отримання дейтаграми з мережного інтерфейсу виконуються такі дії. По-перше, IP-рівень перевіряє, чи належить вказана IP-адреса призначення даному вузлу або чи є вона широкомовною. Якщо ця умова виконується, дейтаграма передається відповідному протоколу верхнього рівня, зазначеному в полі Protocol заголовка IP. По-друге, якщо дейтаграма не призначена для даного вузла і IP-рівень не налаштований для роботи як маршрутизатор, пакет знищується. По-третє, якщо IP-рівень функціонує як маршрутизатор, дейтаграма перенаправляється далі відповідно до таблиці маршрутизації.

IP-рівень, налаштований як маршрутизатор, зберігає в пам’яті таблицю маршрутизації, яка аналізується кожного разу, коли необхідно переслати дейтаграму. Кожен запис цієї таблиці містить IP-адресу призначення, яка може бути як адресою окремого хоста, так і адресою мережі. Адреса хоста має ненульовий ідентифікатор хоста та відповідає конкретному вузлу, тоді як адреса мережі містить нульове значення ідентифікатора хоста і вказує на всі вузли, що належать до певної мережі.

Крім того, запис таблиці маршрутизації містить IP-адресу маршрутизатора наступної пересилки або адресу безпосередньо підключеної мережі. Маршрутизатор наступної пересилки не є кінцевим пунктом призначення, але приймає дейтаграми та пересилає їх у напрямку адресата. Також у таблиці зберігаються прапори, які визначають, чи є адреса призначення адресою мережі або хоста, а також чи є вузол наступної пересилки маршрутизатором або безпосередньо підключеним інтерфейсом. Окремо вказується мережний інтерфейс, через який має бути переданий пакет.

IP-маршрутизація реалізується за принципом «пересилка за пересилкою». Це означає, що протокол IP не володіє повною інформацією про весь маршрут до

пункту призначення, за винятком випадків безпосереднього підключення. У процесі маршрутизації визначається лише адреса маршрутизатора наступної пересилки, при цьому передбачається, що він знаходиться ближче до кінцевого пункту призначення, ніж поточний вузол.

Процес маршрутизації IP-дейтаграм виконується в кілька етапів. Спочатку в таблиці маршрутизації здійснюється пошук запису, який повністю відповідає IP-адресі призначення, тобто збігаються ідентифікатори мережі та хоста. Якщо такий запис знайдений, пакет надсилається на вказаний маршрутизатор наступної пересилки або безпосередньо підключений інтерфейс. Далі виконується пошук запису, що відповідає ідентифікатору мережі призначення з урахуванням маски підмережі. Якщо відповідний запис знайдено, пакет пересилається згідно з цим маршрутом. У разі відсутності таких записів використовується маршрут за умовчанням.

Якщо жоден із зазначених варіантів не дозволяє визначити шлях доставки, дейтаграма вважається недоставленою. У такому випадку вузол, що згенерував пакет, зазвичай отримує повідомлення про помилку типу «хост недоступний» або «мережа недоступна», яке передається відповідному застосуванню.

Важливою особливістю IP-маршрутизації є те, що спочатку завжди перевіряється відповідність повній IP-адресі хоста, потім — ідентифікатору мережі, і лише після цього використовується маршрут за умовчанням. Використання маршрутів за умовчанням, а також повідомлень ICMP про перенаправлення є ключовими елементами механізму IP-маршрутизації.

Ще однією фундаментальною властивістю IP-маршрутизації є можливість визначення маршрутів до цілих мереж замість окремих хостів. Саме завдяки цьому таблиці маршрутизації вузлів мережі Internet містять тисячі записів, а не мільйони, що забезпечує масштабованість глобальних мереж.

2.4 Протокол TCP

Протокол керування передаванням (Transmission Control Protocol, TCP) належить до транспортного рівня стека TCP/IP і забезпечує надійне передавання даних між кінцевими вузлами мережі. Він є протоколом, орієнтованим на встановлення з'єднання, тому перед початком обміну даними між клієнтом і сервером необхідно виконати процедуру встановлення з'єднання. Опис принципів роботи протоколу TCP наведено в документі RFC 793.

Надійність передавання даних у TCP досягається завдяки використанню комплексу спеціальних механізмів. Дані, що надходять від прикладного рівня, поділяються на блоки визначеного розміру, які передаються мережею у вигляді TCP-сегментів. На відміну від протоколу UDP, у якому кожен запис застосування безпосередньо формує IP-дейтаграму відповідного розміру, TCP виконує власну сегментацію даних.

Під час передавання сегмента TCP встановлює таймер очікування підтвердження від віддаленої сторони з'єднання. Якщо підтвердження не надходить протягом визначеного інтервалу часу, сегмент передається повторно. У разі приймання даних TCP формує підтвердження їх отримання, яке, як правило, надсилається з незначною затримкою з метою оптимізації передавання.

Для виявлення помилок TCP обчислює контрольну суму, яка охоплює як заголовок сегмента, так і поле даних. Контрольна сума розраховується на обох кінцях з'єднання і дозволяє виявити спотворення інформації під час передавання. Якщо сегмент надходить із некоректною контрольною сумою, він відкидається, а підтвердження не генерується, що призводить до повторної передачі сегмента відправником після спливу тайм-ауту.

Оскільки TCP-сегменти передаються у складі IP-дейтаграм, які можуть надходити до пункту призначення в довільному порядку, TCP забезпечує відновлення правильної послідовності даних перед їх передаванням прикладному рівню. Крім

того, протокол TCP здійснює виявлення та відкидання дубльованих сегментів, що можуть виникати в процесі передавання.

Важливою функцією TCP є контроль потоку даних. Кожна сторона TCP-з'єднання має буфер обмеженого розміру, і приймальна сторона регулює швидкість передавання даних, повідомляючи відправника про доступний обсяг буфера. Такий механізм запобігає переповненню буферів менш продуктивних вузлів і забезпечує стабільну роботу з'єднання.

2.4.1 Формат протоколу TCP

Структуру заголовка TCP-сегмента наведено на рис. 2.6. Заголовок містить набір полів, які забезпечують ідентифікацію з'єднання та керування процесом передавання даних.

Поля порт джерела (Source Port) і порт одержувача (Destination Port), кожне довжиною 16 біт, використовуються для ідентифікації прикладних процесів, що взаємодіють за допомогою протоколу TCP. У поєднанні з IP-адресами джерела та призначення ці значення забезпечують унікальну ідентифікацію кожного TCP-з'єднання. Комбінація IP-адреси та номера порту часто називається сокетом.

Поля порядковий номер (Sequence Number) і номер підтвердження (Acknowledgement Number), довжиною по 32 біти, призначені для нумерації переданих і прийнятих байтів даних. Значення цих полів реалізовані у вигляді беззнакових цілих чисел і циклічно оновлюються після досягнення максимального значення. Кожна сторона TCP-з'єднання веде власну послідовність нумерації, що дозволяє забезпечити правильний порядок доставлення даних.

Поле довжина заголовка (Offset) має довжину 4 біти та визначає розмір TCP-заголовка в 32-бітових словах. Наявність цього поля обумовлена тим, що поле

параметрів може мати змінну довжину. Фактично воно задає зсув від початку TCP-сегмента до початку поля даних.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															

Рис. 2.6 Формат заголовка пакету TCP

Після основних полів заголовка розташоване зарезервоване поле (Resrvd) довжиною 6 біт, яке на даний час не використовується та зарезервоване для можливого розширення протоколу в майбутньому.

Поле прапорів має довжину 9 біт і містить набір однобітових ознак, що керують роботою TCP-з’єднання. Прапор NS використовується в механізмах захисту ECN та пов’язаний із приховуванням одноразових значень. Прапор CWR (Congestion Window Reduced) встановлюється відправником для інформування про зменшення розміру вікна у відповідь на перевантаження мережі. Прапор ECE (ECN Echo) застосовується для сигналізації про наявність ознак перевантаження.

У разі встановлення прапора SYN значення поля ECE використовується для вказівки підтримки механізмів ECN. Якщо прапор SYN не встановлений, ECE сигналізує про те, що пакет був отриманий з ознаками перевантаження, визначеними на основі інформації з IP-заголовка.

Прапор URG встановлюється у випадку використання поля покажчика на термінові дані та вказує на наявність даних, що потребують негайної обробки.

Прапор ACK встановлюється тоді, коли поле номера підтвердження містить коректні значення, і в іншому випадку ігнорується.

Прапор PSH означає, що приймальна сторона повинна негайно передати отримані дані прикладному рівню без очікування заповнення буфера. У деяких сучасних реалізаціях TCP цей прапор може не враховуватися під час обробки сегментів. Прапор RST використовується для аварійного скидання з'єднання у разі помилок, отримання некоректних сегментів або спроби встановлення з'єднання за відсутності відповідного сервісу.

Прапор SYN застосовується під час ініціації TCP-з'єднання та використовується для синхронізації початкових порядкових номерів. Прапор FIN сигналізує про завершення передавання даних і використовується для коректного розриву з'єднання.

Керування потоком у протоколі TCP реалізується за допомогою механізму ковзного вікна змінного розміру. Поле розмір вікна (Window) довжиною 16 біт визначає кількість байтів, які можуть бути передані після байта, отримання якого вже підтверджене. Нульове значення цього поля означає, що приймальна сторона тимчасово не готова приймати додаткові дані. Відновлення передавання можливе після надсилання сегмента з ненульовим значенням розміру вікна.

Поле контрольної суми TCP (Checksum) має довжину 16 біт і використовується для перевірки цілісності всього TCP-сегмента. Контрольна сума обчислюється з урахуванням заголовка, даних та спеціального псевдозаголовка, структура якого наведена на рис. 2.7. Під час обчислення значення цього поля тимчасово встановлюється в нуль, а поле даних вирівнюється нульовими байтами.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source address																															
4	32	Destination address																															
8	64	0	0	0	0	0	0	0	0	Protocol								TCP length															
12	96	Source port																Destination port															
16	128	Sequence number																															
20	160	Acknowledgment number																															
24	192	Data offset				Reserved								Flags								Window											
28	224	Checksum																Urgent pointer															
32	256	Options (optional)																															
	288	Data																															

Рис. 2.7 Структура пакету TCP при обчисленні контрольної суми

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	IP-адрес джерела (Source address)																															
4	32	IP-адрес одержувача (Destination address)																															
8	64	0	0	0	0	0	0	0	0	Протокол (Protocol)								Длина TCP-сегмента (TCP length)															

Рис. 2.8 Структура псевдозаголовка пакету TCP

На початку псевдозаголовка розміщуються поля IP-адреси джерела та IP-адреси одержувача, кожне довжиною 32 біти. За ними слідує зарезервоване поле довжиною 8 біт, яке заповнюється нульовими значеннями. Поле Protocol довжиною 8 біт використовується для ідентифікації протоколу верхнього рівня, що інкапсулюється в IP-пакеті; для протоколу TCP його значення дорівнює 6. Далі розташоване поле довжини TCP, яке має довжину 16 біт.

Поле покажчика на термінові дані має довжину 16 біт і визначає зсув у байтах від поточного порядкового номера до позиції термінових даних у сегменті. Обробка таких даних здійснюється протоколами та застосуваннями верхніх рівнів.

Поле параметрів має змінну довжину, кратну 32 бітам, і використовується для розширення функціональних можливостей стандартного заголовка TCP. Це поле є необов’язковим і може бути відсутнім у заголовку сегмента.

Поле даних TCP-сегмента може бути як заповненим, так і порожнім, оскільки характер і формат переданої інформації визначаються виключно прикладним застосуванням. За відсутності параметрів максимальний теоретичний розмір цього поля становить 65495 байт.

Максимальний розмір сегмента (Maximum Segment Size, MSS) визначає найбільший обсяг даних, який TCP може передати віддаленій стороні в одному сегменті. За замовчуванням значення MSS дорівнює 536 байтам, що відповідає розміру IP-дейтаграми 576 байт за умови використання 20-байтного IP-заголовка і 20-байтного TCP-заголовка.

У загальному випадку збільшення значення MSS є доцільним доти, доки не відбувається фрагментація пакетів. Великі розміри сегментів дозволяють передавати більший обсяг даних у кожному сегменті, зменшуючи відносні накладні витрати на заголовки IP і TCP. Під час надсилання SYN-сегмента або ініціації з’єднання значення MSS може встановлюватися рівним значенню MTU вихідного інтерфейсу за вирахуванням розміру фіксованих заголовків TCP та IP. Так, для мереж Ethernet максимальне значення MSS може досягати 1460 байт, тоді як при використанні інкапсуляції за стандартом IEEE 802.3 воно становить до 1452 байт.

2.4.2 Встановлення сесії TCP

Поля порядковий номер (Sequence Number) та номер підтвердження (Acknowledgment Number) виконують функцію лічильників під час обміну даними між сторонами TCP-з’єднання. На етапі встановлення сесії їх значення використовуються спільно з полями прапорів TCP-заголовка.

Встановлення з’єднання між клієнтом і сервером у протоколі TCP здійснюється за допомогою триступінчатої процедури, відомої як триетапне рукостискання (three-

way handshake), що проілюстровано на рис. 2.9. Розглянемо цей процес на прикладі ініціації з'єднання між хостами A та B.

На першому етапі, який називається активним відкриттям (Active Open), хост A ініціює з'єднання, надсилаючи сегмент із встановленим прапором SYN. У полі порядкового номера при цьому задається початкове 32-бітове значення ISN_a (Initial Sequence Number), яке визначає стартове значення лічильника для передавання даних від хоста A.

Другий етап відповідає пасивному відкриттю (Passive Open). Хост B, отримавши SYN-сегмент, надсилає у відповідь сегмент з установленими прапорами SYN та ACK. У полі порядкового номера встановлюється початкове значення ISN_β, а поле номера підтвердження містить значення ISN_a, збільшене на одиницю. Таким чином сервер підтверджує отримання запиту на встановлення з'єднання та повідомляє власний початковий порядковий номер.

На третьому етапі клієнт завершує процедуру рукостискання, надсилаючи сегмент із встановленим прапором ACK. У цьому сегменті поле порядкового номера містить значення $ISN_a + 1$, а поле номера підтвердження — $ISN_\beta + 1$. Надсилання цього сегмента означає завершення триетапного рукостискання, після чого TCP-з'єднання вважається встановленим.

Після успішного встановлення з'єднання клієнт може передавати сегменти з даними серверу через створений віртуальний TCP-канал, використовуючи узгоджені значення порядкових номерів і підтверджень.

З наведеної схеми випливає, що основними параметрами, які ідентифікують TCP-абонентів і конкретне TCP-з'єднання, є саме 32-бітові значення полів порядкового номера та номера підтвердження. Вони забезпечують синхронізацію обміну даними, контроль доставки та коректний порядок передавання інформації.

Як приклад встановлення TCP-з’єднання можна розглянути ситуацію, у якій ініціююча сторона (клієнт) надсилає SYN-сегмент, указуючи номер порту сервера та власний початковий порядковий номер, після чого між сторонами виконується описана вище процедура рукостискання.

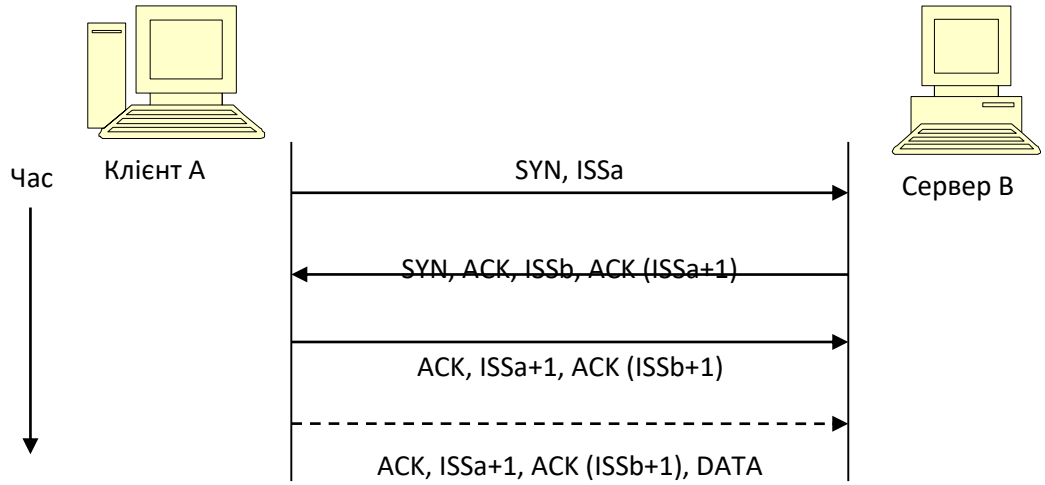


Рис. 2.9 Триступінчате рукостискання (three-way handshake)

У відповідь на початковий SYN-сегмент клієнта сервер надсилає власний SYN-сегмент, у якому міститься початковий порядковий номер сервера. Одночасно сервер підтверджує отримання SYN-сегмента клієнта, встановлюючи прапор ACK і передаючи значення номера підтвердження, що дорівнює початковому порядковому номеру клієнта, збільшеному на одиницю. Слід зазначити, що SYN-сегмент використовує один порядковий номер.

Після цього клієнт надсилає підтвердження отримання SYN-сегмента від сервера, встановлюючи прапор ACK і передаючи значення номера підтвердження, що дорівнює початковому порядковому номеру сервера, збільшеному на одиницю. Надсилання цього третього сегмента завершує процедуру триетапного рукостискання, і TCP-з’єднання вважається встановленим.

Сторона, яка першою надсилає SYN-сегмент, вважається ініціатором з’єднання та здійснює активне відкриття. Інша сторона, яка приймає цей сегмент і відповідає власним SYN-сегментом, бере участь у пасивному відкритті з’єднання.

Під час встановлення TCP-з’єднання кожна зі сторін обирає власний початковий порядковий номер (ISN). Значення ISN повинно змінюватися для кожного нового з’єднання, що забезпечує їх унікальність. Згідно з рекомендаціями RFC 793, ISN є 32-бітовим лічильником, який збільшується з часом. Використання порядкових номерів дозволяє уникнути ситуацій, коли пакети, що затрималися в мережі, помилково сприймаються як частина нового з’єднання.

Для встановлення TCP-з’єднання достатньо обміну трьома сегментами, тоді як для його коректного завершення зазвичай необхідно чотири сегменти. Це пов’язано з тим, що TCP-з’єднання може перебувати у напівзакритому стані. Оскільки TCP є повнодуплексним протоколом і передавання даних у кожному напрямі здійснюється незалежно, закриття кожного з напрямів виконується окремо.

Загальне правило полягає в тому, що кожна сторона повинна надіслати сегмент із встановленим прапором FIN після завершення передавання даних. Отримавши FIN-сегмент, протокол TCP інформує прикладне застосування про те, що віддалена сторона припинила передавання даних у відповідному напрямі. Як правило, FIN-сегмент надсилається у результаті завершення роботи прикладного застосування.

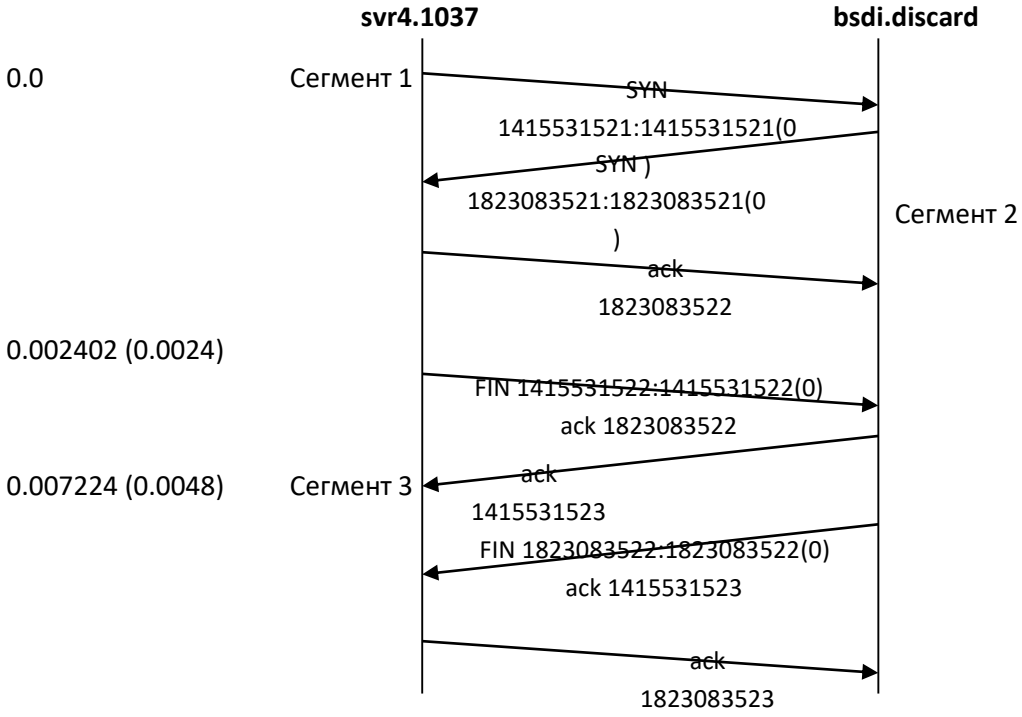


Рис. 2.10 Часова діаграма встановлення і розриву з'єднання.

На часовій діаграмі використовується формат виводу утиліти tcpdump. Запис виду 1415531521:1415531521 (0) означає, що порядковий номер сегмента дорівнює 1415531521, а обсяг переданих даних у цьому сегменті становить 0 байт.

Отримання TCP-сегмента з установленим прапором FIN означає припинення передавання даних лише в одному напрямі. При цьому TCP-вузол, який отримав FIN, може продовжувати надсилати дані у зворотному напрямі. Хоча протокол TCP підтримує роботу з напівзакритими з’єднаннями, на практиці більшість прикладних застосувань не використовують цю можливість. Найпоширеніший сценарій розриву з’єднання наведено на рис. 2.10.

Сторона, яка першою ініціює завершення з’єднання шляхом надсилання FIN-сегмента, вважається такою, що виконує активне закриття. Відповідно, сторона, яка отримує цей сегмент, здійснює пасивне закриття. У більшості випадків одна

сторона виконує активне закриття, а інша — пасивне, однак можливі ситуації, коли обидві сторони ініціюють активне закриття одночасно.

Незважаючи на те, що TCP-з'єднання є повнодуплексним, процес його завершення доцільно розглядати як закриття двох незалежних сімплексних з'єднань. Кожен напрям передавання даних завершується окремо. Для закриття з'єднання будь-яка зі сторін може надіслати TCP-сегмент із установленим прапором FIN, що сигналізує про завершення передавання даних у відповідному напрямі. Після отримання підтвердження цей напрям вважається закритим, тоді як передавання даних у протилежному напрямі може тривати.

TCP-з'єднання остаточно завершується лише після закриття обох напрямів передавання. У типовому випадку для цього необхідний обмін чотирма TCP-сегментами: двома сегментами з прапором FIN і двома сегментами з прапором ACK. В окремих ситуаціях підтвердження ACK і сегмент FIN можуть передаватися в одному TCP-сегменті, що зменшує загальну кількість сегментів до трьох.

Аналогічно до телефонної розмови, обидві сторони TCP-з'єднання можуть ініціювати завершення одночасно, надсилаючи FIN-сегменти паралельно. У цьому випадку кожна зі сторін отримує відповідні підтвердження, після чого з'єднання закривається. З погляду функціонування протоколу різниця між послідовним і одночасним розривом з'єднання є несуттєвою.

Для запобігання зависанню з'єднань у TCP використовуються таймери. Якщо підтвердження на надісланий FIN-сегмент не надходить протягом двох максимальних інтервалів часу життя пакета, відправник примусово завершує з'єднання. Інша сторона з часом також виявляє відсутність відповіді та закриває з'єднання

3 ДОСЛІДЖЕННЯ БАЗОВИХ МЕТОДІВ УПРАВЛІННЯ ПОТОКОМ ЗА ПРОТОКОЛОМ TCP/IP В МЕРЕЖАХ NGN

3.1 Механізм управління потоком – метод ковзаючого вікна

Для підвищення швидкодії та оптимізації процесу передавання великих обсягів даних у протоколі TCP використовується механізм управління потоком, відомий як метод ковзного вікна. Його особливістю є можливість передавання кількох TCP-сегментів без необхідності очікування підтвердження отримання кожного попереднього сегмента від приймальної сторони.

Розглянемо основні принципи функціонування методу ковзного вікна. У протоколі TCP підтвердження формується не для кожного окремого сегмента, а для всього безперервного блоку даних, починаючи від початку передавання і до певного порядкового номера ACK SN (Acknowledgment Sequence Number). Таким чином, підтвердження сигналізує про успішне отримання всіх байтів до вказаного значення.

Нехай ISN — початковий порядковий номер байта, встановлений під час ініціації TCP-з’єднання, а SN — поточний порядковий номер. Після встановлення з’єднання виконується співвідношення

SN=ISN+1(3.1)SN = ISN + 1 \tag{3.1}SN=ISN+1(3.1)

Якщо після встановлення з’єднання відправник передав декілька TCP-сегментів, у яких загалом було передано nnn байтів даних, то підтвердження успішного приймання цього обсягу інформації має вигляд

ACK SN=SN+n+1=(ISN+1)+n+1(3.2)ACK\,SN = SN + n + 1 = (ISN + 1) + n + 1 \tag{3.2}ACKSN=SN+n+1=(ISN+1)+n+1(3.2)

Це означає, що всі байти потоку даних з порядковими номерами від ISN+1ISN + 1ISN+1 до ISN+1+nISN + 1 + nISN+1+n були успішно отримані приймальною стороною.

Разом із передаванням значення ACK SN одержувач повідомляє відправника про поточний розмір вікна. Оголошений розмір вікна визначає діапазон порядкових номерів, у межах якого відправник має право надсилати дані без очікування додаткових підтверджень. Зокрема, відправник може передавати сегменти з порядковими номерами від поточного значення ACK SN до ACK SN+розмір вікна−1ACK\,SN + \text{розмір вікна} - 1ACKSN+розмір вікна−1.

У разі відсутності нових підтверджень відправник продовжує передавання доти, доки обсяг непідтверджених даних не перевищить оголошений розмір вікна. Після досягнення цієї межі передавання тимчасово припиняється до отримання чергового підтвердження та оновленого значення вікна. Розмір вікна підбирається таким чином, щоб підтвердження надходили вчасно і не виникали паузи в передаванні даних. При цьому одержувач має можливість динамічно змінювати розмір вікна залежно від стану власних буферів.

Для тимчасового припинення передавання даних одержувач може оголосити нульове вікно. Навіть у цьому випадку протокол TCP періодично надсилає спеціальні сегменти мінімального розміру, що містять один октет даних. Це необхідно для того, щоб відправник гарантовано отримав інформацію про відновлення ненульового розміру вікна, оскільки одержувач зобов’язаний підтвердити отримання таких сегментів і вказати актуальне значення вікна у відповіді.

Таким чином, механізм ковзного вікна в протоколі TCP використовується для регулювання інтенсивності трафіку та запобігання переповненню буферів приймальної сторони. Завдяки цьому відправник може передавати кілька

сегментів підряд без зупинок на очікування підтверджень, що істотно підвищує ефективність і швидкість обміну даними.

3.2 Механізм управління потоком – повільний старт

Алгоритм повільного старту призначений для визначення оптимальної початкової швидкості передавання даних у мережі. Його основна ідея полягає в поступовому збільшенні інтенсивності передавання пакетів відповідно до швидкості надходження підтверджень від віддаленої сторони з'єднання.

Під час використання повільного старту в TCP вводиться додаткове керуюче вікно — вікно перевантаження (Congestion Window, CWnd). При встановленні нового з'єднання з вузлом, що розташований в іншій мережі, початковий розмір вікна перевантаження встановлюється рівним одному сегменту MSS, значення якого оголошується приймальною стороною. Кожного разу при отриманні підтвердження ACK розмір вікна перевантаження збільшується на величину одного сегмента. Хоча значення CWnd вимірюється в байтах, на етапі повільного старту його зростання відбувається дискретно — на розмір сегмента.

Максимальний обсяг даних, який може бути переданий відправником без отримання підтверджень, визначається мінімальним значенням між вікном перевантаження та оголошеним приймальним вікном. Таким чином, за допомогою вікна перевантаження відправник регулює навантаження на мережу, тоді як оголошене вікно використовується одержувачем для управління власними буферами.

На початковому етапі роботи відправник передає один TCP-сегмент і очікує підтвердження його отримання. Після надходження ACK розмір вікна перевантаження збільшується з одного сегмента до двох, що дозволяє передати вже два сегменти. Після підтвердження кожного з них розмір вікна

перевантаження зростає до чотирьох сегментів. Таким чином, на етапі повільного старту спостерігається експоненціальне зростання обсягу передаваних даних.

На певному етапі досягається максимальна пропускна спроможність для даного TCP-з’єднання та мережі загалом. У цьому випадку проміжні маршрутизатори починають відкидати пакети, що сигналізує про виникнення перевантаження.

Роботу механізму повільного старту та взаємодію розмірів вікон при передаванні неінтерактивних даних ілюстровано на рис. 3.1. У початковий момент часу відправник передає один сегмент. Оскільки вікно перевантаження дорівнює одному сегменту, передавання наступних даних можливе лише після отримання відповідного підтвердження.

Переданий сегмент послідовно проходить мережу, після чого приймальна сторона формує підтвердження, яке повертається до відправника. Час між відправленням сегмента та отриманням підтвердження визначає час кругового проходження пакета (RTT). Після отримання ACK відправник може передати два сегменти, що відповідає збільшенню вікна перевантаження до двох сегментів. Надалі цей процес повторюється, забезпечуючи поступове нарощування швидкості передавання даних.

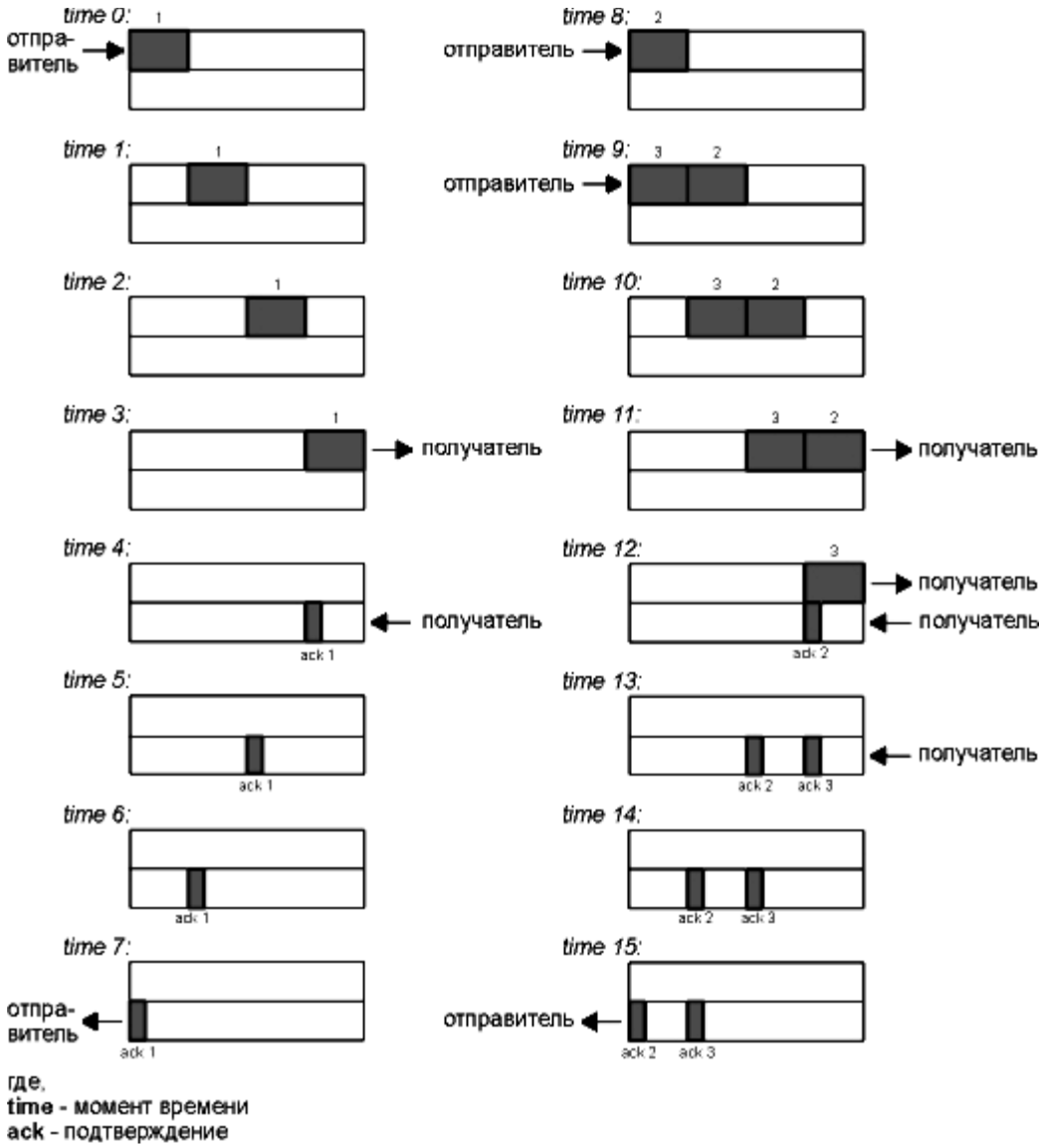


Рис. 3.1 Моменти часу 0-15, що ілюструють пропускну спроможність при передаванні даних.

Інтервали часу між надходженням підтверджень (ACK) відповідають інтервалам між передаванням сегментів даних. Така властивість роботи TCP отримала назву самоналаштування за часом (self-clocking). Оскільки приймальна сторона може сформувати підтвердження лише після фактичного отримання даних, часові інтервали між ACK відображають реальну швидкість надходження інформації до одержувача. Завдяки цьому відправник опосередковано отримує інформацію про поточний стан мережі та може коригувати інтенсивність передавання.

Перевантаження мережі виникає у випадках, коли швидкість надходження даних перевищує можливості подальшої передачі. Зокрема, така ситуація можлива, коли дані передаються з високошвидкісного каналу, наприклад локальної мережі, у повільніший канал глобальної мережі. Крім того, перевантаження може виникати в маршрутизаторах, до яких одночасно надходить декілька потоків даних, а сумарна швидкість вхідного трафіку перевищує пропускну спроможність вихідного інтерфейсу.

3.3 Алгоритм запобігання перевантаження AIMD

Регулювання трафіку в протоколі TCP ґрунтується на функціонуванні двох незалежних механізмів. Перший з них — контроль потоку, який здійснюється приймальною стороною за допомогою параметра розміру вікна (Window). Другий механізм — контроль перевантаження, що реалізується відправником на основі вікна перевантаження (Congestion Window, CWnd) та порогового значення повільного старту (Slow Start Threshold, SSThresh).

Алгоритм повільного старту використовується для початкового встановлення інтенсивності передавання даних у TCP-з’єднанні. Водночас під час зростання швидкості передавання може бути досягнута межа пропускної спроможності одного з проміжних маршрутизаторів, що призводить до втрати пакетів. Для зменшення ймовірності таких втрат у протоколі TCP застосовується алгоритм запобігання перевантаженню.

Основне припущення, на якому базується алгоритм запобігання перевантаженню, полягає в тому, що за нормальних умов у мережі втрачається незначна кількість пакетів. Тому поява втрат інтерпретується як ознака перевантаження на одному з вузлів між джерелом і пунктом призначення. Виявлення втрат здійснюється за двома основними індикаторами: виникненням тайм-аутів, коли підтвердження не надходить у встановлений термін, а також отриманням дубльованих підтверджень ACK.

Алгоритми повільного старту та запобігання перевантаженню є незалежними між собою і використовують різні керуючі параметри. Проте на практиці вони функціонують спільно. У разі виникнення перевантаження швидкість передавання даних зменшується, після чого знову застосовується повільний старт для поступового відновлення пропускної спроможності з'єднання.

Під час керування передаванням даних необхідно враховувати дві потенційні обмежувальні умови: пропускну спроможність мережі та ємність буферів приймальної сторони. З цією метою відправник використовує два вікна — вікно, оголошене одержувачем, і вікно перевантаження CWnd. Кожне з них визначає максимально допустимий обсяг даних, які можуть бути передані без підтвердження.

Фактичний обсяг передавання визначається мінімальним із цих двох значень. Наприклад, якщо приймальна сторона дозволяє передавання 8 Кбайт даних, але відправник оцінює, що мережа здатна пропустити лише 4 Кбайт без виникнення перевантаження, передавання буде обмежене цим меншим значенням. Якщо ж пропускна спроможність мережі є достатньою, відправник може передати весь обсяг, дозволений приймачем.

На етапі встановлення з'єднання розмір вікна перевантаження ініціалізується значенням, що дорівнює максимальному розміру сегмента, який використовується в даному з'єднанні. Після цього відправляється один сегмент максимального розміру. Якщо підтвердження його отримання надходить до закінчення часу очікування, розмір вікна перевантаження збільшується на розмір сегмента, що фактично призводить до подвоєння дозволеного обсягу передавання.

У подальшому, за умови своєчасного надходження підтверджень, розмір вікна перевантаження збільшується пропорційно його поточному значенню. Якщо розмір вікна дорівнює nnn сегментам, то після отримання підтверджень для всіх цих сегментів воно збільшується ще на nnn сегментів. Таким чином, на

початковому етапі має місце експоненціальне зростання розміру вікна перевантаження.

Цей процес триває до моменту досягнення межі, визначеної розміром приймального вікна, або до виникнення тайм-ауту, що свідчить про перевантаження мережі. Наприклад, якщо сегменти розміром 1024, 2048 та 4096 байт були доставлені успішно, але підтвердження для сегмента розміром 8192 байти не надійшло у встановлений час, розмір вікна перевантаження зменшується до 4096 байт. У цьому стані передавання більших сегментів не виконується незалежно від значення вікна, оголошеного приймальною стороною.

Для реалізації алгоритмів повільного старту та запобігання перевантаженню в TCP для кожного з’єднання використовуються дві основні змінні: вікно перевантаження CWnd та порогове значення повільного старту SSThresh. Початкове значення порогу зазвичай встановлюється на рівні 64 Кбайт. У разі виникнення тайм-ауту нове значення порогу встановлюється рівним половині поточного розміру вікна перевантаження, після чого саме вікно перевантаження зменшується до розміру одного максимального сегмента.

3.4 Параметри які впливають на пропускну здатність протокола TCP

Розглянемо підхід до визначення максимально можливої пропускної спроможності TCP-з’єднання, яка залежить від розміру вікна передавання, затримки в мережі та швидкості передавання даних.

Введемо такі позначення:

- WWW — розмір вікна передавання, байт;
- RRR — швидкість передавання даних, біт/с, доступна на стороні відправника;
- DDD — затримка передавання даних між відправником і одержувачем, Припустимо, що відправник починає передавання послідовності байтів одержувачу через вже встановлене TCP-з’єднання. Перший байт досягне одержувача через проміжок часу, що дорівнює DDD секундам. Такий самий час буде необхідний для надходження підтвердження від одержувача до відправника. Протягом цього інтервалу часу відправник теоретично здатний передати ще 2RD2RD2RD біт інформації, що відповідає RD/4RD/4RD/4 байтам. Однак на практиці обсяг переданих даних обмежується розміром вікна передавання WWW, оскільки зсув вікна можливий лише після отримання підтвердження.

Максимальна пропускна спроможність TCP-з’єднання досягається за умови

$$W \ge \frac{R D}{4} W \ge 4RD$$

Якщо ж розмір вікна є меншим за це значення, фактична пропускна спроможність з’єднання буде нижчою за максимально можливу. У такому випадку ступінь наближення реальної пропускної спроможності до максимальної визначається співвідношенням між розміром вікна WWW та величиною RD/4RD/4RD/4.

На рис. 3.2 наведено залежність максимальної пропускної спроможності TCP-з’єднання від параметра RDRDRD, яка наочно демонструє вплив затримки та швидкості передавання на ефективність використання мережного каналу.

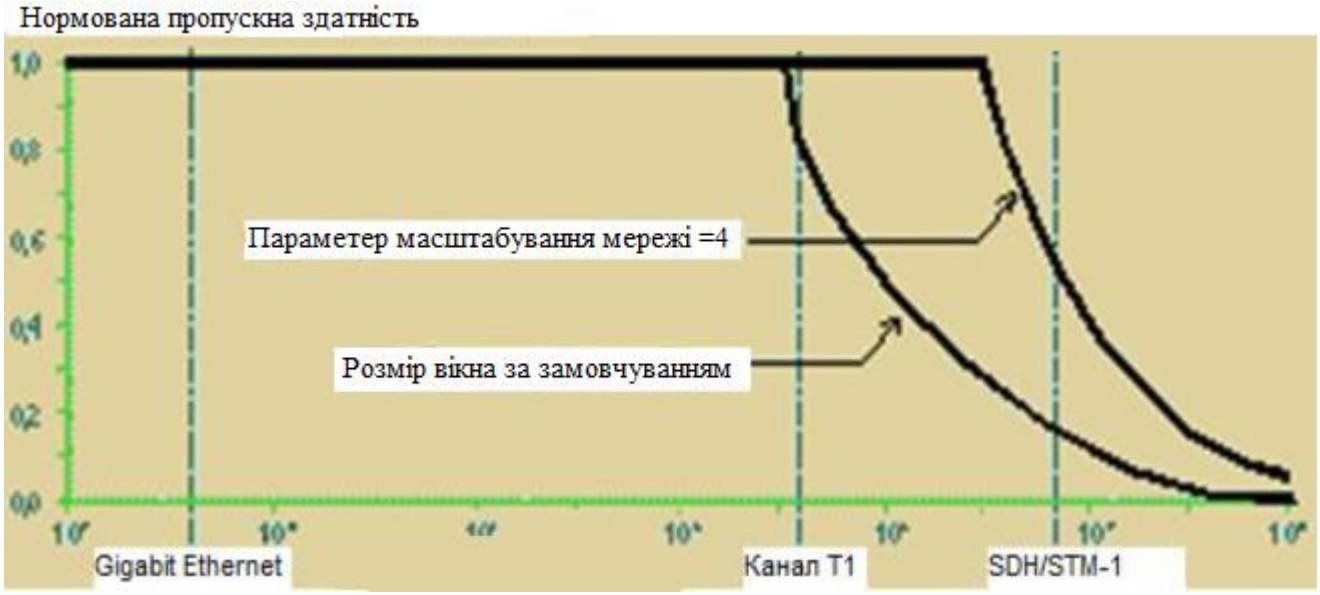


Рис. 3.2 Вплив параметра масштабування вікна на максимальну пропускну здатність

Максимальний розмір стандартного вікна TCP, що визначається 16-бітовим полем заголовка, становить приблизно 65 тис. байт, чого достатньо для більшості прикладних застосувань, які використовуються в мережах передачі даних. Розглянемо декілька типових технологій передавання, у яких TCP використовує дане обмежене за розміром вікно.

Так, наприклад, для каналу Gigabit Ethernet довжиною близько 100 м значення параметра RDRDRD є відносно малим і не перевищує 10310^3103 біт. За таких умов стандартного розміру вікна TCP цілком достатньо для ефективного використання пропускної спроможності каналу.

На більших відстанях або за меншої швидкості передавання ситуація змінюється. Наприклад, при використанні каналу T1 зі швидкістю 1,544 Мбіт/с значення параметра RDRDRD зростає, що призводить до зменшення ефективності передавання. Проте навіть у цьому випадку ефективність залишається на прийнятному рівні, близько 0,8. За подальшого збільшення відстані або швидкості

передавання стандартний механізм TCP стає менш ефективним. Так, для каналу SDH зі швидкістю 155 Мбіт/с нормована пропускна спроможність може знижуватися до рівня близько 0,1, що свідчить про недостатність стандартного розміру вікна.

У таких умовах виникає необхідність використання механізму масштабування вікна, який дозволяє збільшити ефективний розмір вікна та забезпечити повніше використання пропускної спроможності каналу. Наприклад, збільшення параметра масштабування лише на кілька одиниць дозволяє суттєво розширити розмір вікна і значно підвищити ефективність передавання. Аналіз взаємного впливу розміру вікна, затримки та швидкості передавання дає змогу оцінити потенціал протоколу TCP щодо підвищення його продуктивності.

Водночас слід враховувати низку факторів, які ускладнюють реальну картину. По-перше, через один канал зв’язку часто мультиплексується декілька TCP-з’єднань, у результаті чого кожне з них отримує лише частину доступної пропускної спроможності. Це призводить до зменшення ефективної швидкості передавання та зниження значення параметра RRR.

По-друге, TCP-з’єднання зазвичай проходять через декілька маршрутизаторів, унаслідок чого загальна затримка DDD дорівнює сумі затримок у всіх каналах і вузлах маршрутизації на шляху передавання. При виникненні перевантажень затримка в маршрутизаторах може вносити основний вклад у загальне значення параметра DDD.

По-третє, параметр RRR визначає швидкість передавання даних, доступну на стороні відправника. Якщо на будь-якій ділянці шляху від відправника до одержувача пропускна спроможність є меншою, виникає «вузьке місце», що неминуче призводить до збільшення затримки та зниження ефективності передавання. Додаткове зменшення фактичної пропускної спроможності

спостерігається у випадку повторних передавань втрачених сегментів, вплив яких залежить від механізмів повторної передачі.

Для підвищення ефективності TCP-з'єднань у мережах із великими затримками в операційних системах сімейства Microsoft Windows NT 5.0 було впроваджено алгоритм масштабування вікна. За його використання ефективний розмір приймального вікна визначається як значення поля Window, помножене на коефіцієнт 2F2^F2F, де FFF — параметр масштабування. Максимальне допустиме значення цього параметра для TCP дорівнює 14.

Слід зазначити, що механізм масштабування вікна активується лише на етапі ініціації TCP-з'єднання. В операційних системах Windows NT розмір приймального вікна збільшується на величину, кратну максимальному розміру сегмента MSS. Значення MSS визначається під час встановлення з'єднання та залежить від характеристик мережного середовища, що наведено в таблиці 3.1.

Таблиця. 3.1

Параметри реєстру Windows NT, які регулюють роботу протоколу TCP

Параметер	TcpipWindowSize	Tcp1323Opts
Ключ в реєстру	Tcpip\Parameters, Tcpip\Parameters\Interface\<interface>	Tcpip\parametrs
Тип запису	REG_DWORD – Number of bytes (кількість байт)	REG_DWORD – Number (флаги)
Можливі значення	0-0x3FFFFFFF	0,1,2,3
Значення задане мережею	17520 (для мережі Ethernet)	3

Розмір приймального вікна, що використовується в мережах на базі технології Ethernet, дозволяє передавати близько 8760 байт даних для операційної системи Windows NT 4.0 та приблизно 17520 байт для Windows NT 5.0. Використання вікон із розміром, що перевищує 64 Кбайт, можливе лише за умови, що обидві сторони TCP-з'єднання підтримують відповідний механізм масштабування.

Параметр розміру вікна може розглядатися як глобальне налаштування системи, а також як окремий параметр для кожного мережного інтерфейсу. При цьому значення, задане для конкретного інтерфейсу, має вищий пріоритет і перекриває загальносистемне значення. Такий підхід дозволяє гнучко налаштовувати параметри TCP залежно від характеристик конкретного мережного підключення.

Для керування використанням розширених можливостей протоколу TCP, визначених у RFC 1323, застосовується параметр TCP1323Opts. Він може приймати такі значення:

- 0 — використання опцій RFC 1323 заборонене;
- 1 — дозволено застосування лише механізму масштабування вікна;
- 2 — дозволено використання лише часових міток;
- 3 — дозволено одночасне використання масштабування вікна та часових міток.

Застосування відповідних значень параметра TCP1323Opts дозволяє оптимізувати роботу TCP-з’єднань у мережах з великими затримками та високою пропускною спроможністю, забезпечуючи більш ефективне використання доступних мережних ресурсів.

3.5 “Вузькі місця” в мережі

У протоколах канального рівня керування потоком даних із використанням механізму ковзного вікна надає одержувачу можливість регулювати швидкість передавання інформації від відправника. У цьому випадку одержувач підтверджує отримані сегменти та змінює розмір приймального вікна відповідно до наявного вільного буферного простору. Аналогічний підхід застосовується і в протоколі TCP, де швидкість передавання даних визначається, зокрема, швидкістю надходження підтверджень на раніше передані сегменти.

У класичному випадку швидкість передавання даних TCP-з’єднання обмежується швидкістю отримання підтверджень, яка, у свою чергу, залежить від

характеристик мережі між відправником і одержувачем. При цьому основним чинником обмеження є наявність так званих «вузьких місць» у мережі зв’язку.

Під «вузьким місцем» розуміють мережевий елемент або ділянку каналу, пропускна спроможність якої є значно меншою порівняно з пропускною спроможністю всієї мережі в цілому. У деяких випадках вузьким місцем може виступати і станція-одержувач, якщо її обчислювальні або буферні ресурси є обмеженими.

Вузькі місця доцільно поділяти на фізичні та логічні (рисунок 3.3). Фізичні вузькі місця виникають тоді, коли швидкість одного з каналів значно нижча за швидкість інших ділянок мережі. Наприклад, якщо відправник має пропускну спроможність близько 10 Мбіт/с, а канал між маршрутизаторами забезпечує швидкість лише 1,5 Мбіт/с, то саме цей канал стає фізичним вузьким місцем для TCP-з’єднання. Після встановлення сталого режиму передавання протокол TCP у такій ситуації використовує лише доступну пропускну спроможність цього каналу.

Однак на практиці частіше зустрічаються логічні вузькі місця, які формуються внаслідок виникнення черг у маршрутизаторах, комутаторах або на приймальних вузлах. Затримки в таких чергах зазвичай мають змінний характер і схильні до флуктуацій, що істотно ускладнює формування стабільного потоку даних.

Флуктуації затримок, характерні для розподілених IP-мереж, ускладнюють вибір оптимальної політики передавання даних протоколом TCP на стороні відправника. Якщо швидкість передавання є занадто малою, мережні ресурси використовуються неефективно. У випадку ж, коли один або кілька відправників передають дані з підвищеною швидкістю, інші TCP-потоки можуть бути витіснені або обмежені, що призводить до нерівномірного розподілу пропускної спроможності між з’єднаннями.

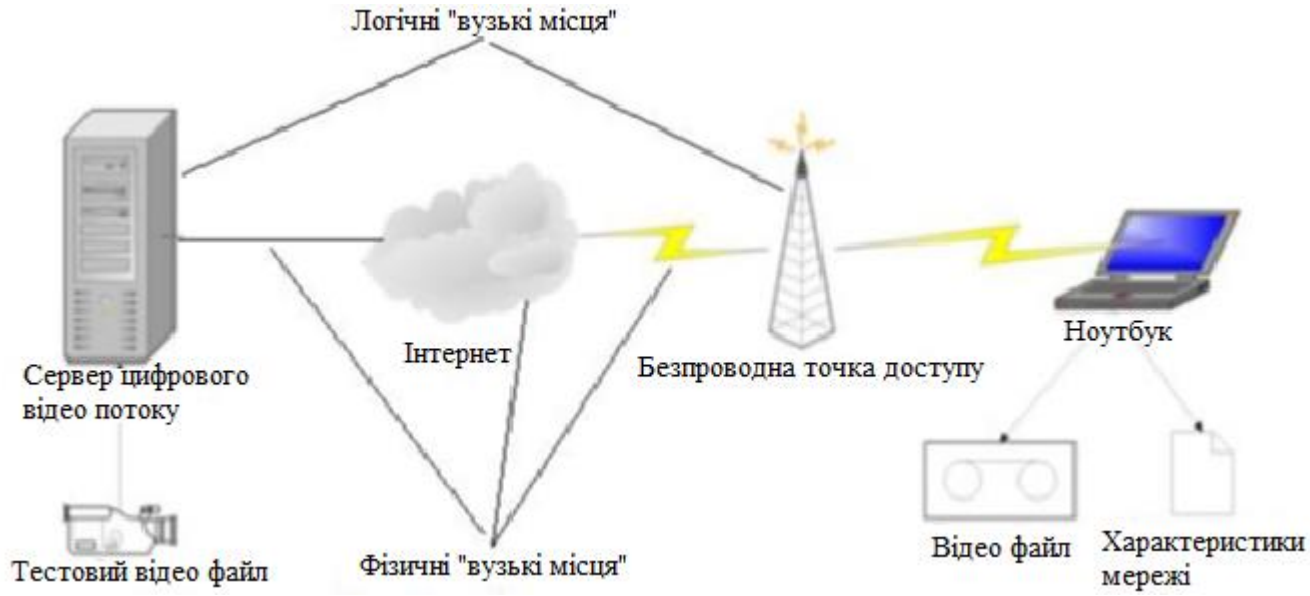


Рис. 3.3 Приклади логічних і фізичних «вузьких місць»

У випадку, коли значна кількість відправників протоколу TCP використовує надмірно високу швидкість передавання даних, у мережі починають виникати втрати сегментів або істотно зростає затримка надходження підтверджень. Це, у свою чергу, призводить до запуску механізмів повторної передачі сегментів, які не були підтверджені в установлений проміжок часу.

Зі збільшенням кількості повторно переданих сегментів навантаження на мережу зростає ще більше. Така ситуація формує замкнене коло: додаткові повторні передачі спричиняють подальше зростання затримок, збільшення кількості відкинутих сегментів і, як наслідок, ще більшу кількість повторних передавань. У результаті загальна ефективність роботи мережі істотно знижується.

Розглянемо випадок, коли «вузьке місце» безпосередньо знаходиться в мережі зв’язку. На рисунку 3.4 показано низькошвидкісний канал між відправником і одержувачем, пропускна спроможність якого є обмеженою. Ширина такого каналу умовно відображає максимальну швидкість передавання даних, доступну

для TCP-з’єднання. Саме цей канал визначає граничну продуктивність передавання незалежно від можливостей кінцевих вузлів.

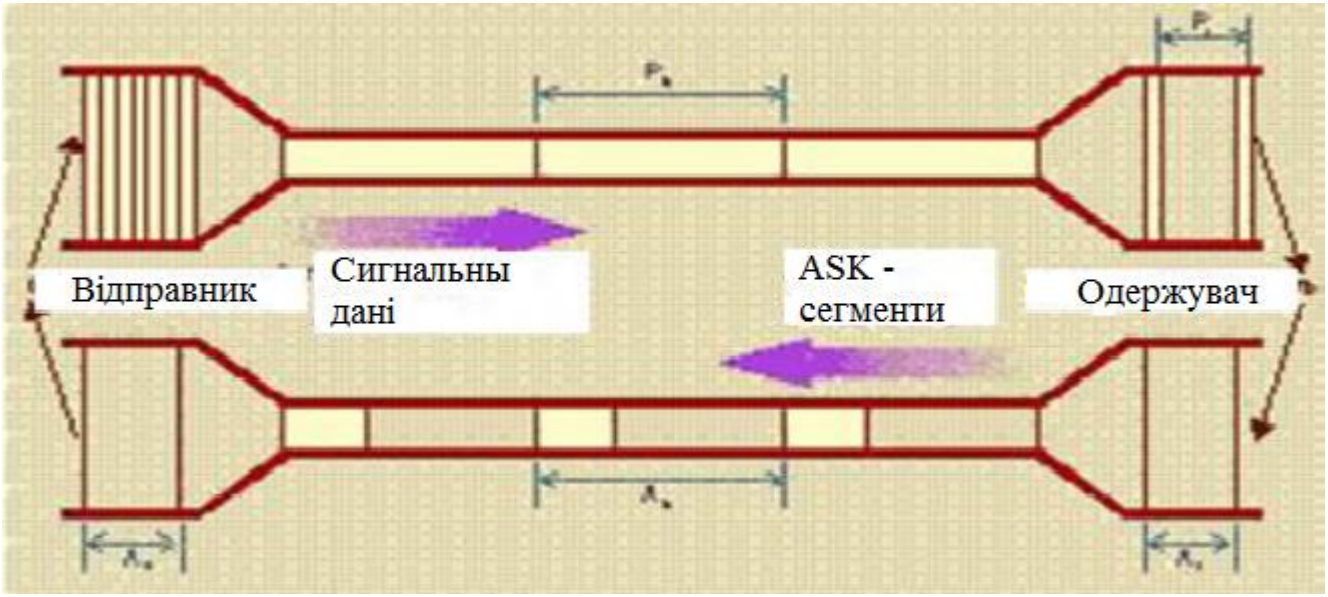


Рис. 3.5 Приклад «вузьких місць» в мережі

Нехай відправник і одержувач підключені до високошвидкісних мереж зв’язку, тоді як канал між ними має значно меншу пропускну спроможність і саме він утворює «вузьке місце». Кожен переданий сегмент даних можна умовно зобразити у вигляді прямокутника, площа якого пропорційна кількості байтів інформації в сегменті. Під час проходження через низькошвидкісний канал такий сегмент ніби «розтягується» в часі, що призводить до збільшення тривалості його передавання.

Позначимо через PbP_bPb час проходження сегмента через фіксовану точку повільного каналу. Цей час визначається різницею між моментами проходження передньої та задньої меж сегмента через зазначену точку. Припустимо, що в повільному каналі сегменти передаються без розривів, тобто задня межа одного сегмента безпосередньо прилягає до передньої межі наступного. У такому

випадку величина PbP_bPb також визначає інтервал часу між проходженням через задану точку передніх меж двох послідовних сегментів.

Після надходження сегментів у високошвидкісний канал зв’язку цей часовий інтервал не змінюється, незважаючи на більшу допустиму швидкість передавання. Це пояснюється тим, що темп надходження сегментів з повільного каналу залишається незмінним. У високошвидкісному каналі сегмент передається швидше, тобто «стискається» в часі, а проміжок між сегментами збільшується. У результаті інтервал часу між передніми межами двох сусідніх сегментів у цьому каналі, який позначимо через PrP_rPr, дорівнює PbP_bPb.

Якщо одержувач формує підтвердження одразу після отримання кожного сегмента, то інтервал між відправленням ACK-сегментів на виході одержувача, який позначимо через ArA_rAr, дорівнює PrP_rPr. Далі, час проходження підтверджуючих сегментів через повільний канал AbA_bAb буде дорівнювати ArA_rAr, а інтервал між надходженням ACK-сегментів до відправника AsA_sAs — значенню AbA_bAb.

Після завершення перехідного процесу система переходить у сталий режим, у якому швидкість передавання сегментів даних відправником дорівнює швидкості надходження підтверджень. Очевидно, що ця швидкість обмежується пропускною спроможністю найповільнішого каналу на шляху передавання. Таким чином, протокол TCP автоматично виявляє «вузьке місце» в мережі та відповідним чином регулює інтенсивність потоку даних. Такий механізм отримав назву самосинхронізації (self-clocking).

3.6 Математична модель встановлення TCP-з'єднання

Для математичного опису телекомунікаційних систем на сьогодні застосовується широкий спектр підходів, що ґрунтуються на використанні різних математичних апаратів. Серед них можна виділити теоретико-ігрові моделі, моделі керуючих автоматів, моделі прийняття оптимальних рішень, імовірнісні моделі, а також агрегативні моделі складних систем. Кожен із зазначених підходів формує власний спосіб формалізації задач, пов'язаних як із процесами маршрутизації, так і з керуванням мережевими параметрами та інформаційним обміном, які зрештою зводяться до оптимізаційних задач.

Поряд із наведеними методами особливу увагу привертає апарат диференціально-різницевих рівнянь стану мережі, який найбільш повно відповідає вимогам, сформульованим у межах даного дослідження. Модель у просторі станів дозволяє враховувати динамічний характер процесів, що відбуваються в мережі, зокрема TCP-сеансів, зміну структурних і функціональних параметрів мережі, а також можливість одночасного існування декількох потоків даних [12].

Аналіз різних реалізацій протоколу TCP показує, що в процесі передавання даних виділяється кілька стандартних режимів (фаз) роботи, зокрема slow start, congestion avoidance та fast recovery. Перехід між цими режимами, починаючи з моменту встановлення з'єднання, визначається рівнем завантаженості каналів зв'язку, станом буферів проміжних вузлів і приймальних пристроїв, а також кількістю помилок і втрат даних. У такому разі модель TCP-сеансів у просторі станів може бути представлена у вигляді системи неавтономних диференціальних рівнянь, що описують динаміку передавання даних між вузлом-відправником і вузлом-одержувачем за наявності кількох TCP-потоків у каналі зв'язку:

$$P(t) = \frac{1}{m} \cdot \frac{N(t) - N_{\min}}{N_{\max} - N_{\min}}, \tag{3.1}$$

де m - знаменник граничної ймовірності;

N (t) - поточний середній розмір черги;

Nmin, Nmax - відповідно мінімальний і максимальний середній розмір черги.

У межах цієї моделі ймовірність блокування або відкидання пакетів P(t)P(t)P(t) може задаватися у вигляді довільної, але обґрунтованої функції. Як окремий випадок, при використанні алгоритмів керування чергами типу RED, ймовірність блокування визначається співвідношенням:

$$\frac{d\lambda_i(t)}{dt} = \begin{cases} slow_start \\ (1-P(t))\cdot\frac{MSS}{RTT}\cdot\lambda_i(t) - P(t)\cdot(\lambda_i(t))^2 + \\ +P(t)\cdot MSS\cdot\lambda_i(t); \\ congestion_avoidance \\ (1-P(t))\cdot\left(\frac{MSS}{8\cdot RTT}\cdot\lambda_i(t) + \frac{MSS\cdot MSS}{RTT^2}\right) - \\ -P(t)\cdot(\lambda^2(t))^2 + P(t)\cdot MSS\cdot\lambda_i(t), \end{cases} \quad (3.2)$$

де ai (t) - інтенсивність TCP-потоку, переданого в i-му сеансі;

i = 1, M;

MSS - максимальний розмір сегмента (MaximumSegmentSize);

RTT - період кругового обходу пакета (RoundTripTime);

P (t) - ймовірність блокування (відкидання) пакетів.

Тоді чергу на маршрутизаторі будуть включатись ті пакети сумарного потоку, що дозволяє врахувати їх розподіл пропускної спроможності між тими TCP-сеансами в межах одного каналу зв'язку. У механізмі типу WRED, передбачається використання окремих показників граничної ймовірності, а так же ваг для різних служб IP-пріоритетів, що забезпечує різні рівні якості обслуговування для різних типів трафіку в періоди переповнення черг мережевих вузлів і каналів зв'язку.

У моделі так само передбачається, що середній розмір його черги на мережевому вузлі можна розрахувати згідно з положенням теорії СМО, виходячи з формули Літтла

$$N(t) = \frac{\lambda(t)}{B - \lambda(t)} \cdot \frac{\lambda(t)}{B}, \tag{3.3}$$

де B - пропускна здатність каналу зв'язку.

3.7 Розрахунок часу затримки інформації в мережі за протоколом TCP/IP з урахуванням алгоритмів запобігання перевантаження.

Важливим параметром, що визначає робочі параметри таймерів, є RTT (час подорожі пакету до адресата і назад). TCP-агент самостійно вимірює RTT. Такі вимірювання проводяться періодично і за їх результатами коригується середнє значення RTT[8]:

$$RTTm = a \times RTTm + (1 - a) \times RTTi, \tag{3.4}$$

де RTTi - результат чергового вимірювання,

RTTm - величина, отримана в результаті усереднення попередніх вимірювань,

a - коефіцієнт згладжування, зазвичай рівний 0.9.

RFC-793 рекомендує встановлювати час тайм-ауту для ретрансміссіі (повторної передачі), значення RTO - Retransmission TimeOut одно

RTO = RTTm * b,

де b дорівнює 2.

Розрахуємо по формулі середнє значання показника RTT в момент часу з попередніх та щойно знятих показників і представемо у відношенні з показником RTTi, Рис.3.6

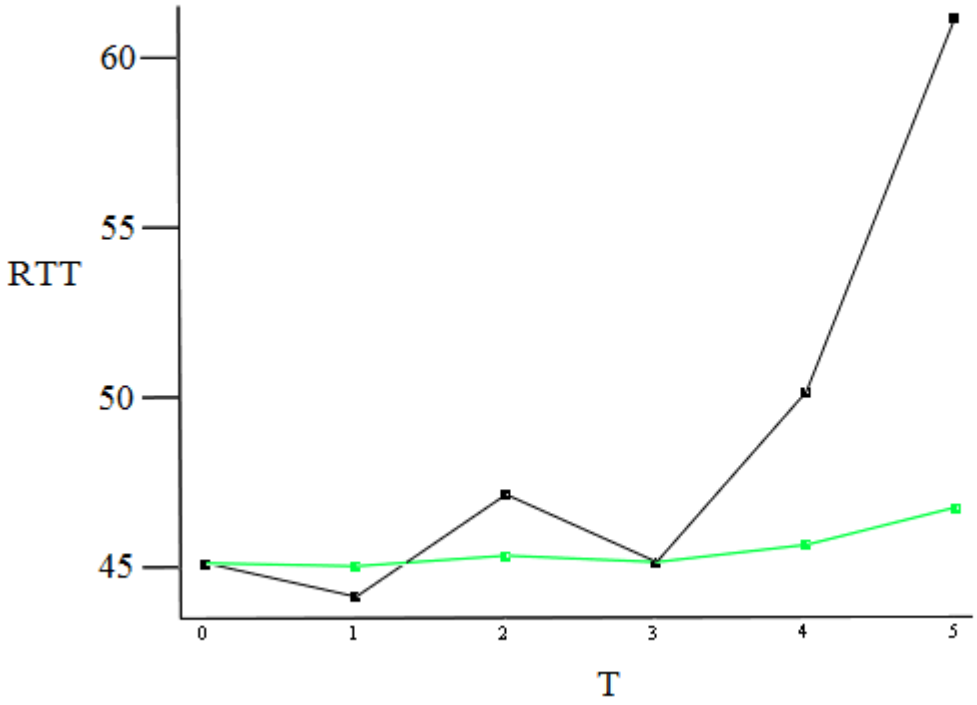


Рис. 3.6 Графік коригування середнього значення кругового обходу

На рисунку представлені результати вимірювання часу кругового обходу інформації, тобто часу затримки інформації в мережі, і середні значення виведені з формули 3.6., а саме опираючись на середні значення tcp робить зміну вікна. Звідси витікає що формула глажує середнє значення RTT що робить tcp менш агресивним.

3.8 Розрахунок залежності часу кругового обходу (RTT) та частоти втрат від вікна в базовому алгоритмі TCP в режимі виключення перевантаження

Останнім часом було запропоновано значну кількість варіантів протоколів, які позиціонуються як такі, що перевершують TCP за ефективністю. Більшість із них реалізуються у складі вебсерверів і, як правило, являють собою модифіковані версії стандартного TCP. Зокрема, деякі реалізації використовують таймери з підвищеною роздільною здатністю, що дозволяє швидше реагувати на зміни стану мережі.

Інші модифікації передбачають збільшене початкове значення вікна перевантаження CWND або прискорений алгоритм повільного старту, у якому розмір вікна зростає не вдвічі, а, наприклад, утричі за кожен інтервал RTT. Подібні підходи можуть також застосовуватися на фазі уникнення перевантаження, змінюючи характер лінійного зростання CWND. Усі ці рішення змушують відправника поводитися більш агресивно, значно підвищуючи навантаження на буфери мережевих пристроїв.

Існують також програмні рішення, у яких під час передавання файлів через мережу замість одного TCP-з'єднання створюється декілька паралельних віртуальних з'єднань. Подібні методи дозволяють досягти збільшення пропускної спроможності для конкретного застосування, однак це відбувається переважно за рахунок погіршення умов роботи інших одночасних TCP-сесій.

Такий підхід суперечить базовій філософії Інтернету, яка ґрунтується на принципах справедливого та кооперативного використання мережних ресурсів. Надмірно агресивна поведінка TCP-застосувань у кінцевому підсумку призводить до зниження загальної ефективності передавання даних і часто виявляється менш ефективною, ніж використання стандартного TCP.

Як правило, подібні модифікації спричиняють зростання втрат пакетів, оскільки прагнення максимально завантажити ресурси мережі знижує чутливість протоколу до сигналів перевантаження. У випадку масового поширення таких застосувань це може призвести до суттєвої деградації пропускної спроможності каналів зв'язку через хронічні перевантаження мережі.

Прийнятними вдосконаленнями протоколу TCP слід вважати лише ті, які не погіршують умови роботи інших користувачів мережі. Саме такий підхід є найбільш ефективною стратегією, оскільки він забезпечує раціональне та справедливе використання ресурсів мережі в цілому.

Роботу протоколу TCP AIMD в режимі виключення перевантажень можна характеризувати формулою 3.7:

$$BW = \frac{MSS}{(RTT \cdot \sqrt{(1,33 \cdot \rho)}) + (RTO \cdot \rho \cdot \left[1 + 32 \cdot \rho^2\right] \cdot \min(1, 3 \cdot \sqrt{0,75 \cdot \rho}))}, \quad (3.5)$$

де BW - смуга пропускання;

MSS - максимальний розмір сегмента в байтах, який використовується сесією.

RTT - час кругового обходу в секундах;

RTO - таймаут повторної пересилки.

ρ - частота втрати пакетів (0.01 означає 1% втрат)

Ця формула є найкращою апроксимацією. Деяке спрощення формули можна отримати, вважаючи $RTO = 5 \cdot RTT$.

Ще більш спрощена формула:

$$BW = 0,93 \cdot \frac{MSS}{RTT\sqrt{\rho}}, \quad (3.6)$$

Дані формули виведені при наступних припущеннях:

Вікно одержувача не обмежує робочих характеристик з'єднання.

Обидві формули дозволяють смузі досягати нескінченності при нульових втрати.

Значення RTT є середнім і включає в себе затримку, пов'язану з перебуванням пакета в черзі.

Формули працюють тільки для одного TCP-з'єднання. Якщо через канал здійснюється кілька TCP-з'єднань, кожне з них слід вказаними формулами.

Формули припускають довготривалі TCP-з'єднання. Для особливо коротких з'єднань (<10 пакетів), коли немає втрат, працює алгоритм повільного старту. Для сесій проміжної тривалості, де в середньому втрачається кілька сегментів, параметри виявляться дещо краще, ніж передбачають формули.

Різниця між простою і складною формулами полягає в тому, що складна формула враховує вплив таймаутів повторних передач. Для низького рівня втрат (<1%), тайм-аути зазвичай не відбуваються і формули дають практично ідентичні результати. При високих втратах (≥1%) складна формула дає більшу точність оцінки.

Таким чином при ймовірності втрати пакету 1% і RTT = 50 мсек максимальна пропускна здатність буде менше 3 Мбіт / с незалежно від широкополосности каналу.

Звідси випливає, що канали з BER ~ 1e-3 не здатні ефективно використовувати свою смугу пропускання. Це стосується практично всіх мобільних каналах.

Тому в RFC 5348 представлене рівняння для середньої швидкості передачі TCP (в байтах в секунду) X_Bps має вигляд:

$$X_Bps = \frac{s}{RTT \cdot \sqrt{2 \cdot b \cdot \frac{p}{3}} + \left(t_RTO \cdot \left(3 \cdot \sqrt{3 \cdot b \cdot \frac{p}{8}} \cdot p \cdot (1 + 32 \cdot p^2) \right) \right)}, \quad (3.7)$$

де:

X_Bps - середня швидкість передачі TCP (байт / сек); параметр X_Bps ідентичний X_Calc в RFC 3448;

s - розмір сегмента (байт) з урахуванням заголовків транспортного рівня і IP;

RTT - час кругового обходу в секундах;

p - ймовірність втрат (0 - 1.0) - частка втрачених пакетів в від загального числа переданих пакетів;

t_RTO - тайм-аут повторної передачі TCP в секундах;

b - максимальне число пакетів, які підтверджуються одним пакетом TCP ACK.

Вибір значення тайм-ауту повторної передачі TCP - t_RTO.

Реалізацій слід використовувати значення t_RTO = 4 * RTT. Можливе використання більш точного методу розрахунку t_RTO. Реалізації можуть також вибирати в якості t_RTO значення max (4 * RTT, 1 секунда) у відповідності з рекомендованим для значення RTO мінімумом [RFC2988].

Вибір параметра b для відкладених підтверджень.

Деякі сучасні реалізації TCP використовують відкладені підтвердження, передаючи один пакет підтвердження для кожної пари прийнятих пакетів даних. Однак TCP також дозволяє передавати підтвердження для кожного прийнятого пакета даних. Для переглянутих механізмів контролю насичення TCP [RFC2581bis] в даний час задає алгоритм відкладених підтверджень, який слід використовувати. Однак [RFC2581bis] рекомендує збільшувати розмір вікна насичення при запобіганні перевантаження по одному сегменту за період RTT навіть при використанні відкладених підтверджень, відповідно до рівняння пропускної здатності TCP для випадку b = 1. На основі експериментальних даних [RFC2581bis] дозволяє збільшувати вікно насичення в процесі уповільненої старту, що також відповідає рівняння пропускної здатності TCP для випадку b = 1. Таким чином, використання b = 1 узгоджується з [RFC2581bis]. Рекомендується встановлювати значення b = 1.

При t_RTO = 4 * RTT і b = 1 рівняння для пропускної здатності X_Bps (швидкість передачі TCP в байтах за секунду) може бути спрощено до[11]:

$$X_Bps = \frac{s}{RTT \cdot \left(\sqrt{2 \cdot \frac{p}{3}} + 12 \cdot \sqrt{3 \cdot \frac{p}{8}} \cdot p \cdot \left(1 + 32 \cdot p^2\right) \right)}, \qquad (3.8)$$

В майбутньому оновлені варіанти цього документа можуть використовувати інші рівняння для TCP замість наведеного тут. Вимога до таких рівнянь полягає в тому, що вони повинні давати розумне наближення швидкості передачі TCP для відповідності з механізмом контролю насичення TCP.

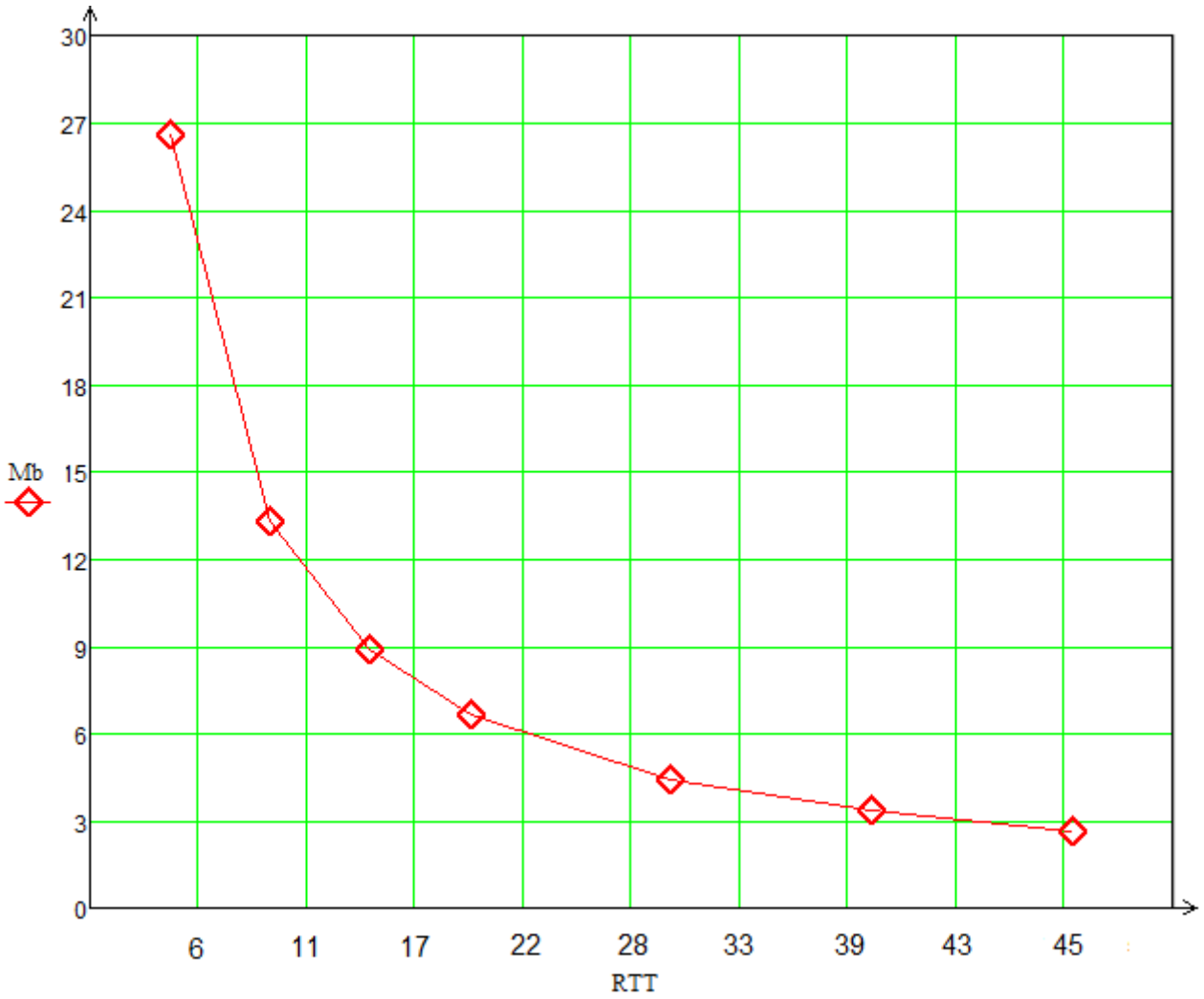


Рис. 3.7 Залежність часу затримки інформації відносно середньої швидкості передачі по протоколу TCP

Розрахуємо залежність часу затримки інформації відносно середньої швидкості передачі TCP і одержимо графік залежності часу затримки інформації в мережі від пропускної здатності мережі по формулі представленої в RFC 5348 рис. 3.7.

Отримали графік залежності при p=1%, звідси випливає що при збільшенні кругового часу обходу інформації тобто часу затримки її в мережі TCP реагує

зменьшенням вікна тим самим регулює поток у мережі але показує майже такіж дані як і робота протоколу TCP AIMD.

Розрахуємо залежність часу затримки інформації відносно середньої швидкості передачі TCP і одержимо графік залежності часу затримки інформації в мережі від пропускної здатності мережі по формулі (3.11.)

$$TW < \left(\frac{MSS}{RTT} \cdot \frac{C}{\sqrt{p}} \right), \tag{3.9}$$

TW – теоретична пропускна здатність мережі;

MSS – максимальний розмір сегмента;

RTT - час кругового обходу в секундах;

p - ймовірність втрат (0 - 1.0) - частка втрачених пакетів в від загального числа переданих пакетів; А параметр C=1.

Розглянемо формулу (3.7) При різних значеннях RTT 10, 50, 100.

На рисунці 3.8. ми бачимо руйнівну залежність середньої швидкості передачі від втрат в мережі але з формули витікає: алгоритм починає зменшувати інтенсивність передачі тим самим регулює потік даних не створюючи нескінченні черги що б призвело до росту RTT і ще більшого зменшеня швидкості передачі.

Ще одним фактором є максимальний розмір вікна. Якщо ви згадайте, то розмір цього вікна вказується в поле заголовка TCP, яке називається розмір вікна і має розмір 16 біт, що обмежує вікно 2 ^ 16 байтами (65536). Цього може бути мало для поточних високошвидкісних з'єднань , тому придуманий RFC 1323, де описується обхідний спосіб [10]. Чому мало? Тому що настройка цього вікна за замовчуванням така:

Для мереж зі швидкістю менше 1 мегабіта - 8 КБ (якщо точніше, 6 раз за стандартним MSS - 1460 байт)

Для мереж зі швидкістю 100 Мбіт і менш, але більше 1 Мбіт - 17 КБ (12 разів за стандартним MSS - 1460 байт)

Для мереж зі швидкістю вище 100 Мбіт - 64 КБ (максимальне значення без підтримки RFC 1323)

Номінальна швидкість, тобто за інформацією з інтерфейсу в момент його ініціалізації, а не про фактичну, яка може бути і відчутно нижче.

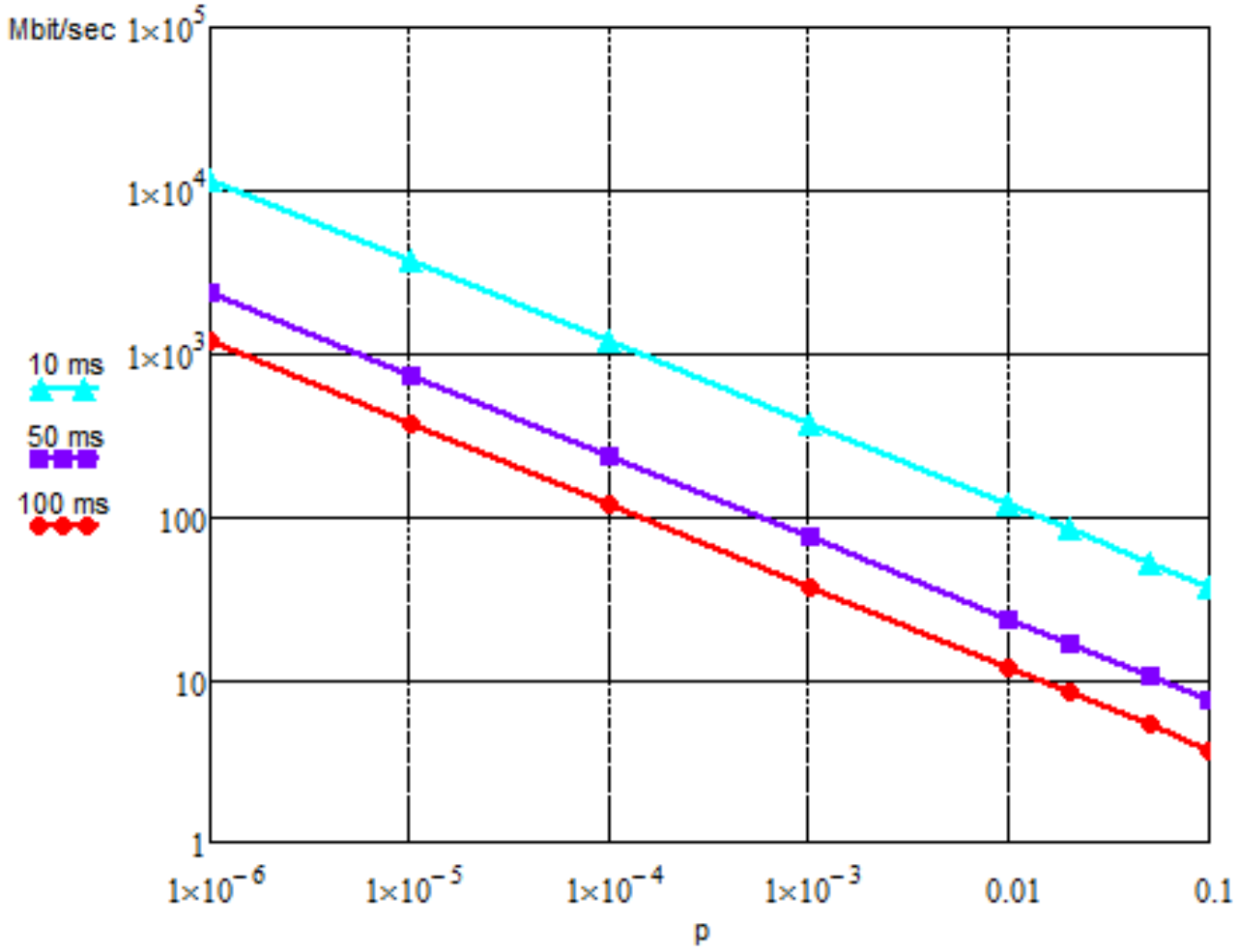


Рис. 3.8 залежність часу затримки інформації відносно середньої швидкості передачі TCP

Спосіб обходу, пропонований в RFC 1323, простий і красивий. Два хоста, що ставлять TCP-сесію, узгоджують між собою параметр, який є кількістю біт, на які буде зрушено значення поля вікна розмір. Тобто, якщо вони погодять цей параметр рівний 2, то обидва з них будуть читати це поле зсунутим "вліво" на 2 Біта, що дасть збільшення параметра в 2 ^ 2 = 4 рази. І, допустимо, значення цього поля в 64К перетвориться в 256К. Якщо узгодять 5 - то поле буде зрушено вліво на 5 Біт, і 64К

в 2Мб перетвориться. Максимальний підтримуваний Windows, поріг цього значення (масштабування) - 14, що дає максимальний розмір вікна в 1 ГБ. Все це не буде працювати без включення підтримки RFC 1323.

4 ДОСЛІДЖЕННЯ МОДЕЛЕЙ АЛГОРИТМІВ УПРАВЛІННЯ ПОТОКОМ ЗА ПРОТОКОЛОМ TCP

4.1 TCP Reno

У моделі TCP Reno за нормальних умов роботи зміна розміру вікна перевантаження має, як правило, циклічний характер. Розмір вікна поступово зростає доти, доки в мережі не відбувається втрата сегмента, що сигналізує про появу перевантаження. Алгоритм TCP Reno передбачає дві основні фази зміни розміру вікна перевантаження: фазу повільного старту та фазу уникнення перевантаження. У кожній з цих фаз використовується різний механізм регулювання швидкості передавання даних, що дозволяє адаптувати роботу TCP-з'єднання до поточного стану мережі.При отриманні відправником підтвердження про доставку в момент часу порядку $t+t_A$ [сек], поточне його значення розміру її вікна перевантаження $cwnd(t)$ перетворюється в $cwnd(t+t_A)$ згідно (4.1):

$$cwnd(t+t_A)=\begin{cases} \textit{Фаза повільного старту} \\ cwnd(t)+1, \quad if \quad cwnd(t)<ssth(t); \\ \textit{Фаза виключення перевантаження} \\ cwnd(t)+\dfrac{1}{cwnd(t)}, \quad if \quad cwnd(t)\geq ssth(t); \end{cases}, \tag{4.1}$$

де $ssth(t)$ [пакетів] - значення порога, при якому TCP переходить з фази повільного старту в фазу виключення перевантаження.

Коли в результаті таймаута детектується втрата пакета значення $cwnd(t)$ і $ssth(t)$ оновлюються наступним чином:

$$cwnd(t)=1, \tag{4. 2}$$

$$ssth(t) = \frac{(cwnd(t))}{2}, 2 \quad (4.3)$$

Тим самим, коли в TCP детектирует втрату її пакета згідно з алгоритмом швидкої повторної передачі, cwnd (t) і ssth (t) оновлюються інакше[15]

$$ssth(t) = \frac{(cwnd(t))}{2}, \quad (4.4)$$

$$cwnd(t) = ssth(t), \quad (4.5)$$

TCP-Reno після того переходить в наступну фазу швидкого відновлення. У цій самій фазі розмір її вікна збільшується як мінімум на один пакет, коли виходить дубльоване її підтвердження. З іншого боку, cwnd (t) робиться рівним параметру ssth (t), коли приходить не дубльований відгук для її пакета, посланого вже повторно. У разі таймаута порядку ssth (t) = (cwnd (t)) / 2; cwnd = 1 (дивись опис алгоритму TCP-Tahoe).

На сьогодні найбільш поширеною реалізацією є модель TCP NewReno, описана у стандарті RFC 3782, яка використовує алгоритми швидкої повторної передачі та швидкого відновлення. У разі підтримки опції вибіркового підтвердження SACK відправник отримує інформацію про конкретні сегменти, що були втрачені, і може повторно передавати лише ці пакети під час фази швидкого відновлення (fast recovery). За відсутності опції SACK або за недостатності інформації щодо втрачених сегментів повторна передача здійснюється на основі отриманих підтверджень ACK.

Після отримання трьох або більше дубльованих підтверджень (DUPACK) відправник вважає відповідний сегмент втраченим та ініціює його негайну повторну передачу. Надалі відправник може отримувати додаткові дубльовані підтвердження, оскільки одержувач продовжує підтверджувати сегменти, що вже

перебувають у мережі, у той час як відправник знаходиться в режимі fast retransmit.

У випадку втрати одного сегмента, за умови відсутності порушення порядку доставки пакетів, підтвердження повторно переданого сегмента означає успішну доставку всіх попередніх сегментів, відправлених до переходу в режим швидкої повторної передачі. Якщо ж у межах одного вікна перевантаження втрачено декілька сегментів, підтвердження повторно переданого пакета підтверджує доставку лише частини попередніх сегментів. Такі підтвердження називаються частковими і є характерною особливістю роботи алгоритму TCP NewReno.

4.2 TCP NewReno

Даний алгоритм використовує змінну типу recover (відновлення), який відновлює початкове значення і дорівнює вихідному порядковому номеру його пакета. До вузловим етапам реалізації алгоритму відносяться:

а) Три задубльованних запитів ACK .:

$$ssthreh = \max\left(\frac{FlightSize}{2}, 2 \cdot SMSS\right), \tag{4.6}$$

Крім того, змінна recover робиться рівною найбільшому порядковому номеру переданого пакета, і здійснюється перехід до пункту 2.

б) Вхід в даний режим швидкої повторної передачі пакетів, в якому передається втрачений сегмент і встановлюється параметр cwnd рівним ssthresh плюс 3 * SMSS (SenderMaximumSegmentSize).

Це штучно збільшує його вікно перевантаження на кілька сегментів (три), які покинули мережу і буферизованного одержувачем.

в) Швидке відновлення. Коли для кожного нового дубльованого запиту ACK, отриманого в режимі швидкого відновлення, параметр cwnd збільшується на SMSS.

Це штучно збільшує його вікно перевантаження, для того щоб врахувати даний сегмент, який покинув цю мережу.

г) Швидке відновлення, продовження. Коли даний сегмент передається, якщо це дозволено новим значенням параметра cwnd і значенням параметра window, оголошеним одержувачем.

д) Коли приходить повідомлення ACK, що підтверджує отримання нових цих даних, це повідомлення ACK може бути як підтвердженням, пов'язаних з повторною під час другого етапу або з більш пізньої повторною.

Під повним підтвердженням розуміється ситуація, коли ACK підтверджує доставку всіх даних аж до позиції, визначеної змінною recover. У такому випадку ACK підтверджує успішну доставку всіх проміжних сегментів, починаючи з повторної передачі втраченого сегмента після отримання трьох дубльованих підтверджень.

Після отримання повного підтвердження розмір вікна перевантаження cwnd встановлюється відповідно до одного з двох правил:

---

1. cwnd = min(ssthresh, FlightSize + SMSS), або
2. cwnd = ssthresh,
   де значення ssthresh відповідає величині, визначеній на етапі входу в режим швидкого відновлення.

Часткове підтвердження має місце тоді, коли ACK не підтверджує доставку всіх даних до позиції, заданої змінною recover. У такому випадку ACK сигналізує про доставку лише частини даних і вказує на наявність інших втрачених сегментів.

При отриманні часткового підтвердження відправник негайно ініціює повторну передачу першого непідтвердженого сегмента. Значення вікна перевантаження при цьому зменшується на обсяг нових переданих даних, що були підтверджені кумулятивним ACK. Якщо часткове підтвердження підтверджує доставку одного сегмента розміру SMSS, то до значення cwnd додається SMSS. Такий механізм

дозволяє штучно підтримувати розмір вікна перевантаження на рівні, що компенсує сегмент, який залишив мережу.

Після цього, за умови дозволу нового значення cwnd, виконується передавання наступного сегмента. Така стратегія забезпечує збереження режиму швидкого відновлення, а обсяг недоставлених даних підтримується на рівні, що не перевищує ssthresh. Вихід з режиму швидкого відновлення при часткових підтвердженнях не здійснюється.

Для першого часткового підтвердження, отриманого в режимі швидкого відновлення, таймер повторної передачі скидається під час повторної відправки сегмента.

У разі спрацювання таймауту повторної передачі (RTO) у змінну recover записується найбільший порядковий номер переданого сегмента. Після цього ініціюється вихід із режиму швидкого відновлення, якщо це дозволяють умови роботи протоколу.

Одним з підходів до оптимізації роботи TCP у разі множинних втрат пакетів є механізм, подібний до повільного старту, за якого таймер повторної передачі скидається при кожному частковому підтвердженні. Однак ефективність такого підходу є обмеженою за відсутності підтримки опції SACK.

У деяких реалізаціях у відповідь на часткове підтвердження може виконуватися повторна передача більш ніж одного сегмента, при цьому таймер повторної передачі скидається лише після повторної відправки. У разі множинних втрат передавання двох сегментів у відповідь на часткове підтвердження забезпечує швидше відновлення роботи з’єднання, ніж підхід з очікуванням N RTT, де N — кількість втрачених пакетів.

За відсутності опції SACK швидке відновлення забезпечується за рахунок переходу до повільного старту, що дозволяє відновити передачу без надмірної кількості повторних сегментів.

В алгоритмах швидкої повторної передачі та швидкого відновлення, зокрема в TCP Reno, проблема множинних повторних передач є менш вираженою, ніж у TCP Tahoe, який не використовує режим швидкого відновлення. Проте зайві повторні передачі можуть виникати і в Reno, і в інших алгоритмах за відсутності додаткових механізмів контролю.

Алгоритм, описаний у стандарті RFC 3782, відповідає варіанту Careful для TCP NewReno, наведеному в RFC 2582, та дозволяє усунути проблему множинних повторних передач. Для цього використовується змінна recover, яка зберігає початковий порядковий номер сегмента, відправленого до входу в режим швидкого відновлення.

Після кожного таймауту повторної передачі в змінну recover записується найбільший порядковий номер переданого сегмента. Якщо після цього відправник знову передає сегменти та отримує три дубльовані підтвердження, що перекривають діапазон, визначений recover, такі підтвердження не трактуються як ознака перевантаження. Вони лише сигналізують про повторну передачу вже надісланих сегментів.

У випадку, коли повторно передані сегменти також втрачаються, відправник отримує дубльовані підтвердження, що перекривають менший діапазон, ніж визначено змінною recover. У такій ситуації TCP не пов’язує ці підтвердження з новими втратами сегментів.

Сучасні підходи також використовують часові мітки та переваги кумулятивних підтверджень, що дозволяє відправнику відрізняти три дубльовані підтвердження, спричинені втратою нових сегментів, від підтверджень, пов’язаних із повторною передачею. Для цього застосовуються евристичні методи, які дозволяють

приймати рішення щодо запуску режиму швидкої повторної передачі навіть у випадках, коли дубльовані ACK не повністю перекривають діапазон, заданий змінною recover.

4.3 TCP Vegas

В алгоритмі TCP Vegas керування розміром вікна здійснюється на основі безперервного моніторингу часу кругового обходу (RTT) для сегментів, які були раніше передані відправником. На відміну від класичних алгоритмів TCP, що реагують на втрати пакетів, TCP Vegas використовує проактивний підхід, орієнтований на аналіз затримок у мережі.

Якщо в процесі передавання даних фіксується зростання значення RTT, відправник робить висновок про наближення мережі до стану перевантаження та ініціює зменшення розміру вікна перевантаження. У разі, коли значення RTT зменшується, це інтерпретується як ознака зниження навантаження в мережі, після чого відправник поступово збільшує розмір вікна.

Таким чином, TCP Vegas прагне підтримувати розмір вікна на рівні, який відповідає оптимальному режиму роботи мережі, запобігаючи як перевантаженню, так і неефективному використанню пропускної здатності. В ідеальному випадку розмір вікна стабілізується поблизу бажаного значення, яке забезпечує баланс між швидкістю передачі та затримками.

Зокрема, у фазі уникнення перевантаження (Congestion Avoidance) алгоритм TCP Vegas визначає розмір вікна на основі різниці між очікуваною та фактичною швидкістю передачі даних, що обчислюється з використанням мінімального та поточного значень RTT.

$$cwnd(t+t_A)=\begin{cases} cwnd(t)+1, & if \quad diff < \dfrac{a}{base_rtt} \\ cwnd(t); & \dfrac{a}{base_rtt} \le diff \le \dfrac{\beta}{base_rtt}; \quad diff = \dfrac{cwnd(t)}{base_rtt} - \dfrac{cwnd(t)}{rtt}, \\ cwnd(t)-1, & \dfrac{\beta}{base_rtt} < diff \end{cases} \qquad (4.7)$$

де rtt [сек] зареєстроване RTT;

$base_rtt$ [сек] найменше в даному циклі значення RTT.

Ця модифікація TCP вимагає високого дозволу таймера відправника.

4.4 Compaund TCP

Compound TCP (CTCP) - алгоритм, розроблений Microsoft і вперше включений в мережевий стек Windows Vista і Window Server 2008 TCP. Розроблений для агресивного збільшення TCP вікна (congestion window) для оптимізації TCP швидкості передачі даних при великій затримці (en: Bandwidth-delay_product) з мінімальними відхиленнями від стандарту. Реалізація існує для Linux, а також для Windows XP і Windows Server 2003.

Також як TCP Vegas, Compound TCP використовує прогноз затримки черги як міру завантаження каналу; якщо затримка мала, вважається, що перевантаження на шляху прямування пакетів немає, і швидкість різко підвищується. На відміну від Vegas, алгоритм не намагається підтримувати фіксовану кількість пакетів в черзі.

Compound TCP працює з двома вікнами: звичайне AIMD вікно і залежне від затримки. Розмір ковзного вікна розраховується як сума цих двох вікон. Вікно AIMD збільшується так само як в алгоритмі TCP Reno. При маленькій затримки швидко збільшується розмір вікна затримки для підвищення ефективності використання пропускної здатності мережі. При попаданні пакета в чергу поступово зменшується розмір вікна затримки щоб компенсувати збільшення основного (AIMD) вікна, як показано на рис. 4.1. Мета цього - підтримувати суму вікон в районі деякої константи, спрогнозованої алгоритмом (en: bandwidth-delay

product шляху). Зокрема, при попаданні пакета в чергу, вікно затримки зменшується на прогнозований розмір черги для уникнення проблеми "постійного перевантаження" (persistent congestion), що виникає в Vegas.

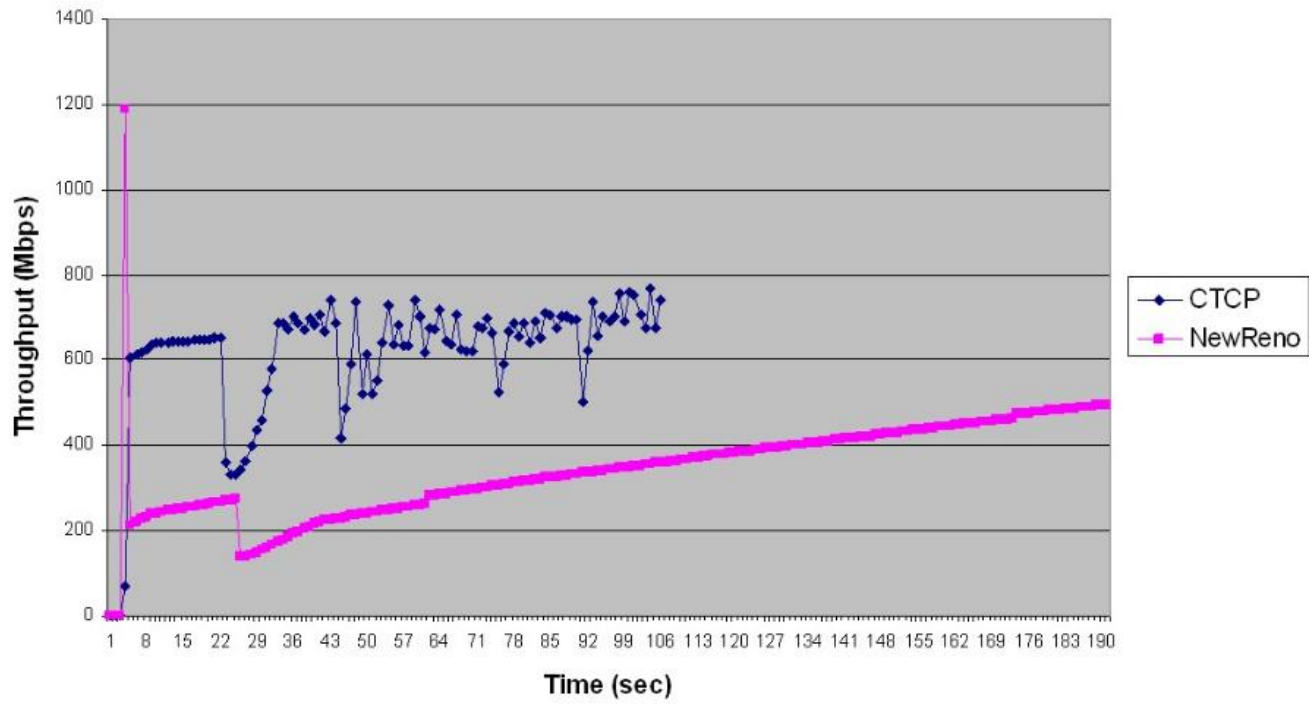


Рис. 4.1 Порівняння роботи NewReno та Compound

4.5 TCP Tahoe

Алгоритм типу TCP-Tahoe є одним з найстаріших і широко поширених. Даний алгоритм був винайдений Джакобсоном в 1988 році, більш вдосконалений уже набагато пізніше.

Якщо ж буфер буде переповнений, то якесь число сегментів загубиться. При таких діях буде запущено кілька сценаріїв. Основний сценарій - це повільний старт, який запускається в рамках класичного алгоритму TCP-Tahoe, коли відбувається втрата сегмента і зв'язаних з ним значенням таймаута (RTO) у відправника, так як відправник не одержує сигналу підтвердження типу ACK для втраченого сегмента. Сценарій повільного старту є установкою вікна перевантаження (CWND) рівним значенню 1 і значення порога повільного старту (ssthresh) рівним половині показника CWND, при якому стався цей таймаут. Скорочення типу CWND до

одиниці відбувається через малу інформованості відправника про стан мережі. Після кожного підтвердження типу повинна виконуватися умова: CWNDi + 1 = CWNDi + 1. Ця формула працює до тих пір, поки значення показника CWND не стане рівним значенню типу ssthresh. Зростання показника CWNDпосле чого стає лінійним. Зміст даного алгоритму в тому, щоб утримати значення CWND в області максимально можливих значень. Така оптимізація здійснюється за рахунок втрати пакетів. Якщо ж такої втрати пакетів не відбувається, то значення типу CWND може досягти значення window за замовчуванням, що задається при конфігурації TCP-драйвера. На рис. 4.2. показана еволюція CWND при втраті пакетів.

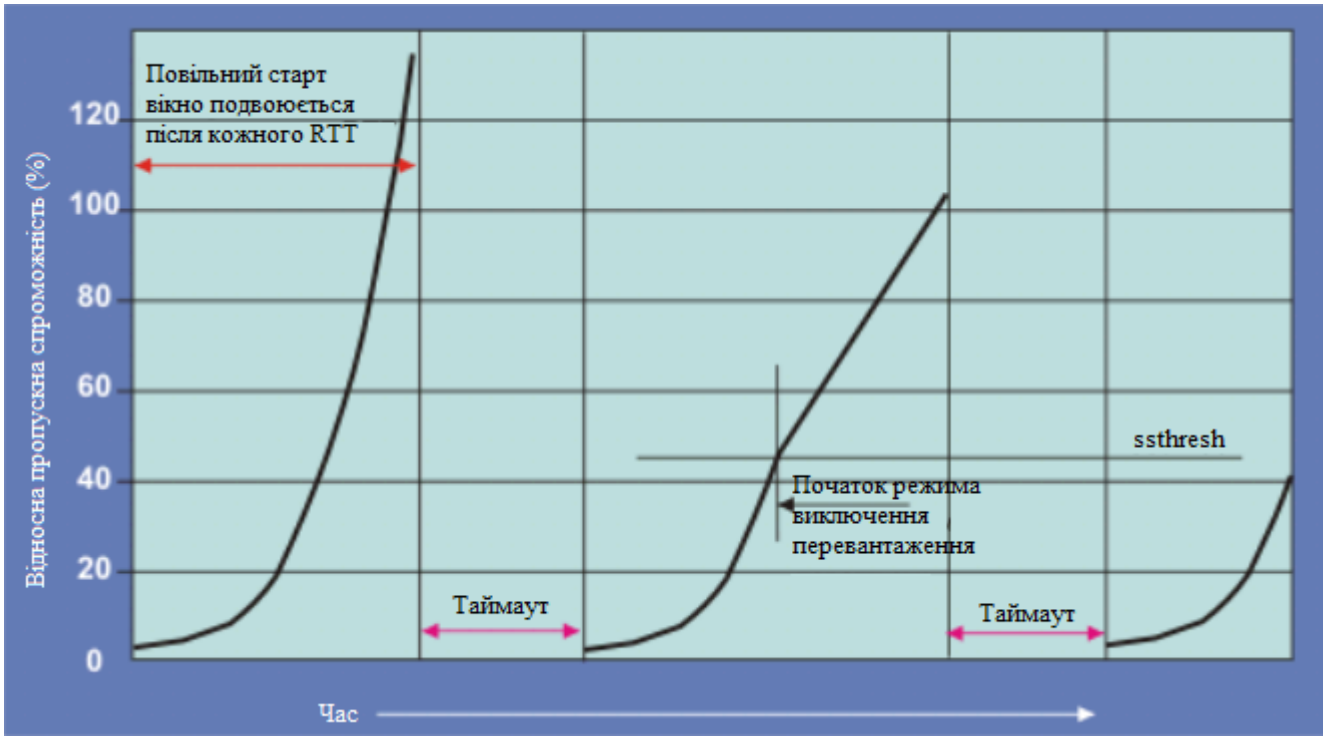


Рис. 4.2 Еволюція CWND при втраті пакетів Значення таймаута

Показник RTO обчислюється за формулою:

$$RTO = \overline{RTT} + 4\sigma_{\overline{RTT}}, \tag{4.8}$$

В даному випадку, втрачений пакет і всі пакети, які будуть надіслані після нього, (незалежно від того, підтверджено їх отримання чи ні) будуть пересилатися

повторно. При високій ймовірності втрати пакетів це знижує пропускну здатність і досить сильно збільшує завантаження каналу і без того високу.

Так чому ж при втраті пакету показник CWND робиться рівним 1, а не CWND-1 або CWND / 2? Це пов'язано з тим, що ефективність каналу максимальна в разі найбільшого значення показника CWND. Якщо буде здійснена втрата пакета через переповнення буфера, оптимальне значення показника CWND може бути вибрано лише при точному значенні стану віртуального каналу. Дана інформація найчастіше є недоступною, тому система переходить в режим звільнення буфера зі значенням показника рівним 1 (CWND = 1). Якби втрата пакетів була пов'язана з початком сесії обміну з конкуруючим клієнтом, то операція була б дорівнює показнику CWND = CWND-1, але проблема б не була вирішена, втрата пакета викликала б значення тайм-ауту і канал був би блокований мінімум на 1 секунду, що викликало б різке падіння швидкості передачі.

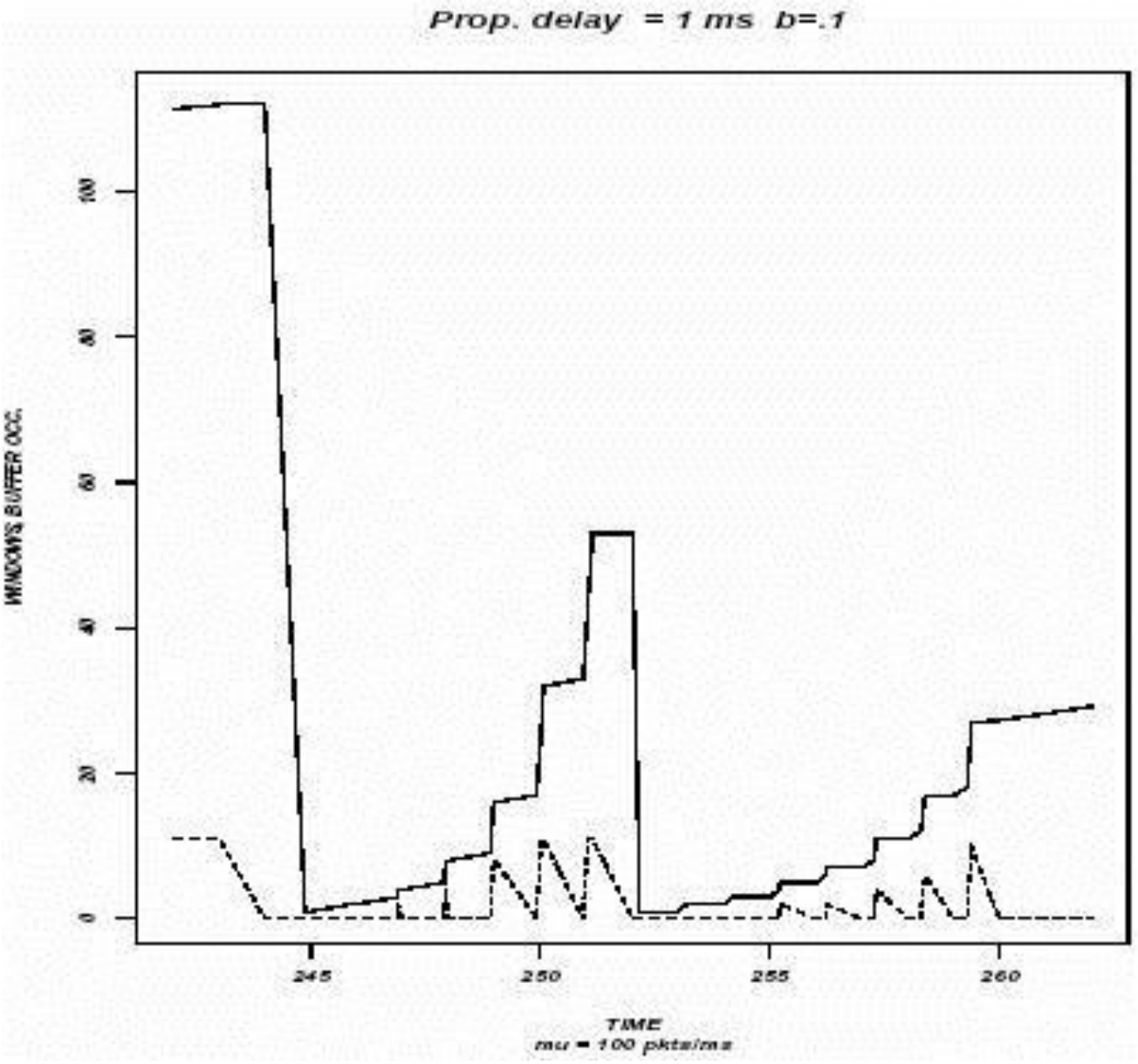


Рис. 4.3. - Еволюція cwnd при двох повільних стартах

При використанні стартового показника зі значенням CWND> 1 можна збільшити ефективність віртуального TCP-каналу. Так само стартове значення CWND = 2 * MSS може бути встановлено. Зрозуміло, що в критичних ситуаціях при значенні CWND = 1 має бути дозволено.

4.6 TCP Westwood

Серед існуючих моделей реалізації протоколу TCP доцільно виділити модель TCP Westwood (TCPW). Дана модель забезпечує більш ефективне використання пропускної здатності каналу завдяки модифікації алгоритму керування вікном перевантаження, що ґрунтується на оцінюванні швидкості передавання даних (rate estimation, RE) та поточного значення доступної смуги пропускання.

На основі отриманих оцінок обчислюються значення параметрів cwnd та ssthresh. Для з'єднань з великим добутком пропускної здатності на затримку (bandwidth-delay product) дана модель може демонструвати кращі результати порівняно з TCP NewReno.

У моделі TCP Westwood параметр seg_size визначає довжину TCP-сегмента у бітах, dupack — кількість дубльованих підтверджень, a be (bandwidth estimation) — оцінку пропускної здатності каналу. Після отримання *n* дубльованих підтверджень здійснюється повторна передача втраченого сегмента, аналогічно до механізму fast retransmit у TCP Reno.

Під час швидкого відновлення вікно перевантаження зростає до значення ssthresh, яке визначається на основі оцінки пропускної здатності мережі за формулою:

$$window = be \cdot rttmin$$

де rtt_min — мінімальне значення часу кругового обходу, зафіксоване під час сеансу.

Після переходу до фази уникнення перевантаження здійснюється плавний пошук доступної пропускної здатності каналу. Значення cwnd встановлюється рівним ssthresh лише у випадку, якщо cwnd > ssthresh. Оцінювання параметра be виконується на основі частоти надходження підтверджень ACK та обсягу даних, доставлених за одиницю часу.

Об’єкти мережевого з’єднання виявляються частково синхронізованими, що пов’язано зі зростанням вікна перевантаження при заповненні буферів мережевих пристроїв. У таких умовах частина пакетів може бути відкинута. Окрім механізму тайм-ауту, у TCP передбачено виявлення втрат сегментів за допомогою дубльованих підтверджень ACK. Отримання трьох дубльованих підтверджень розглядається як надійний сигнал втрати пакета та ініціює механізм швидкої повторної передачі.

У режимі fast recovery при втраті одного пакета значення cwnd встановлюється на три сегменти більше за ssthresh, після чого при отриманні нового підтвердження cwnd зменшується до ssthresh і далі зростає лінійно. Такий режим відомий як congestion avoidance або AIMD (additive increase, multiplicative decrease).

У мережі може виникнути ситуація, коли перевантажений комутатор підтримує механізм зворотного тиску та надсилає повідомлення ICMP про перевантаження. У відповідь відправник знижує швидкість передачі даних щонайменше вдвічі. Реакція на ICMP-повідомлення залежить від конкретної реалізації TCP-драйвера.

Окрім керування перевантаженням з боку відправника, у TCP передбачено механізм керування потоком з боку одержувача. Одержувач у кожному підтвердженні ACK передає значення параметра window, яке визначає обсяг даних, що може бути прийнятий. Даний механізм особливо важливий для мереж із великим значенням RTT, зокрема супутникових каналів.

Оптимізація TCP-з’єднань може здійснюватися у двох напрямах: шляхом удосконалення самого протоколу або шляхом модифікації мережевого середовища. При цьому всі зміни TCP мають забезпечувати зворотну сумісність з існуючими реалізаціями, що значною мірою стримує радикальні зміни протоколу.

Альтернативними підходами є використання протоколів UDP або T/TCP, який оптимізований для коротких транзакцій. Проте T/TCP потребує змін як на стороні

клієнта, так і сервера та має обмеження з точки зору безпеки, що ускладнює його широке впровадження.

Особливі труднощі виникають при використанні TCP у супутникових каналах зв’язку, де значення RTT перевищує 250 мс, а рівень бітових помилок є високим. У таких умовах час відновлення після перевантаження може сягати десятків секунд, що робить короткі TCP-сесії малоефективними. Частково ця проблема компенсується використанням кількох паралельних TCP-з’єднань.

4.7 TCP Hybla

Версія алгоритму TCP Hybla, яка була розроблена на початку 2003 року, дозволяє виключати витрати в TCP-з'єднання, які включають ділянки каналів з великою затримкою. Дане досягається в результаті проведення аналітичної оцінки динаміки поведінки вікна перевантаження, яка пропонує різні модифікації, які виключають залежність від параметра RTT.

Модель TCP Hyblaето деяка удосконалена модифікація протоколу TCP, який блокує погіршення тих робочих характеристик з'єднання при яких значення параметра RTT збільшується, що типово для супутникових каналів зв'язку. Протокол складається з наступних фаз і процедур:

- фаза поліпшення стандартного алгоритму з управління перевантаженнями;
- процедура обов'язкового використання політики типу SACK;
- процедура використання тимчасових міток;
- процедури прийняття оцінки поточної смуги каналу Ное;
- фази використання техніки позиціонування пакетів;

Діючі на сьогоднішній момент версії протоколу TCP, такі як Tahoe чи Reno, а також NewReno мають проблеми, які пов'язані з поганою роботою характеристикою при великих значеннях затримок в каналі зв'язку, що досить часто зустрічається в наземних або супутникових радіо каналах зв'язку. Протокол TCP був спочатку створений для протидій з перевантаженнями, а так само в боротьбі з

втратами TCP-сегментів через перекручування при передачі, помилково приписувані переповнення свого буфера, запускаються механізми виключення такого перевантаження, які в такій же ситуації приносять тільки шкоду .

Дана проблема вирішується як шляхом відповідної модифікації протоколу TCP, так і за допомогою підвищення надійності нижче лежачого протокольного шару. У разі відсутності помилок в каналі зв'язку, радіо з'єднання часто характеризуються великими показаннями затримок. Це пов'язано з алгоритмом передачі, який заснований на алгоритмі CWND, який запускається шляхом приходу пакетів типу ACK, завжди залежать тільки від мережевих затримок. Велике число часу для параметра RTT стримує зростання вікна його перевантаження і призводять до значного зменшення значення його пропускної здатності і поганого перерозподілу його смуги пропускання між різними типами з'єднання. В даному випадку при деградації робочих параметрів властивостями TCP-протоколу неможливо усунути дану проблему без істотного поліпшення алгоритму управління поточного перевантаження. Це завдання і вирішує алгоритм TCP Hybla.

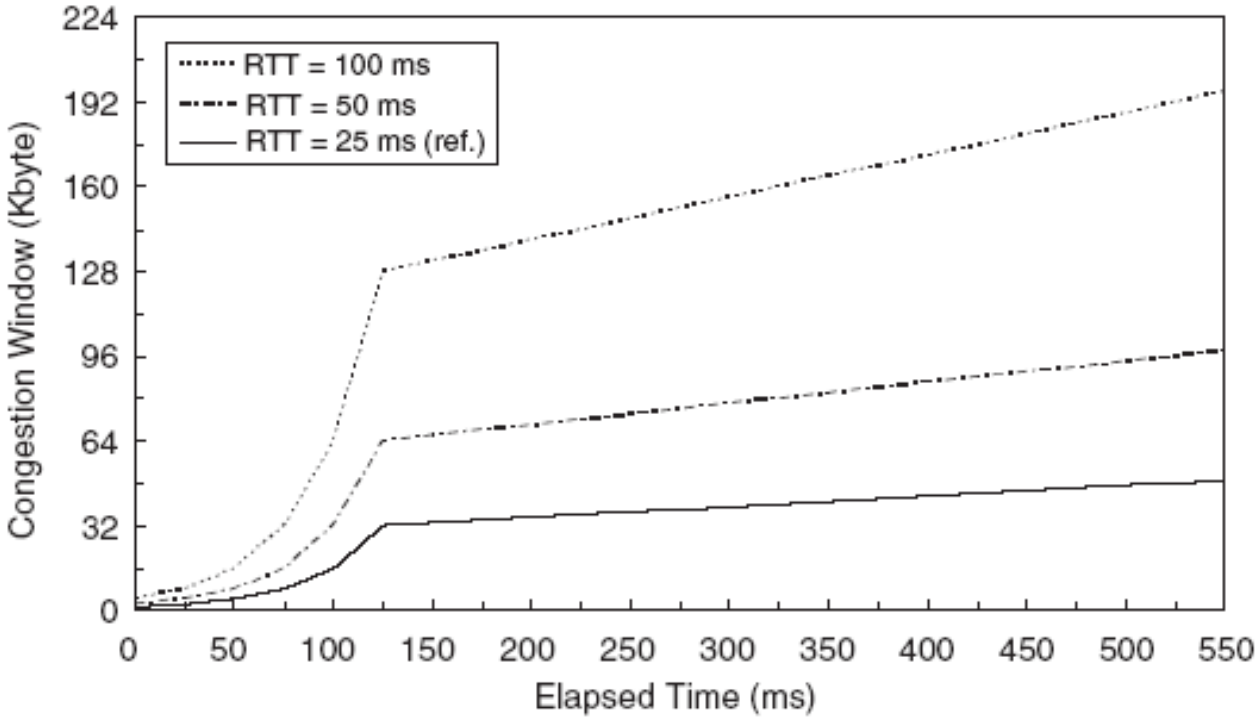


Рис. 4.4 Еволюція вікна перевантаження з часом для кількох значень RTT для TCP Hybla

Основою для даного алгоритму TCP Hybla послужило досягнення для з'єднань з великим значенням тих же швидкостей передачі, B (t), що і для провідних TCP-каналів.

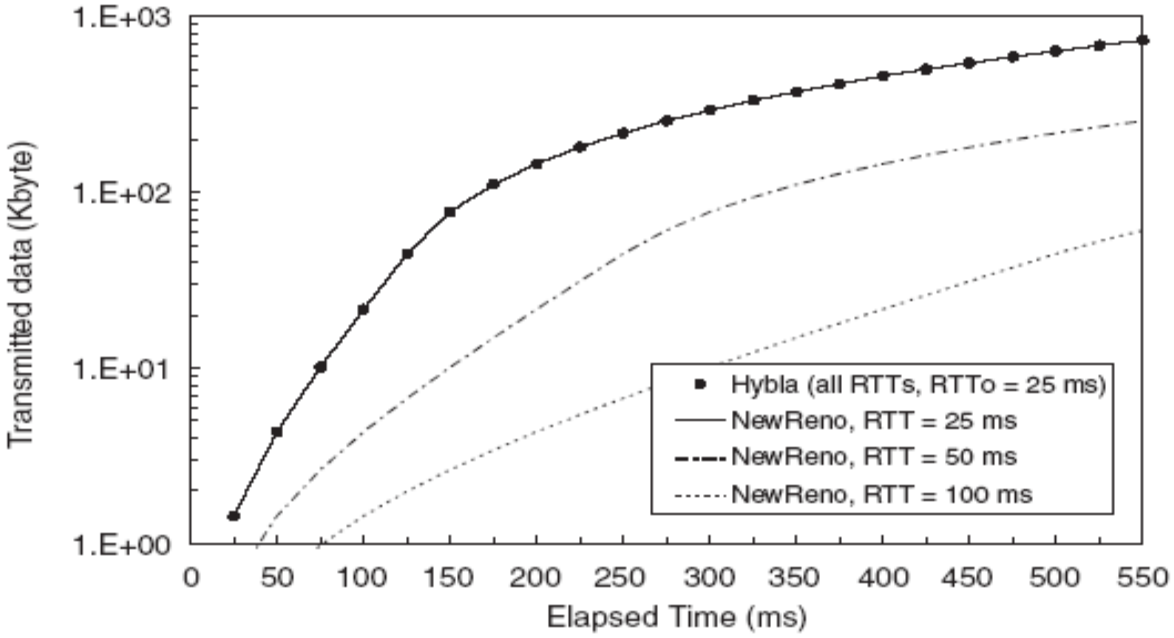


Рис. 4.5 Процес передачі даних при декількох значеннях параметра RTT для каналу без втрат в порівнянні з іншими алгоритмами

Дані співвідношення B (t) = W (t) / RTT, W (t) -де вікно перевантаження, яке показує, що мета може бути досягнута в два етапи: по-перше, зробивши параметр W (t), незалежно від значення RTT , за рахунок зміни тимчасового масштабу, по-друге, компенсувавши ефект поділу на параметр RTT.

З малюнка видно, що алгоритм TCP Hybla поводиться не залежно від реального значення його параметра показника RTT. Дана дія ще раз доведено числовими значеннями, які представлені на рисунку 4.5., де показані дані в порівнянні зі стандартним алгоритмом TCP і алгоритмом TCP Hybla при різних значень показника RTT, за умови відсутності втрат. Обсяг переданих даних в стандартному TCP-з'єднання знижується зі збільшенням показань значення RTT, TCP Hybla демонструє характеристики близькі до еталонних.

4.8 Алгоритми BIC-TCP і CUBIC-TCP

Алгоритм BIC-TCP забезпечує досить велику масштабованість для швидкісних мереж передачі даних, а так само еквівалентність для конкуруючих потоків даних і її стабільність сбудучі низьким рівнем осциляції розміру його вікна. Однак дана функція зростання вікна моделі BIC-TCP буде досить агресивною для протоколу TCP, особливо при малих значеннях паказателі RTT або для низькошвидкісних мереж зв'язку. Більш того, існування кількох фаз (двійкове пошукове зростання, пошук max, Smax і Smin) управління його вікном додає якусь складність в реалізації даного протоколу та аналіз його робочих характеристик. Вже згадана модель типу CUBIC TCP позбавлена багатьох недоліків в порівнянні з BIC TCP.

У моделі типу CUBIC використовується вже кубічний функція зростання його вікна, яка за формою дуже близька до відповідної функції моделі BIC-TCP. У моделі CUBIC при реалізації функції зростання його вікна використовується той же час, яке проходить пакет з моменту останнього події перевантаження. У той же час як більшість моделей реалізації стандартного протоколу TCP використовують опуклі функції зростання після втрати таких пакету, модель же CUBIC використовує як опуклі, так і увігнуті профілі кубічної функції зростання вікна. На рис. 4.6. показана еволюція вікна в BIC (a) і CUBIC (b).

Після зменшення розміру його вікна при втраті даних пакета, записується значення вікна Wmax, яке воно мало в момент втрати, і зменшується розмір вікна в n раз, де n одно постійної зменшення вікна TCP. Після цього система переходить з режиму швидкого відновлення в режим виключення перевантаження, починається збільшення вікна з використанням увігнутого профілю кубічної цієї функції. Кубічна функція параметиризована таким чином, що її плато доводилося на Wmax, так що увігнутий зростання триває до тих пір поки ширина вікна не стане рівним Wmax. Після цього дана кубічна функція стає опуклою. Такий спосіб підстроювання ширини вікна (увігнутий, а потім опуклий) покращує протокол і мережеву стабільність, зберігаючи високий рівень використання смуги каналу. Це

відбувається зі збільшенням розміру вікна і залишається майже постійним, утворюючи плато в області Wmax, де рівень використання мережі вважається високим і стабільним.

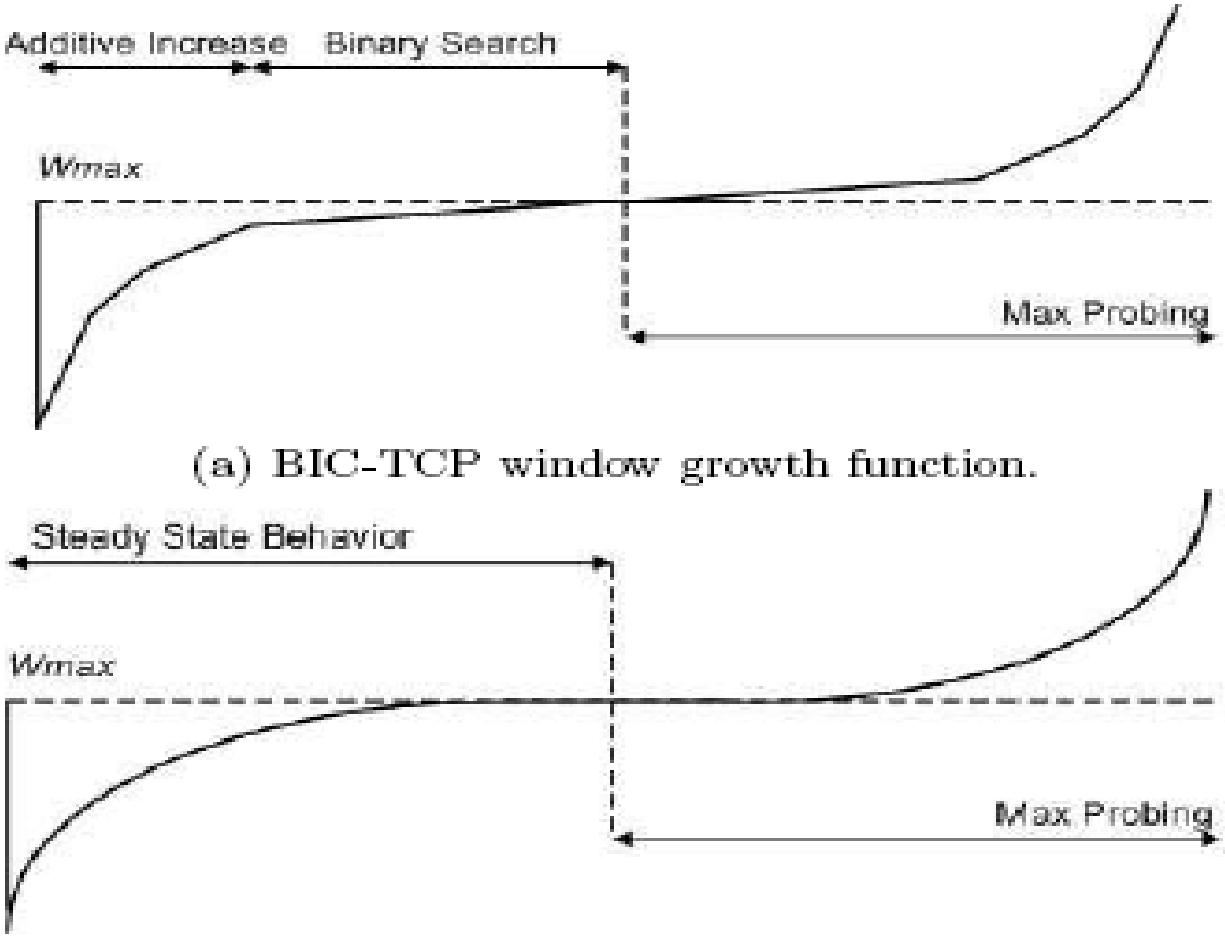


Рисунок 4.6 Функція зростання вікна для BIC-TCP і CUBIC

Збільшення значення вікна в моделі CUBIC здійснюється відповідно до вираження

$$W(t) = C(t-K)3 + W\max, \tag{4.9}$$

де C - параметр CUBIC;

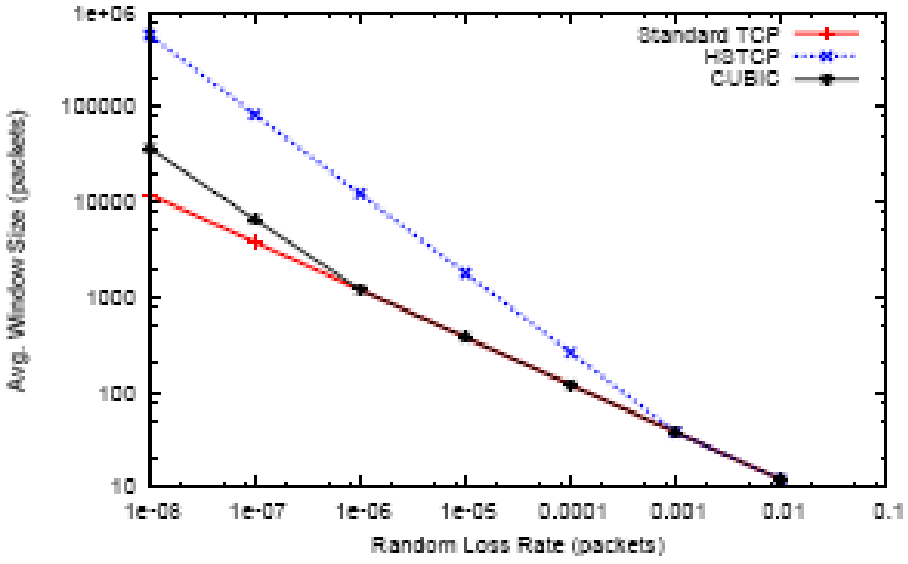
t - час з моменту останнього зменшення ширини вікна;

K - одно періоду часу, який необхідний для збільшення W до Wmax, його значення обчислюється з залученням формули

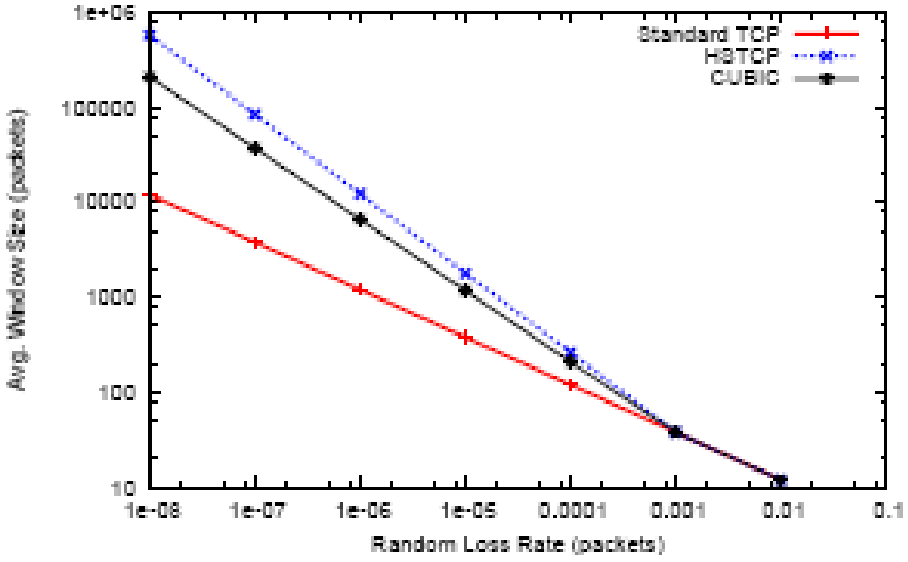
$$K = \sqrt[3]{\frac{W\max \cdot \beta}{C}}, \tag{4.10}$$

При отриманні підтвердження типу ACK на фазі виключення перевантаження, алгоритм CUBIC обчислює швидкість росту ширини вікна за час подальшого RTT-періоду. Програма встановлює параметр із значенням W (t + RTT) в якості ймовірної величини вікна перевантаження. Припустимо, що ширина такого вікна дорівнює значенню cwnd. Залежно від цього значення, параметр CUBIC реалізує три різних режиму. Якщо параметр cwnd буде меньший ніж розмір вікна, який був би досягнутий в стандартному TCP, через час t після останнього випадку втрати, тоді параметр CUBIC виявляється в TCP-режимі. В іншому випадку, якщо параметр cwnd менше Wmax, тоді параметр CUBIC знаходиться в увігнутій області, і якщо параметр cwnd більше значення Wmax, параметр CUBIC знаходиться в опуклій області функції зростання.

Якщо параметр cwnd менше значення Wtcp (t), то тоді протокол знаходиться в режимі TCP, параметр cwnd робиться рівним значенню Wtcp (t) при кожному отриманні підтвердження типу ACK. Рисунок 4.7. дозволяє порівняти характеристики стандартної моделі протоколу TCP з такими моделями як HSTCP і CUBIC.



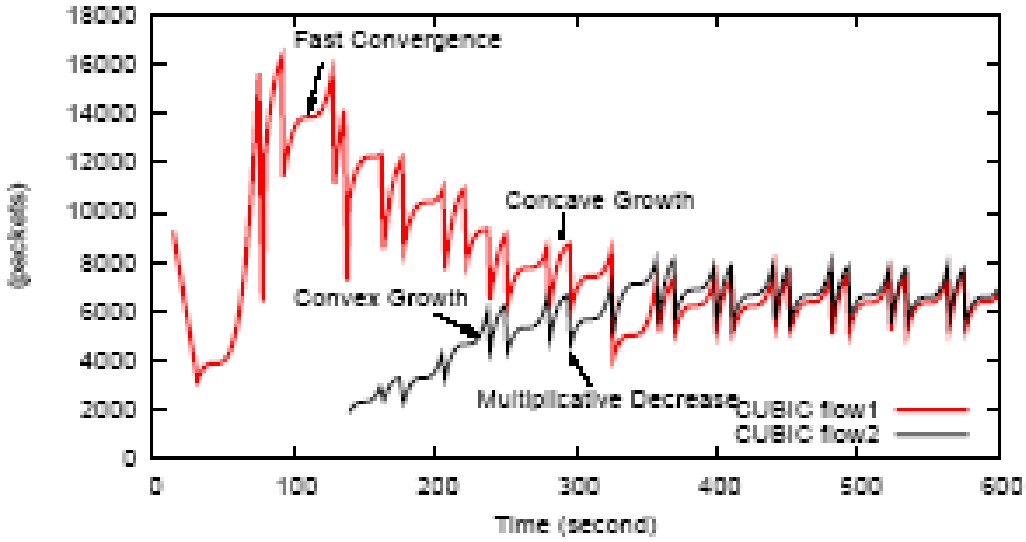
(a) Networks with 10ms RTT.



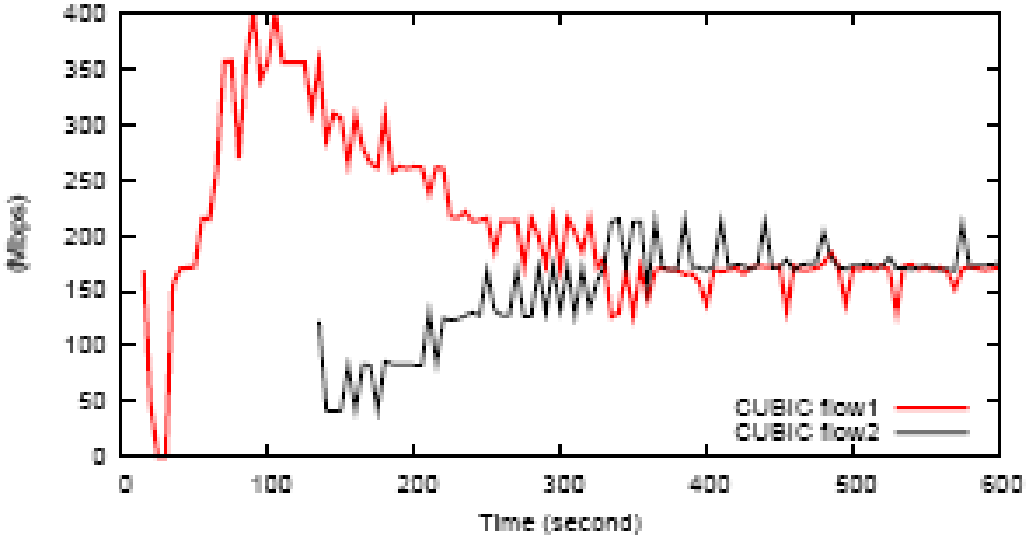
(b) Networks with 100ms RTT.

Рис. 4.7 Функції відгуку для стандартів TCP, HSTCP і CUBIC в мережах з RTT 10 мсек (a) і 100 мсек (b) відповідно.

На рис. 4.9 показано, як можуть конкурувати один з одним два потоку CUBIC при RTT = 246 мсек.



(a) CUBIC window curves.



(b) Throughput of two CUBIC flows.

Рис. 4.9 Two CUBIC flows with 246ms RTT

Середнє значення вікна в CUBIC можна оцінити з формули:

$$\mathrm{E}\{W_{cubic}\} = \sqrt[4]{\frac{C\cdot(4-\beta)}{4\cdot\beta}\cdot\left(\frac{RTT}{p}\right)^3}, \qquad (4.11)$$

4.9 Алгоритм TCP RED/WRED

З поведінкою TCP-джерел в моменти роботи алгоритму “відкидання хвоста” зв'язана необхідність проведення превентивного управління чергою з метою сигналізації про перевантаження мережі до фактичного переповнювання черги і контролю за розміром черги для зниження затримки обробки пакетів. Алгоритм довільного раннього виявлення (RED) володіє істотними перевагами в порівнянні з традиційним механізмом “відкидання хвоста”.

Хоча протокол TCP використовує з'єднання по типу точка-точка, все одно є можливість спробувати поліпшити робочі характеристики каналу, оптимізуючи алгоритм управління чергами. Одним з підходів до такого є алгоритм RED (RandomEarlyDetection). Даний алгоритм RED дозволяє маршрутизатора відкидати пакети, якщо навіть в черзі ще є місце.

Алгоритм RED використовує метод зваженого значення довжини черги в якості фактора, що визначає ймовірність відкидання пакету. Зі збільшенням середньої довжини черги збільшується і ймовірність відкидання пакетів. Якщо ж середня довжина черги менше певного порогу, то ймовірність відкидання пакету дорівнюватиме нулю. Невеликі кластери пакетів можуть успішно пройти через фільтр алгоритму RED. Більше ж великі кластери можуть бути втрачені. Дана дія призводить до того, що TCP-сесії з великими відкритими вікнами мають велику ймовірність відкидання таких пакетів. Головною метою для алгоритму RED є виключення такої ситуація, коли кілька TCP-потоків перевантажуються майже одночасно і потім синхронно починають процедуру відновлення.

Алгоритм RED організовує черзі таким чином, що їх розмір відстежує осциляції величини RTT. Алгоритм RED збільшує кількість коротких перевантажень і уникає довгих. Завдання алгоритму RED полягає в тому, щоб повідомити відправнику про можливість перевантаження, і відправник повинен адаптуватися до цієї ситуації.

При визначенні імовірності відкидання пакетів механізм RED обчислює не поточний, а експоненціально зважений середній розмір черги. Поточний середній розмір черги визначається на підставі попереднього середнього і поточного дійсного розміру. Використання механізмом RED середнього розміру черги обумовлено прагненням реагувати лише на тривале перевантаження мережі і “не помічати” моментальних сплесків трафіку.

Середній розмір черги обчислюється за формулою:

$$Q_{\text{сред}} = Q'_{\text{сред}} \cdot \left(1 - \frac{1}{2^n}\right) + Q_{\text{тек}} \cdot \frac{1}{2^n}, \qquad (4.12)$$

де $Q'_{\text{сред}}$ - попередній середній розмір черги;

$Q_{\text{тек}}$ - поточний розмір черги;

n - експоненціальний ваговий коефіцієнт, визначуваний користувачем.

Експоненціальний ваговий коефіцієнт є ключовим параметром, який визначає відносний вклад попереднього середнього і поточного розміру черги в новий середній розмір черги. Практика показала, що найбільш прийнятним значенням експоненціального вагового коефіцієнта *n* є 9. Збільшення експоненціального вагового коефіцієнта приведе до домінування попереднього середнього розміру черги над її поточним розміром в аспекті обчислення нового середнього розміру черги. Навпаки, зменшення експоненціального вагового коефіцієнта приведе до зростання значущості поточного розміру черги при обчисленні її нового середнього розміру.

Велике значення коефіцієнта n обумовлює математичну близькість нового і попереднього середнього розміру черги, а також дозволяє механізму RED стриманіше реагувати на моментальні зміни в її поточному розмірі, що виражається в наступній поведінці.

- Значення середнього розміру черги змінюється повільно, для нього украй нехарактерні різкі скачки. Механізм RED досить стримано

відноситься до часових сплесків трафіку, прагнучи вирівняти поточний розмір черги.

- Механізм RED не квапиться ініціювати процес відкидання пакетів, який, проте, може продовжуватися деякий час після зниження дійсного розміру черги нижче за мінімальне порогове значення.

- Якщо значення коефіцієнта *n* дуже велике, механізм RED може і зовсім перестати реагувати на перевантаження мережі, оскільки в цьому випадку поточний розмір черги практично не впливатиме на обчислення її середнього розміру. Пакети передаватимуться або відкидатимуться так, як якби механізм RED і зовсім не використовувався.

Маленьке значення коефіцієнта n обумовлює математичну близькість нового середнього і поточного розміру черги, що виражається в наступній поведінці механізму RED.

- Середній розмір черги змінюється дуже швидко, для нього характерна сильна залежність від флуктуацій потоку трафіку.

- Механізм RED негайно реагує на довгу чергу, проте як тільки її розмір виявляється нижчим за мінімальне порогове значення, відкидання пакетів припиняється.

- Якщо значення коефіцієнта *n* дуже мало, механізм RED починає дуже гостро реагувати на тимчасові сплески трафіку, що виражається в невиправданому відкиданні пакетів.

Імовірність відкидання пакетів є функцією, залежною від середнього розміру черги, мінімального порогового значення, максимального порогового значення і знаменника граничної імовірності (mark probability denominator), що визначає частину відкиданих пакетів при досягненні середнім розміром черги максимального порогового значення. Наприклад, якщо знаменник (дільник) граничної імовірності дорівнює 10, то при досягненні середнім розміром черги максимального порогового значення механізм RED відкидатиме 1 з 10 пакетів. Нижче приведена формула, по якій розраховується імовірність відкидання пакетів.

$$P_{drop} = \left(\frac{Q_{\text{ср}} - Q_{\min}}{Q_{\max}} \right) \cdot p' \quad , \tag{4.13}$$

де P_{drop} - імовірність відкидання пакетів;

$Q_{\text{ср}}$ - середній розмір черги;

$Q_{\min}$ - мінімальне порогове значення розміру черги;

$Q_{\max}$ - максимальне порогове значення розміру черги;

p' - знаменник (дільник) граничної імовірності.

Коли середній розмір черги перевищує мінімальне порогове значення, механізм RED починає відкидати пакети. Інтенсивність відкидання пакетів зростає прямо пропорційно зростанню середнього розміру черги до тих пір, поки він не досягне максимального порогового значення.

Коли середній розмір черги перевищує максимальне порогове значення, механізм RED відкидає всі пакети, призначені для постановки в чергу. Графік імовірності відкидання пакетів схематично представлений на рис. 4.10

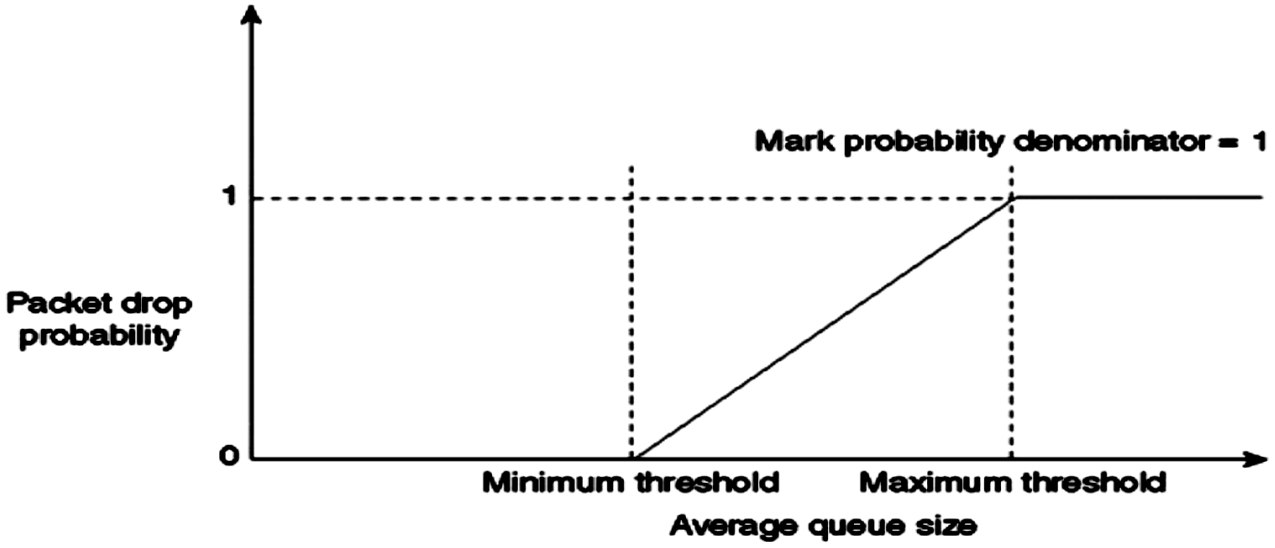


Рис. 4.10 Графік імовірності відкидання пакетів механізмом RED

Існує модифікація алгоритму обробки черг який дав назву WRED (WeightedRandomEarlyDetection), на даний момент широко використовувана в маршрутизаторах. Цей алгоритм застосовується і при реалізації алгоритму DiffServ

і гарантованої переадресації (AF), коли рішення про обробку пакета приймається в кожному транзитному вузлі незалежно (PHB). При цьому може враховуватися код DSCP IP-заголовка.

Реалізація алгоритму зазвичай проводиться таким чином: за умовчанням всі рівні пріоритету трафіку характеризуються однаковим максимальним пороговим значенням і різними мінімальними пороговими значеннями. Таким чином механізм WRED забезпечує інтенсивніше відкидання низькопріоритетних пакетів і менш інтенсивне відкидання високопріоритетних. Стандартне мінімальне порогове значення для трафіку пріоритету 0 складає половину максимального порогового значення. Мінімальне порогове значення механізму WRED, відповідне високопріоритетному трафіку, має бути більшим, ніж мінімальне порогове значення низькопріоритетного трафіку. Виконання цієї умови необхідне для забезпечення строгої черговості відкидання пакетів, згідно якої першими починають відкидатися низькопріоритетні пакети.

Розглянемо функціонування алгоритму WRED на основі потоку.

Алгоритм WRED розрахований на TCP-трафік, якій "реагує" на повідомлення про перевантаження, інакше кажучи, володіє здібністю до адаптації. Але існує ще, наприклад, UDP-трафік, який такою здатністю не володіє і не реагує на повідомлення про перевантаження і, відповідно, не знижує свою інтенсивність. Враховуючи таку поведінку UDP-трафіка, нескладно уявити собі ситуацію, в якій під час перевантаження мережі неадаптивні потоки передають дані з набагато більшою інтенсивністю, чим потоки, що володіють здібністю до адаптації. Отже, на неадаптивні потоки трафіку доводиться більша частка ресурсів в порівнянні з потоками, що знижують свою інтенсивність у відповідь на отримання сигналу про перевантаження.

Алгоритм WRED на основі потоку (flow WRED) є модифікацією алгоритму RED, що передбачає штрафування потоків, що віднімають надмірну долю ресурсів.

З метою забезпечення рівномірного обслуговування активних потоків трафіку механізм WRED класифікує всі встановлювані в чергу пакети залежно від їх пріоритету і потоку трафіку, до якого вони відносяться. Окрім цього, WRED підтримує інформацію про стан всіх активних потоків (active flows), тобто потоків, хоч би один пакет яких поставлений на обробку в яку-небудь з черг.

Інформація про стан активних потоків використовується для визначення справедливої долі виділених потоку ресурсів черги (розмір черги/кількість активних потоків), а також для виявлення і штрафування потоків, що віднімають надмірно великий об'єм ресурсів. Аби механізм WRED адекватніше реагував на сплески потоків трафіку, можна збільшити справедливу долю ресурсів кожного потоку шляхом застосування так званого коефіцієнта масштабування.

Справедлива доля ресурсів для активного потоку трафіку =
розмір черги/кількість активних потоків.

З врахуванням коефіцієнта масштабування:
Справедлива доля ресурсів для активному потоку трафіку =
(розмір черги/кількість активних потоків)*коефіцієнт. масштабування.

Потік, вимоги якого перевищують справедливу долю ресурсів з врахуванням коефіцієнта масштабування, штрафується шляхом збільшення ненульової імовірності відкидання для всіх пакетів, що знов надійшли.

Розглянемо дії, що робляться механізмом WRED на основі потоку у відношенні тільки що поставленого в чергу пакету.

1) При визначенні імовірності відкидання пакету механізм WRED на основі потоку враховує як значення поля IP-пріоритету пакету, так і інформацію про стан активних потоків. Від IP- пріоритету пакету залежать конфігуровані (або стандартні) мінімальне і максимальне порогові значення.

2) Якщо середній розмір черги нижчий за мінімальне порогове значення, то імовірність відкидання пакету встановлюється рівною нулю (іншими словами, цей пакет не буде відкинутий).

3) Якщо ж середній розмір черги знаходиться між мінімальним і максимальним пороговим значенням, то враховується інформація про стан активних потоків трафіку.

4) Якщо пакет належить потоку, що перевищив справедливу долю ресурсів з врахуванням коефіцієнта масштабування, механізм WRED збільшує імовірність відкидання цього пакету шляхом зменшення відповідного максимального порогового значення, як показано нижче.

Нове max порогове значення =

= min поріг. значення + ((max поріг значення - min поріг. значення)/2).

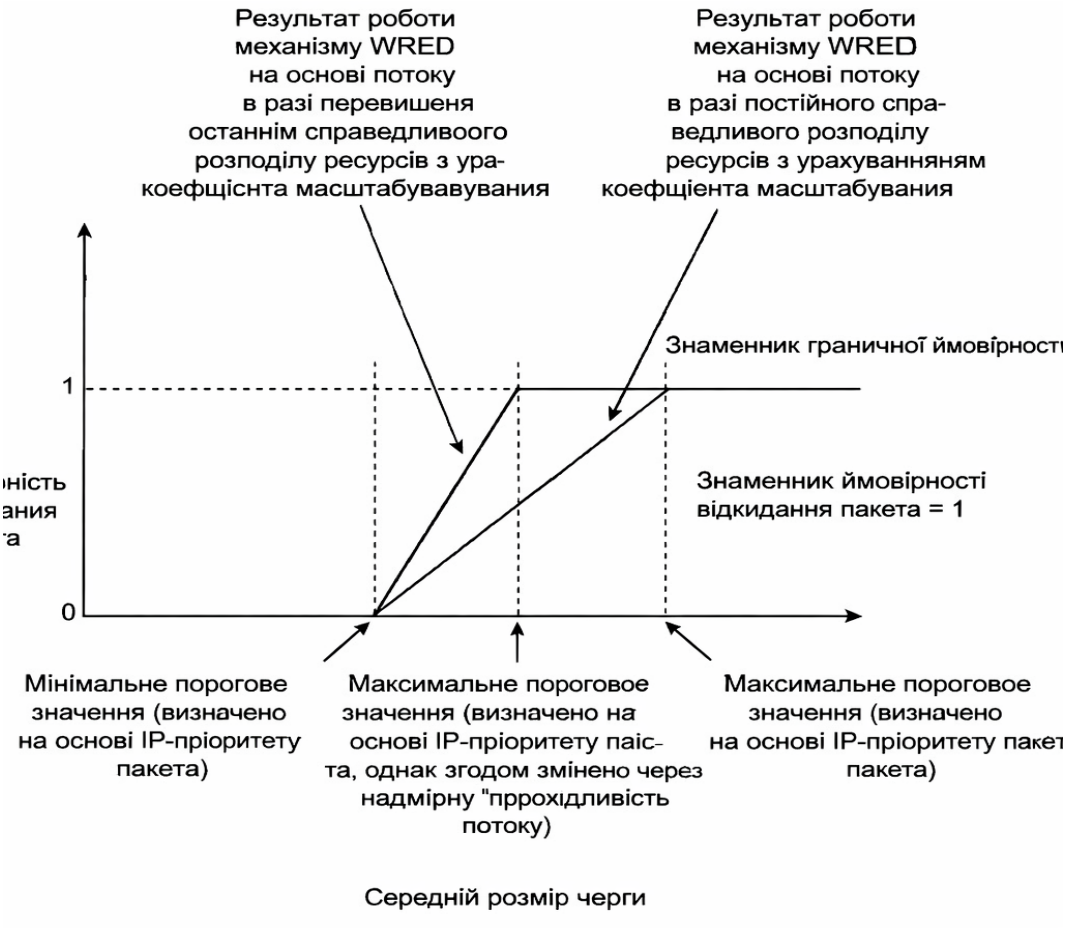


Рис. 4.11 Графік імовірності відкидання пакету при використанні

механізму WRED

Ненульова імовірність відкидання пакету розраховується на підставі мінімального і нового максимального порогового значення. Оскільки результатом зниження максимального порогового значення є істотне збільшення кута нахилу кривої імовірності відкидання (рис. 4.11), шанси пакету бути відкинутим різко зростають.

Якщо ж потік трафіку не перевищує справедливої долі ресурсів з врахуванням коефіцієнта масштабування, то ненульова імовірність відкидання пакету розраховується по стандартному методу.

Коли середній розмір черги перевищує максимальне порогове значення, механізм WRED на основі потоку відкидає всі пакети, призначені для постановки в чергу.

Слід зазначити, що механізм WRED на основі потоку збільшує імовірність відкидання пакетів лише для тих потоків трафіку, чиї вимоги перевищили справедливу долю ресурсів з врахуванням коефіцієнта масштабування. У всьому іншому поведінка механізму WRED на основі потоку аналогічно поведінці стандартного механізму RED.

Механізм WRED на основі потоку встановлює ненульову імовірність відкидання пакетів при знаходженні середнього розміру черги між мінімальним і максимальним пороговими значеннями навіть для тих потоків, які характеризуються всього лише декількома поставленими в чергу пакетами. Можна конфігурувати механізм WRED на основі потоку так, щоб він не відкидав пакети подібних потоків трафіку (вказавши при цьому, яке саме число поставлених в чергу пакетів все ще є “рятівним” для потоку) шляхом установки мінімального порогового значення, близького до максимального порогового значення. При цьому коли середній розмір черги досягне максимального порогового значення, можна застосувати механізм “відкидання хвоста”.

Різниця між максимальним і мінімальним порогами повинна бути досить велика, щоб уникнути глобальної синхронізації TCP агентів (синхронна зміна CWND).

4.10 Метод оптимізації TCP сеансів

В основу даного методу, який пов'язаний з оптимізацією вибору параметрів для протоколу TCP і алгоритмів типу RED і типу WRED, належить таке рішення, яке дозволяє оптимізувати завдання, в процесі якої здійснюється мінімізація і максімізація показників якості. При пошуку компромісу між урахуванням фізики процесів передачі даних в рамках поточних TCP-сеансів, і можливістю отримати чисельні результати розрахунку в рамках відомих таких оптимізаційних процедур, скористаємося критерієм, що визначаючим мінімумом функціоналу

$$J = \min\left[\sum_{i=1}^{M} a_i \cdot \left(\lambda_i^{Трб} - \frac{1}{T}\cdot\int_0^T \lambda_i(t)dt\right)\right], \qquad (4.13)$$

де ai - пріоритет і -го TCP-потоку;

T - це часовий інтервал процесу оптимізації, який має значення схоже із середньою тривалістю TCP-сеансу в мережі.

Використання критерію гарантує мінімізацію зважених щодо пріоритету відмов в обслуговуванні на транспортному рівні. Після перетворень вираз можна привести до виду

$$J = \sum_{i=1}^{M}(a_i \cdot \lambda_i^{Трб}) - \min\left[\sum_{i=1}^{M}\left(\frac{a_i}{T}\cdot\int_0^T \lambda_i(t)dt\right)\right] = \max\left[\sum_{i=1}^{M}\left(\frac{a_i}{T}\cdot\int_0^T \lambda_i(t)dt\right)\right], \qquad (4.14)$$

Фізичний зміст виразу (4.14) полягає в тому, що максимізація середньої сумарно зваженої щодо пріоритету інтенсивності самого TCP-потоку на інтервалі має значення параметра T.

При цьому рішенні такого завдання оптимізації як керовані виступають деякі параметри протоколу TCP, такі як величина параметра MSS, інтервал значенням RTT, розмір такого вікна даної перевантаження, значення порога, режимів передачі, а і такі параметри алгоритмів боротьби з перевантаженнями як RED / WRED. На практиці можна досить просто оптимізувати тільки малу частину з перерахованих нами параметрів, при цьому інші характеристики фіксувати на деяких їх значеннях.

З метою формулювання деяких умов при забезпеченні гарантованої якості обслуговування були введені деякі обмеження за тимчасовими показниками QoS

$$\sum_{j=1}^{K} \tau_i^j(t) \le \tau_i^{Трб}, \tag{4.15}$$

$$\sum_{j=1}^{K} \sigma_i^j(t) \le \sigma_i^{Трб}, \tag{4.16}$$

і по показникам надійності:

$$\prod_{j=1}^{K} P_i^j(t) \le P_i^{Трб}, \tag{4.17}$$

де параметр K визначає кількість ділянок мережі між вузлом-джерелом і вузлом одержувачем.

При такому розкладі в виразі (4.15) формалізована вимога до міжкінцевої, та є сумарною по всіх даних ділянках мережі зв'язку між вузлом-джерелом і вузлом-одержувачем, а так само середньої затримки пакетів в мережі, в вираженні (4.16), а співвідношення (4.17) відображає вимога щодо величини ймовірності відкидання (блокування) пакетів.

Таким чином, реалізація методу оптимізації процесу передачі даних в рамках TCP-сеансів передбачає наступну послідовність дій:

а) моніторинг, а так само збір вихідних даних для передачі про поточні значення TCP-з'єднання, а так само про режим передачі в протоколі TCP, параметрів тих алгоритмів запобігання перевантаження (RED / WRED) на вузлах аналізованої мережі;

б) на основі збору вихідних даних про стан мережі зв'язку встановлюється їх аналітичний взаємозв'язок за допомогою аналізу даної аналітичної моделі (3.5);

в) формалізація таких умов-обмежень за такими параметрами часових показників якості обслуговування QoS (4.15) і (4.16), а так само за показниками її надійності (4.10.5), а також таких умов забезпечення стійкості для TCP-сеансів;

г) постановка задачі оптимізації (4.14), в ході її рішення буде забезпечуватися мінімізація його відмов в обслуговуванні такого трафіку користувача за погодженням буде розраховуватися на основі параметрів протоколу TCP і його алгоритмів боротьби з перевантаженням, наприклад, алгоритмів типу RED / WRED;

д) параметри протоколу TCP і його алгоритмів боротьби з даної перевантаженням, наприклад, алгоритмів типу RED / WRED при автоматичному режимі підлягають якоїсь коригування (налаштування), яка забезпечує, тим самим, оптимізацію TCP-сеансів в цілому;

Періодичний характер розрахунків може бути іноді порушений, наприклад, при «позаштатному» зміні даного стану мережі зв'язку та її TCP-з'єднань, викликаному зміною структури самої мережі, стрибкоподібній зміні інтенсивності надходить в дану мережу різного трафіку або числа їх TCP-потоків.

Отже, з метою оптимізаційних параметрів протоколу TCP, які взаємодіють з даними механізмом боротьби з перевантаженням буде запропонований відповідний метод, який пов'язаний з поданням TCP-сеансів системою нелінійних диференціальних рівнянь. В основу даного методу буде покладено рішення, яке сформульоване на виході нелінійної диференціальної моделі TCP-сеансів даної

оптимізаційної задачі, де в якості критерію (4.14) виступає мінімум зважених щодо пріоритету даних відмов в обслуговуванні на транспортному рівні системи.

ВИСНОВКИ

У результаті виконання даної роботи було проаналізовано принципи функціонування мереж наступного покоління NGN та досліджено механізми управління потоком і перевантаженням у стеку протоколів TCP/IP. Показано, що протокол TCP, незважаючи на наявність певних обмежень, є ефективним і надійним засобом передачі даних у мережах NGN, особливо у порівнянні з протоколом UDP, який не забезпечує механізмів контролю доставки та управління потоком.

У ході дослідження встановлено, що управління потоком у TCP ґрунтується на поєднанні базових механізмів. Механізм ковзаючого вікна дозволяє узгоджувати швидкість передачі даних між відправником і одержувачем та запобігає переповненню приймальних буферів. Механізм повільного старту забезпечує поступове нарощування швидкості передачі відповідно до стану мережі та реалізує принцип самосинхронізації потоку даних. Алгоритм уникнення перевантаження, що базується на принципі AIMD, забезпечує стабільну роботу мережі шляхом лінійного збільшення розміру вікна передачі та мультиплікативного зменшення у разі виникнення втрат пакетів.

Проведений аналіз показав, що пропускна здатність TCP-з’єднання істотно залежить від значень часу кругового обходу (RTT), розміру вікна передачі та наявності «вузьких місць» у мережі. За умов великих затримок або підвищеного рівня втрат пакетів ефективність стандартних алгоритмів TCP знижується, що обумовлює доцільність використання удосконалених моделей управління потоком.

У роботі було досліджено сучасні алгоритми управління перевантаженням TCP, зокрема Reno, NewReno, Vegas, Tahoe, Westwood, Hybla, BIC та CUBIC. Встановлено, що алгоритми NewReno, BIC і CUBIC доцільно застосовувати у швидкісних мережах з великим значенням RTT (long fat networks). Алгоритм TCP

Hybla демонструє підвищену ефективність у супутникових каналах зв’язку, тоді як TCP Westwood завдяки оцінці доступної смуги пропускання краще адаптується до умов бездротових мереж із високою ймовірністю втрат пакетів. TCP Vegas забезпечує більш стабільну швидкість передачі даних, однак може поступатися іншим алгоритмам за максимальною пропускною здатністю.

Отримані результати підтверджують, що універсального алгоритму управління потоком TCP не існує, а ефективність його роботи визначається характеристиками мережі та умовами експлуатації. Вибір конкретного алгоритму управління перевантаженням повинен здійснюватися з урахуванням типу мережі, затримок, рівня втрат пакетів і вимог до якості обслуговування.

Результати даної роботи можуть бути використані при проектуванні та експлуатації телекомунікаційних мереж NGN, а також при виборі оптимальних алгоритмів управління потоком TCP з метою підвищення ефективності використання мережних ресурсів і якості надання інфокомунікаційних послуг.

ПЕРЕЛІК ПОСИЛАНЬ

1. Росляков А. В. *Сети следующего поколения NGN* / за ред. А. В. Рослякова. – М. : Эко-Трендз, 2008. – 400 с.

2. Правило В. В. *Дослідження методів управління телекомунікаційними мережами в умовах перевантаження* / В. В. Правило // І-а НТК Харківського університету повітряних сил. – 2005. – 16 с.

3. Варфоломеєва О. Г., Мороз О. О. Дослідження методів управління потоком в мережах NGN за протоколом TCP/IP // *Телекомунікаційні та інформаційні технології*. – 2016. – № 4.

4. Камер Д. *Сети TCP/IP. Том 1. Принципы, протоколы и структура*. – М. : Вильямс, 2003. – 417 с.

5. Кришнамурти Б., Рексфорд Дж. *Web-протоколы. Теория и практика*. – М. : БИНОМ, 2002. – 592 с.

6. Колченко Г. Ф., Варфоломеева О. Г. Построение модели оптимального проектирования системы управления телекоммуникационными сетями // *Праці 2-ї міжнародної конференції «Проблеми управління мережами та послугами телекомунікацій в умовах конкурентного ринку»*. – 2003. – № 2. – С. 15–18.

7. RFC 793. *Transmission Control Protocol (TCP)* [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc793.html (дата звернення: 10.10.2025).

8. Семенов Ю. В. *Проектирование сетей связи следующего поколения*. – СПб. : Наука и Техника, 2005. – 240 с.

9. RFC 5681. *TCP Congestion Control* [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc5681.html (дата звернення: 10.10.2025).

10. Microsoft Learn. *Network Adapter Performance Tuning in Windows Server* [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics (дата звернення: 10.10.2025).

11. RFC 5348. *TCP Friendly Rate Control (TFRC): Protocol Specification* [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc5348.html (дата звернення: 10.10.2025).

12. Старкова Е. В., Меркулов М. В. Анализ устойчивости TCP-сеанса в рамках нелинейных моделей // *Перспективные технологии в средствах передачи информации* : материалы 8-й международной научно-технической конференции. – Владимир, 2009. – С. 145–149.

13. Сысоев П. А. Математическая модель адаптивного управления скоростью передачи в стеке протоколов TCP/IP // *Технологическая системотехника*. – 2006. – Вып. 11. – С. 153–164.

14. RFC 2581. *TCP Congestion Control* [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc2581.html (дата звернення: 10.10.2025).

15. Fall K., Floyd S. *Simulation-based Comparisons of Tahoe, Reno, and SACK TCP* [Електронний ресурс]. – Режим доступу: https://www.icir.org/floyd/papers/sacks.pdf (дата звернення: 10.10.2025).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НА ТЕМУ:

«МЕТОДИ УПРАВЛІННЯ ПОТОКОМ В МЕРЕЖАХ НАСТУПНОГО ПОКОЛІННЯ»

Розробив: Терещенко
Вадим Олегович

Керівник:
Варфоломеєва
Оксана Григорівна

КИЇВ-2026

МЕТА РОБОТИ

Мета: дослідити, як у мережах NGN працюють механізми **керування потоком** та **запобігання перевантаженню** у TCP/IP і як це впливає на пропускну здатність та затримки.

Завдання:

- розглянути архітектуру NGN та місце TCP/IP;
- описати базові механізми TCP (вікно, slow start, AIMD);
- проаналізувати «вузькі місця»;
- виконати розрахунки RTT/RTO та затримки;
- порівняти алгоритми TCP і запропонувати оптимізаційний підхід.

АРХІТЕКТУРА NGN

- **Ключова ідея NGN:** мультисервісна IP-інфраструктура + розділення функцій (транспорт / керування / послуги).
  Типовий 4-рівневий підхід:
- рівень доступу (підключення абонентів)
- транспортний рівень (перенесення пакетів)
- рівень керування (сесії/сигналізація)
- рівень послуг (логіка сервісів)



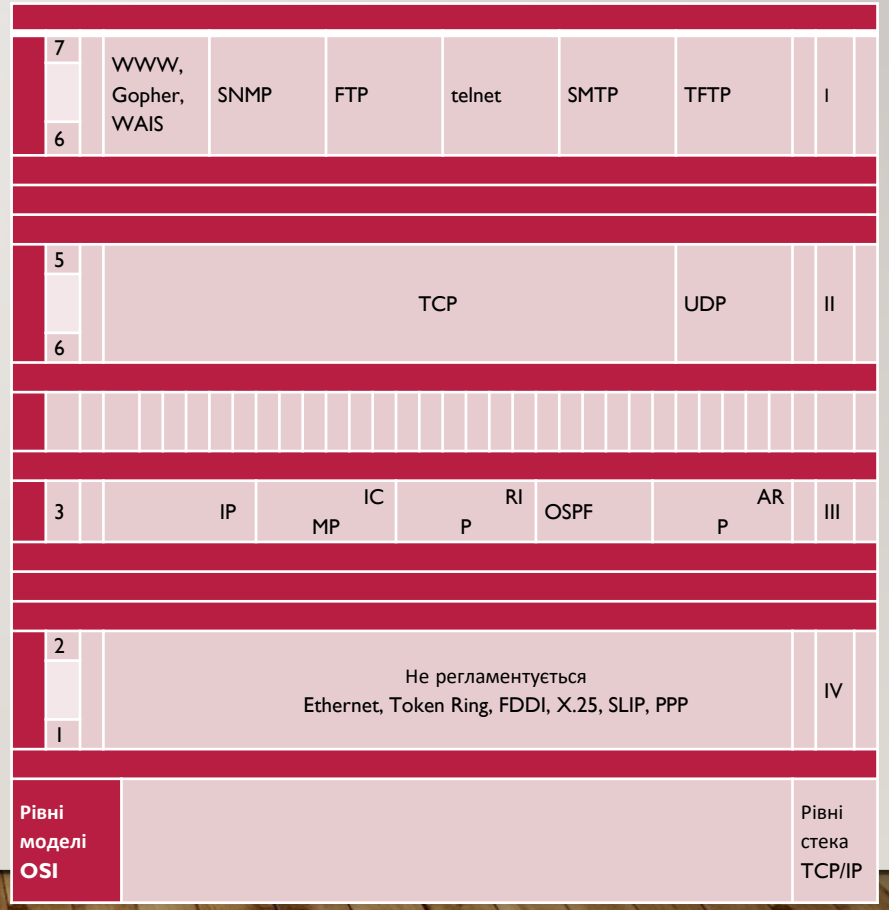
Архітектура мережі наступного покоління

ВИЗНАЧЕННЯ СТЕКА TCP/IP

TCP/IP — набір протоколів для пакетної передачі даних, що є основою Інтернету та NGN.

Переваги багаторівневості:

- сумісність різних технологій доступу;
- масштабованість;
- можливість незалежно розвивати рівні.

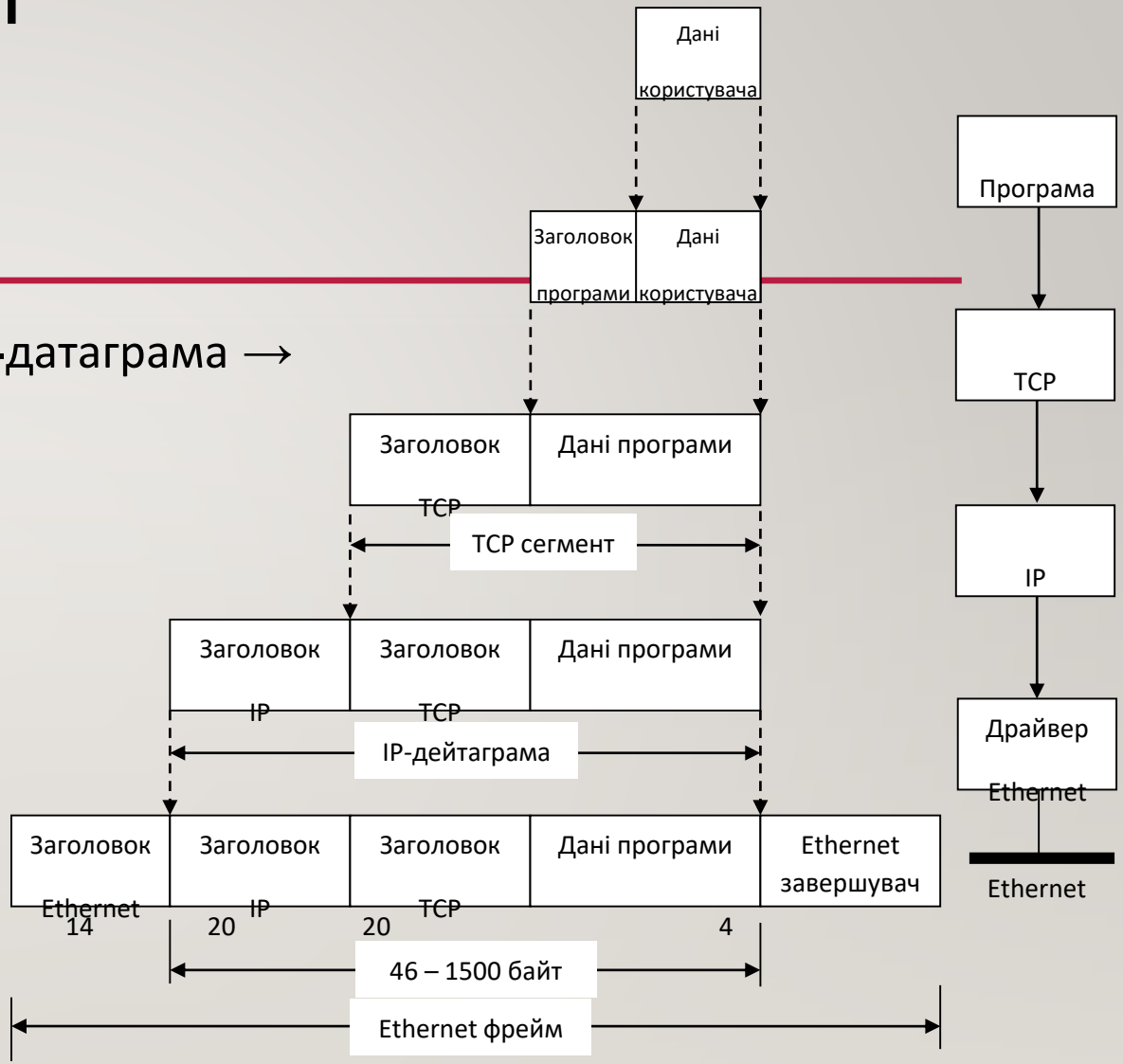


ФОРМАТ ПРОТОКОЛУ TCP/IP

Інкапсуляція: дані застосунку → TCP-сегмент → IP-датаграма → кадр канального рівня.

TCP-заголовок забезпечує:

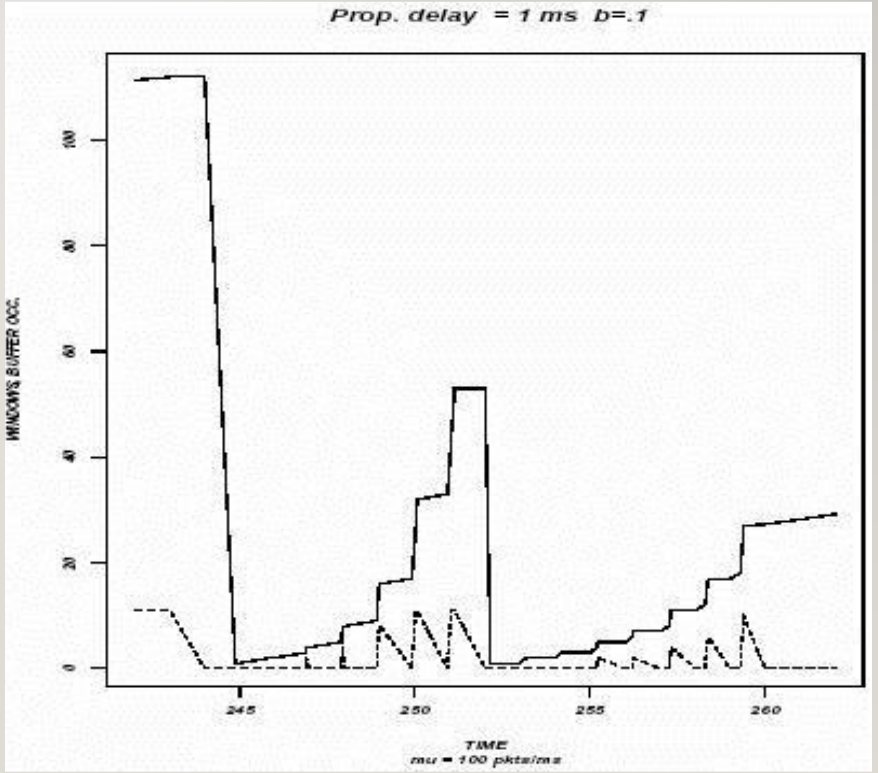
- порти (ідентифікація сервісу),
- sequence/ack (порядок і підтвердження),
- window (керування потоком),
- checksum (контроль цілісності).



МЕХАНІЗМИ УПРАВЛІННЯ ПОТОКОМ (TCP)

Керування потоком у TCP базується на поєднанні механізмів:

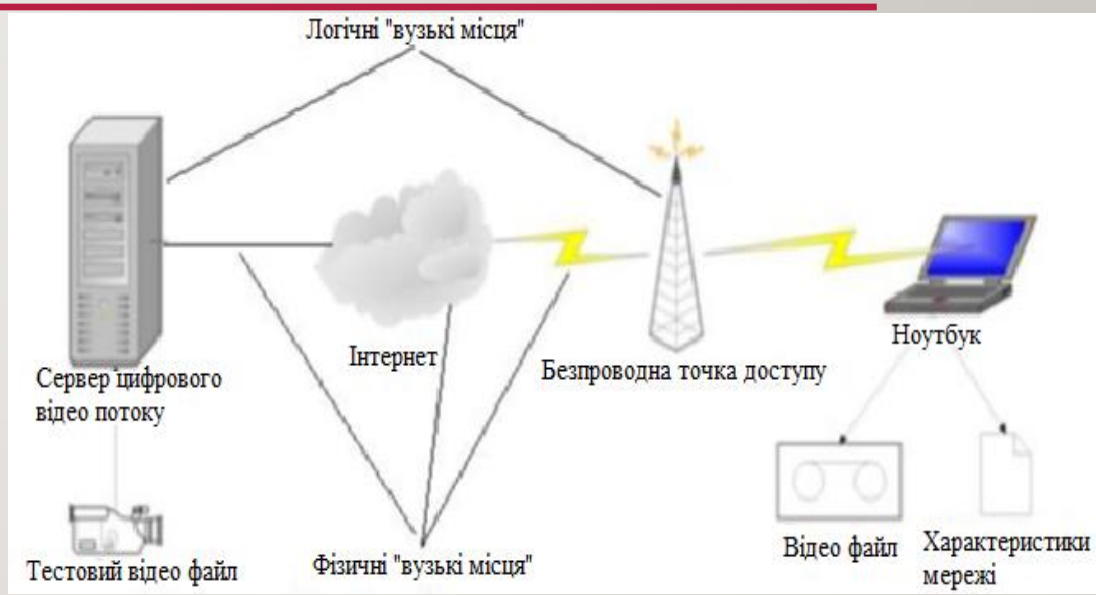
- **ковзаюче вікно** — узгодження швидкості відправника і приймача;
- **повільний старт (Slow Start)** — поступове нарощування швидкості;
- **уникнення перевантаження (AIMD)** — лінійне збільшення вікна та мультиплікативне зменшення при втратах.



«ВУЗЬКІ МІСЦЯ» В МЕРЕЖІ

Що таке «вузьке місце»: елемент/ділянка, де пропускна спроможність значно нижча за інші частини мережі.

Класифікація:
- **фізичні** (повільніший канал між вузлами),
- **логічні** (черги/перевантаження, обмеження приймача).

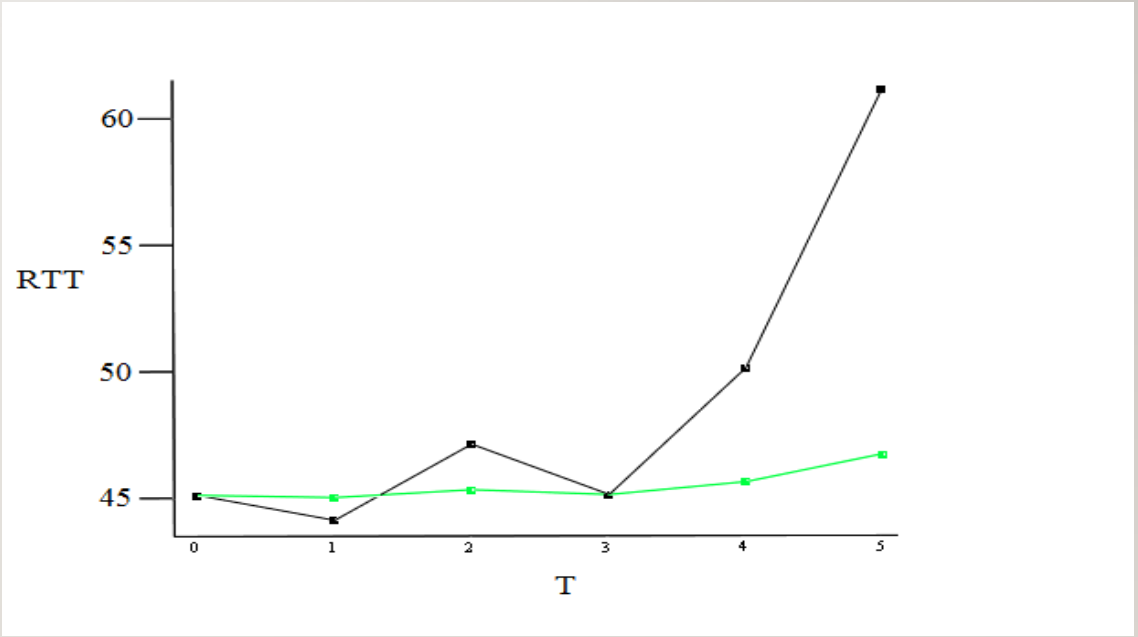


Приклади логічних і фізичних «вузьких місць»

РОЗРАХУНОК ЗАТРИМКИ: RTT ТА ТАЙМ-АУТИ TCP (RTO)

RTT (Round-Trip Time) — час проходження пакета до адресата і назад; TCP вимірює RTT та усереднює значення.

RTO (Retransmission Timeout) — тайм-аут повторної передачі; у роботі використано підхід:
RTO = RTTm · b, де **b ≈ 2**.

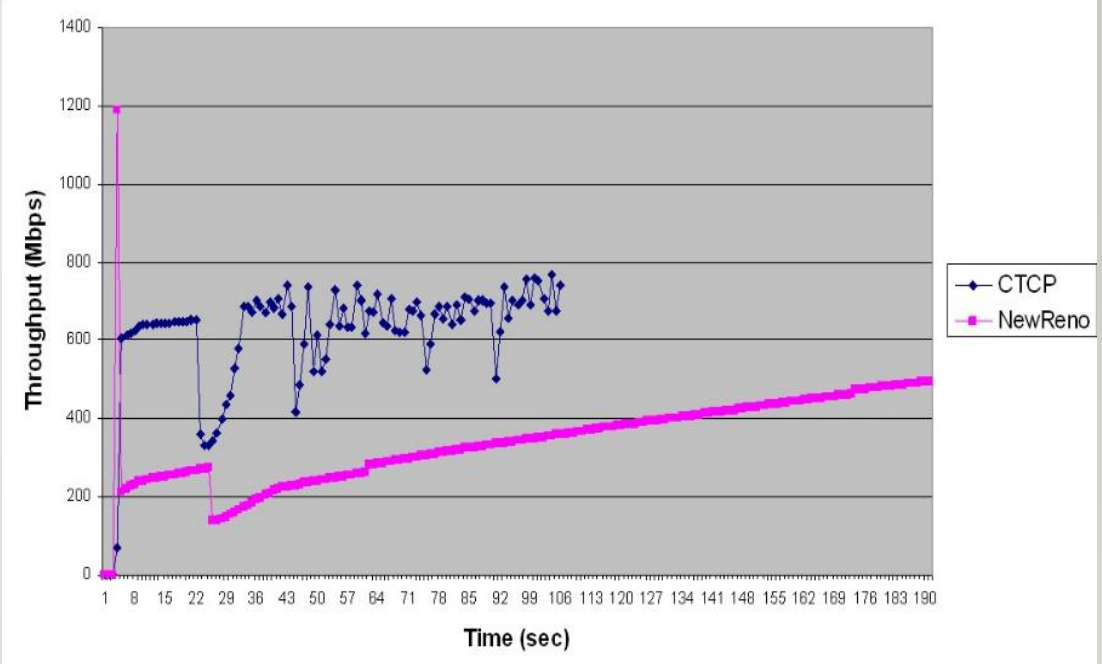


Графік коригування середнього значення кругового обходу

МЕТОД ОПТИМІЗАЦІЇ TCP-З'ЄДНАНЬ

Суть підходу: оптимізувати вибір параметрів TCP та алгоритмів керування чергами **RED/WRED**, шукаючи компроміс між показниками якості.

Критерій: використовується функціонал (4.13) – (4.14), де враховується пріоритет TCP-потоків і інтервал оптимізації **T**.



Порівняння роботи NewReno та Compound

ВИСНОВКИ

•Проаналізовано принципи NGN та досліджено механізми управління потоком/перевантаженням у TCP/IP.

•Показано роль ковзаючого вікна, slow start та AIMD у стабілізації передачі.

•Підтверджено, що TCP є ефективним засобом надійної передачі у NGN (порівняно з UDP, що не має механізмів контролю доставки/потоку).

•Запропоновано підхід оптимізації параметрів TCP і RED/WRED за критерієм якості.

ДЯКУЮ ЗА УВАГУ!