ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ТЕЛЕКОМУНІКАЦІЙ

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему: **“МЕТОДИ КОНТРОЛЮ ЯКОСТІ ФУНКЦІОНУВАННЯ МЕРЕЖЕВОГО ТЕЛЕКОМУНІКАЦІЙНОГО ОБЛАДНАННЯ”**

Виконав: студент 6 курсу, групи ТСДМ-63
спеціальності
172 Телекомунікації та радіотехніка
(шифр і назва спеціальності)
Рутківський Д.С.
(прізвище та ініціали)

Керівник Твердохліб М.Г.
(прізвище та ініціали)

Рецензент
(прізвище та ініціали)

Нормоконтроль
(прізвище та ініціали)

Київ - 2019

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ТЕЛЕКОМУНІКАЦІЙ

Кафедра Телекомунікаційних систем та мереж

Ступінь вищої освіти Магістр

Спеціальність 172 Телекомунікації та радіотехніка
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
телекомунікаційних систем та мереж
_____ В.Ф. Заїка
“ ” _____ 2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Рутківському Дмитру Сергійовичу

1. Тема роботи: “Методи контролю якості функціонування мережевого телекомунікаційного обладнання”,
керівник роботи Твердохліб Микола Григорович, к.т.н., доцент,
затверджені наказом вищого навчального закладу від 14.11.2019 №518.

2. Строк подання студентом роботи 20.12.2019 р.
3. Вихідні дані до роботи:
   1. Алгоритми боротьби з накопиченням черг та їх усунення.
   2. Ефективність роботи сучасних алгоритмів по забезпеченню якості обслуговування в умовах перевантажень мережі.
   3. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   1. Методи організації і обробки черг в умовах перевантаження;
   2. Моделі диференційованих та інтегрованих послуг;
   3. Забезпечення QoS за допомогою алгоритмів активного управління чергами в середовищі NS-2.

5. Перелік графічного матеріалу (назва слайдів презентації):
   1. Мета роботи;
   2. Класифікація типів мережевого трафіку;
   3. Механізми забезпечення якості обслуговування;
   4. Засоби Quality of Service;
   5. Методи боротьби з накопиченням черг: Спрощені методи;
   6. Методи боротьби з накопиченням черг: Удосконалені алгоритми;
   7. Дослідження роботи алгоритму WRED за допомогою програми NS-2;
   8. Результати моделювання;
   9. Висновки.

6. Дата видачі завдання 11.09.2019

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Підбір науково-технічної літератури	27.09.19	Викон.
2.	Методи організації і обробки черг в умовах перевантаження;	15.10.19	Викон.
3.	Моделі диференційованих та інтегрованих послуг	08.11.19	Викон.
4.	Забезпечення QoS за допомогою алгоритмів активного управління чергами в середовищі NS-2	29.11.19	Викон.
5.	Висновки, вступ, реферат	10.12.19	Викон.
6.	Розробка презентації	18.12.19	Викон.

Студент _____ Рутківський Д.С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Твердохліб М.Г.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської кваліфікаційної роботи: 80 сторінок, 21 рисунок, 7 таблиць, 33 джерела.

Об'єкт дослідження – процес управління навантаженням мережевого телекомунікаційного обладнання.

Предмет дослідження – принципи і методи управління навантаженням на телекомунікаційну мережу для забезпечення якості обслуговування в мережі Інтернет.

Мета роботи – дослідити принципи і методи управління навантаженням на телекомунікаційну мережу.

Методи дослідження – теорії ймовірності та математичної статистики, аналітичного та імітаційного моделювання, а також теорії телекомунікаційних систем та мереж.

Досліджено методи забезпечення якості обслуговування мережевих додатків в загальнодоступних мережах Інтернет. У роботі вивчається процес використання алгоритму WRED, оскільки цей механізм реалізований практично в усіх сучасних маршрутизаторах. За допомогою програми NS-2 проведені дослідження по ефективності роботи алгоритму WRED.

QOS, ЗАБЕЗПЕЧЕННЯ ЯКОСТІ, ОБРОБКИ ЧЕРГ, ПЕРЕВАНТАЖЕННЯ, WRED, АЛГОРИТМИ, МОДЕЛЮВАННЯ, ОСЦИЛЯЦІЇ, ТРАФІК, DSCP, DIFFSERV, SERVICE LEVEL AGREEMENTS, TRANSMISSION CONTROL PROTOCOL, WEIGHTED RANDOM EARLY DETECTION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AF	assured forwarding - гарантована передача;
AQM	Active queue management - активне управління чергами;
CIR	Commited information rate - гарантована швидкість інформаційного потоку;
CBWFQ	Class Based Weighted Fair Queueing - зважений алгоритм рівномірного обслуговування черг на основі класів;
DiffServ	Диференційовані послуги;
DSCP	Полі мітки (DiffServ Code Point);
DRTT	Дисперсія RTT;
IntServ	Integrated Services - архітектура інтегрованих послуг;
QoS	Quality of Service - якість обслуговування;
RED	Random early detection - алгоритм довільного раннього виявлення;
RSVP	“Resource ReSerVation Protocol", RFC - 2205;
RTT	Round Trip Time - сума часів доставки сегменту від відправника до одержувача і відгуку від одержувача до відправника;
SLA	(Service Level Agreements) угода;
SVC	Switched Virtual Circuit - перемикана віртуальна схема;
SVP	Switched Virtual Path - перемиканий віртуальний шлях;
TCP	Transmission control protocol - протокол управління передачею;
VCI	Virtual Path Identifier - ідентифікатор віртуального шляху;
VP	Virtual Path - віртуальний шлях;
VPI	Virtual Path Identifier - ідентифікатор віртуального шляху;
WFQ	Weighted Fair Queuing - схема обробки черг «справедлива зважена черга»;
WRED	Weighted random early detection - зважений алгоритм довільного раннього виявлення;

ВСТУП

Магістерська робота присвячена дослідженню методів забезпечення якості обслуговування мережевих додатків в загальнодоступних мережах Інтернет. Ефективність використання смуги пропускання каналу завжди була актуальним завданням, але її важливість зросла останніми роками у зв'язку з появою усе більш жорстких вимог до якості обслуговування (QoS) [1].

Для забезпечення необхідних вимог до різних потоків даних використовуються два методи QoS: управління перевантаженням (congestion management) і запобігання перевантаженням (congestion avoidance). Перший метод заснований на привласненні квот і пріоритетів потокам. У разі перевантаження, потоки отримують якість, обмежену їх квотою і пріоритетом (наприклад, WFQ). Другий метод обмежує розмір черги, сигналізуючи джерелам даних про необхідність зменшити швидкість передачі інформації (наприклад, RED). Найбільш відомі модифікації алгоритму RED: WRED, GRED (Gentle RED), DRED (Dynamic RED), SRED (Stabilized RED), ARED (Adaptive RED), RED - PD.

Окремо слід згадати учених, які першими почали вирішувати проблеми якості обслуговування і запобігання і боротьби з перевантаженнями: Салли Флойд (Sally Floyd), Ван Якобсон (V. Jacobson), Кевин Фолл (Kevin Fall), Ратул Маханджан (Ratal Mahajan), Мартін Мей (Martin May), Жан Болот (Jean Bolot), Вишал Мисра (Vishal Misra), Вейбо Гонг (WeiBo Gong), Дон Тоуслей (Don Towsley), Томас Зейглер (Thomas Ziegler), Давид Везерол (David Wetherall), Добрушин Р.Л., Кузнєцов Н.А., Вишневський В.М., Ляхів А.І., Богуславский Л.Б., Дрожжинов В. І., Башарин Г.Л., Бондарів П.Л.

У роботі вивчається процес використання алгоритму WRED, оскільки цей механізм реалізований практично в усіх сучасних маршрутизаторах, а інші його модифікації лише бурхливо обговорюються і не мають практичної реалізації в мережевих пристроях. Незважаючи на значний об'єм публікацій по темі запобігання перевантаженням, залишається проблема вибору налаштувань параметрів для ал-

горитму WRED. Багато дослідників WRED згодні з тим, що вплив алгоритму на якість передачі потоків сильно залежить від правильного завдання його параметрів, але і досі немає зрозумілої інструкції, як на практиці вибирати значення цих параметрів.

До параметрів якості обслуговування слід віднести доступну смугу пропускання, вірогідність втрати пакету, розкид часу доставки і час доставки пакету від відправника до одержувача. Усі ці параметри залежать від алгоритмів формування і обслуговування черг пакетів в мережевих пристроях (перемикачах і маршрутизаторах). У сучасних мережевих пристроях застосовуються алгоритми RED/WRED, PQ (priority queueing), WFQ (weighted fair queueing), LLQ (low latency queueing), CBWFQ (class based weighted fair queueing) і т.д.

У цій роботі виконано практичне дослідження моделі DiffServ (Differentiated Services - механізм, який залежно від вимог до якості обслуговування записує в IP заголовки пакетів спеціальні мітки DSCP, - DiffServ Code Point, значення яких враховуються мережевим устаткуванням при передачі пакетів через мережу) у тому вигляді, як вона реалізована в мережевому устаткуванні на сьогодні. В результаті досліджень сучасних засобів забезпечення якості обслуговування (QoS, DiffServ, MPLS - TE - Multiprotocol Label Switching Traffic Engineering, RSVP - TE) були сформульовані практичні рекомендації для сервіс-провайдерів, що бажають надати певні параметри якості обслуговування своїм клієнтам (смугу пропускання, гарантовану затримку і її дисперсію, мінімальну вірогідність втрати пакетів).

У рамках вимірів і моделювання віртуального каналу (із залученням програмного пакету NS - 2) показано, що при деяких конфігураційних параметрах можливі осциляції довжини черги (особливо для потоків середнього і низького пріоритету). Цей ефект пов'язаний з механізмом обчислення усередненого значення довжини черги в алгоритмах WRED і RIO (RED with Input Output). Такі осциляції можуть привести до зростання дисперсії RTT (Round Trip Time - сума часів доставки сегменту від відправника до одержувача і відгуку від одержувача до відправника), що украй небажано для роботи з мультимедійними даними і при

рішенні завдань управління в реальному масштабі часу. Осциляції усередненої довжини черги знижують також ефективність використання каналу.

Мета роботи – дослідити методи контролю якості функціонування мережевого телекомунікаційного обладнання.

Об'єкт дослідження – процес управління навантаженням мережевого телекомунікаційного обладнання.

Предмет дослідження – принципи і методи управління навантаженням на телекомунікаційну мережу для забезпечення якості обслуговування в мережі Інтернет.

Метод дослідження. Дослідження виконано на основі використання основних положень теорії ймовірності та математичної статистики, аналітичного та імітаційного моделювання, а також теорії телекомунікаційних систем та мереж.

Апробація результатів магістерської роботи. Основні результати магістерської атестаційної роботи доповідалися на конференціях Державного університету телекомунікацій та опубліковано в науковому журналі "Наукові записки Українського науково-дослідного інституту зв'язку".

1. МЕТОДИ ОРГАНІЗАЦІЇ І ОБРОБКИ ЧЕРГ В УМОВАХ ПЕРЕВАНТАЖЕННЯ

1.1. Забезпечення якості обслуговування у високошвидкісних мережах

Сучасна тенденція конвергенції мереж різних типів, а також збільшення об'єму трафіку і поява додатків, що працюють в режимі реального часу, мультимедійних додатків привели до необхідності перенесення мережею різних видів трафіку, у тому числі, чутливого до затримок. Тому традиційні TCP/IP мережі не гарантують необхідне додаткам якість обслуговування і виникає необхідність в розробці додаткових засобів надання додаткам необхідного рівня сервісу [2].

Якість обслуговування.

Під якістю обслуговування (Quality of Services, QoS) розуміють інтегральний корисний ефект від обслуговування, який визначається мірою задоволення користувача як від отриманої послуги, так і від самої системи обслуговування. Критерій якості обслуговування представляється у вигляді інтегрального показника досконаліть обслуговування, що враховує не лише якість послуги, але і здатність мережі обробляти навантаження.

Послуга може надаватися з різними рівнями якості. Прийнятний рівень якості послуги узгоджується між сервіс-провайдером і його клієнтом і включається в текст відповідної сервісної угоди (Service Level Agreement - SLA). Рівень якості послуги задається конкретним набором значень визначальних параметрів якості послуги. Виділяють наступні рівні: бажаний, гарантований, виміряний. Трафарети сервісної угоди відносно рівня послуг, що надаються, темплети SLA (Service Level Agreement Templates) є визначеннями стандартних рівнів послуги, які можуть бути запропоновані покупцям послуги у рамках SLA. Наприклад, можуть бути запропоновані трафарети, які визначають характеристики так званої "золотої послуги" або "срібної послуги" і т. п.

Вимоги до якості обслуговування додатків різних типів.

Впровадження механізмів QoS припускає забезпечення з боку мережі з'єднання з певними обмеженнями по продуктивності, основними характеристиками якої являються смуга пропускання, затримка, джиттер і рівень втрати пакетів. Характеристики QoS особливо важливі у тому випадку, коли мережа передає одночасно трафік різного типу, наприклад, трафік веб - додатків і голосовий, оскільки різні типи трафіку пред'являють різні вимоги до характеристик QoS. Врахувати одночасно усі характеристики QoS для усіх видів трафіку складно, тому види трафіку, існуючі в мережі, класифікують, відносячи кожного до одного з поширених типів, а потім намагаються досягти одночасного виконання певної підмножини з набору вимог для цих типів трафіку.

Як основні критерії класифікації прийнято три характеристики трафіку: відносна передбачуваність швидкості передачі даних, чутливість трафіку до затримок пакетів і чутливість трафіку до втрат і спотворень пакетів (рис. 1.1).

1.2. Обробка черг

Черги FIFO (First - In - First - Out) використовуються зазвичай в швидкісних інтерфейсах (швидкість > 2048 кбіт/с). Тут перший пакет, що прийшов, першим покидає чергу. Порядок дотримання пакетів при цьому алгоритмі не змінюється. Пріоритетне обслуговування в цьому варіанті також не може бути здійснене. Коли черга заповнена, усі наступні пакети відкидатимуться до тих пір, поки з черги не буде вилучений хоч би один пакет.

Пріоритетне обслуговування черг (PQ). Пріоритетне обслуговування (PQ) являється ефективною і прямою формою керування перевантаженням. PQ дозволяє мережевому адміністраторові виділити до чотирьох черг в мережевому трафіку. Передбачені черги високого, середнього, нормального і низького пріоритету. Маршрутизатор обробляє черги строго відповідно до їх пріоритету. Пакети з черги з високим пріоритетом обробляються, поки в черзі не залишиться жодного пакету, після цього починається обробка черги з середнім пріоритетом, паралельно здійснюється контроль появи пакетів в черзі з високим пріоритетом. Пакети з чер-

ги з низьким пріоритетом обробляються лише тоді, коли інші черги порожні. Низько пріоритетний трафік при певних обставинах може бути повністю блокований, а пакети втрачені. Зазвичай PQ використовується, коли додатки, критичні до затримок, стикаються з проблемами. Якщо високопріоритетний трафік має високу інтенсивність, висока вірогідність того, що інші складові трафіку будуть блоковані. Пакети, не класифіковані PQ, автоматично відносяться до черги з нормальним пріоритетом. Системна черга має пріоритет вище за високий. За замовчанням черги різних пріоритетів мають наступні розміри (CISCO) (табл. 1.1).

Звичайне обслуговування черг (CQ). Щоб уникнути жорсткої регламентації системи обслуговування черг PQ, системний адміністратор може вибрати стратегію CQ (Custom Queuing - звичайне обслуговування черг). CQ дозволяє мережевому адміністраторові реалізувати пріоритетне обслуговування трафіку без побічних ефектів, пов'язаних з блокуванням низькопріоритетних потоків. За допомогою CQ можна сформувати 16 черг трафіку. Кожна з черг обслуговується в карусельному режимі (round - robin). За замовчанням CQ обслуговує 1500 байт за цикл. Проте CQ не може фрагментувати пакети. Це означає, що якщо CQ обробляє пакет завдовжки 1000 байт, наступний 1500-байтний пакет буде оброблений поза чергою. При цьому може виникнути ситуація, коли для коротких черг виникає тайм-аут. CQ може конфігуруватися так само, як і PQ. Процес класифікації трафіку приписує пакети однієї з 16 черг, що конфігуруються, працюють в режимі "дірявого відра". Існує черга з номером 0 (системна черга), яка зарезервована для пакетів управління мережею і має найвищий пріоритет. При великих потоках можлива втрата пакетів через переповнювання черг. Довжина черги може бути задана в межах від 0 до 32767 (20 - значення за умовчанням).

Справедливі черги (WFQ). Стратегія справедливих (зважених) черг WFQ (Weighted Fair Queuing) використовується за замовчуванням для інтерфейсів низької швидкості (іншою назвою цього алгоритму являється SFQ - Stochastic Fairness Queuing). WFQ ділить трафік на декілька потоків, використовуючи як параметри (для IP-протокола): IP-адресу і порти одержувача і відправника, а також поле IP -заголовка ToS. Значення ToS служить для кваліфікації (частини смуги,

що виділяється) потоку. Для кожного з потоків формується своя черга. Максимально можливе число черг дорівнює 256. Черги обслуговуються відповідно до карусельного принципу (round - robin). Вищий пріоритет мають потоки з меншою смугою, наприклад telnet. За умовчанням кожна з черг має місткість 64 пакети (але допускається значення і <4096 пакетів).

У мережах існує 8 рівнів пріоритету. Слід мати на увазі, що WFQ не підтримується у разі тунелювання або шифрування. Субпотік з низькою вагою отримує вищий рівень обслуговування, чим з високим. Не реалізується WFQ і в мережах ATM. Коли задіяні біти ToS, WFQ реалізує пріоритетне обслуговування пакетів згідно зі значенням цього коду. Ваговий чинник зворотно пропорційний рівню пріоритету [3].

Справедливі черги, що базуються на класах (CBWFQ). Подальшим розвитком технології WFQ являється формування класів потоків, що задаються користувачем. Алгоритм побудови черг, що базуються на класах, називається CBWFQ (Class - Based Weighted Fair Queuing). Алгоритм CBWFQ надає механізм управління перевантаженням. Параметри, які характеризують клас, ті ж, що і у разі WFQ (тільки замість ToS використовується пріоритет). На відміну від WFQ тут можна в широких межах перерозподіляти смугу пропускання між потоками. Для виділення класу можуть притягуватися ACL (Access Control List) або навіть номер вхідного інтерфейсу. Кожному класу ставиться у відповідність черга. На відміну від RSVP цей алгоритм гарантує смугу лише в умовах перевантаження. Всього може бути визначений 64 класи. Нерозподілена смуга може використовуватися потоками згідно з їх пріоритетами. Якщо використовується резервування CBWFQ і RSVP, то можливі конфлікти, оскільки IOS маршрутизатора не перевіряє балансу зарезервованої смуги різними протоколами.

Черги з малою затримкою (LLQ). В деяких випадках, наприклад у разі VoIP, більш важливо забезпечити малу затримку, а не смугу пропускання. Для таких завдань розроблений алгоритм LLQ (Low Latency Queuing), який являється модифікацією CBWFQ. У цьому алгоритмі пакети усіх пріоритетів, окрім найвищого, вимушені чекати, поки черга вищого пріоритету буде спустошена (див. PQ). Роз-

кид затримки у високопріоритетному потоці може бути пов'язаний тільки з очікуванням завершення передачі пакету низького пріоритету, що почалася до приходу пріоритетного кадру. Такий розкид визначається діапазоном довжин кадрів (MTU).

Методи роботи в умовах перевантаження. Коли в субмережу (чи якийсь мережевий пристрій) поступає надто багато пакетів, її (його) робочі характеристики деградують. Така ситуація називається перевантаженням. Оптимальність управління мережею в умовах перевантажень визначає ефективність використання мережі. Поки субмережа завантажена трохи, число пакетів, що приймаються і оброблюваних, дорівнює числу тих, що прийшли. При дуже великих завантаженнях пропускна спроможність каналу або мережі може стати нульовою. Варіанти характеристик навантажень мереж показані на рис. 1.1.

Ідеальна (з точки зору споживача) характеристика навантаження мережі спочатку описується прямою лінією, тобто числа посланих і доставлених пакетів рівні або майже рівні. Досягши максимуму пропускної спроможності мережі число пакетів, що доставляються в одиницю часу, стає постійним, тоді як число посланих пакетів продовжує рости. На жаль, такого в реальному житті не відбувається. Бажаною характеристикою навантаження являється така, яка при малих і максимальних завантаженнях поводиться як ідеальна, а при перехідних швидкостях має плавний характер і доставляє декілька менший відсоток пакетів, чим в ідеалі. Реальна характеристика (наприклад, для Ethernet) навантаження багато гірше навіть бажаного варіанту. Спочатку у міру зростання завантаження починає рости відсоток втрати пакетів. Далі при вхідних потоках, що перевищують деякий поріг, число пакетів, що доставляються, починає падати при зростанні числа посланих пакетів, а при подальшому наростанні завантаження число пакетів, що доставляються, може стати рівним нулю ("колапс"). У версіях Ethernet, де зіткнення виключені (у мережі використовуються тільки перемикачі і маршрутизатори), ситуація сприятливіша.

Частково це може бути пов'язано з недоліком пам'яті для вхідних буферів, з цієї причини деяке збільшення пам'яті може допомогти. Але слід пам'ятати, що

всякі ліки добре в міру. Ще в 1987 році Нагле (Nagle) виявив, що якщо маршрутизатор має навіть безмежну пам'ять, ефект перевантаження може виявитися ще важчим. Це зв'язано з часом, протягом якого пакети чекають обробки. Якщо час очікування в черзі перевищує тривалість тайм-ауту, з'являться дублікати пакетів, що знижує ефективність системи. Причиною перевантаження може бути повільний процесор або недостатня пропускна спроможність якоїсь ділянки мережі. Проста заміна процесора або інтерфейсу на більш швидкодіючий не завжди вирішує проблему - частіше переносить "вузьке місце" в іншу частину системи. Перевантаження, як правило, включає механізми, що посилюють її негативну дію. Так, переповнювання буфера призводить до втрати пакетів, які пізніше мають будуть бути передані повторно (можливо, навіть кілька разів). Процесор передавальної сторони отримує додаткове паразитне завантаження. Усе це вказує на те, що контроль перевантаження являється вкрай важливим процесом.

Перевантаження, по суті, пов'язане з часовою або стаціонарною неузгодженістю характеристик якихось частин системи.

Слід робити відмінність між контролем потоком і контролем перевантаження. Під контролем потоку мається на увазі балансування потоку відправника і можливостей прийому і обробки одержувача. Цей вид контролю припускає наявність зворотного зв'язку між одержувачем і відправником. У такому процесі беруть участь, як правило, тільки два партнери. Перевантаження ж - загальніше явище, що відноситься до мережі в цілому або до якоїсь її частини. Наприклад, 10 ЕОМ хочуть передати одночасно якісь файли іншим 10 ЕОМ. Конфлікту потоків тут немає, кожна з ЕОМ здатна переробити дані, що поступають, але мережа не може пропустити потік, генерований 10 мережевими інтерфейсами одночасно. Часто ці явища не так просто розділити. Відправник може отримати повідомлення ICMP (quench - у разі стека протоколів TCP/IP) при перевантаженні одержувача або при перевантаженні якогось мережевого пристрою [4].

Починати потрібно з вирішення проблеми виявлення перевантажень. Перевантаженням слід вважати ситуацію, коли навантаження протягом деякого обумо-

вленого часу перевищує задану величину. Параметрами, які дозволяють судити про наявність перевантаження, можуть служити:

- відсоток пакетів, що відкидаються через відсутність вільного буферного простору;
- середня довжина черги (яка впливає на час доставки);
- відсоток пакетів, що пересилаються повторно;
- середній час затримки пакету і деякі інші величини.

Коли перевантаження виявлене, треба передати необхідну інформацію з точки, де вона виявлена, туди, де можна щось зробити для виправлення ситуації.

Можна послати повідомлення про перевантаження відправникові, завантажуючи додатково і без того переобтяжену ділянку мережі. Альтернативою цьому може бути застосування спеціального поля в пакеті, куди маршрутизатор може записати відповідний код при перевантаженні і послати його сусідам. Можна також ввести спеціальний процесор або маршрутизатор, який розсилає періодично запити про стан елементів мережі. При отриманні сповіщення про перевантаження інформаційний потік може бути посланий в обхід.

При використанні зворотного зв'язку шляхом посилки повідомлення запиту пониження швидкості передачі слід ретельно настроювати часові характеристики. Інакше система або потрапляє в незгасаючий коливальний режим, або пониження потоку, що коригує, здійснюватиметься надто пізно. Для коректного вибору режиму зворотного зв'язку потрібне деяке усереднювання. Але застосування занадто великих часів усереднювання може породити осциляцію довжини черги.

Подолання перевантаження може бути здійснене пониженням навантаження або додаванням ресурсів приймачу.

Позитивний результат може бути досягнутий зміною механізму підтвердження (наприклад, зменшенням розміру вікна), варіацією значень таймаутів, зміною політики повторної передачі пакетів. В деяких випадках позитивного результату можна досягти модифікацією схеми буферизації. Іноді розв'язати проблему може маршрутизатор, наприклад, перерозподіляючий трафік по декількох шляхах.

Однією з причин перевантаження часто являються імпульсні завантаження сегменту мережі або мережевого пристрою. З цієї причини будь-які заходи (наприклад, pipelining - конвеєр), які можуть вирівнювати потік пакетів, безумовно, поліпшать ситуацію (наприклад, traffic shaping в мережах ATM). У TCP же з його вікнами імпульсні завантаження зумовлені.

Алгоритм leaky bucket ("діряве відро"). Для систем без зворотного зв'язку вирішення проблеми вирівнювання швидкості передачі даних може бути вирішене за допомогою алгоритму leaky bucket ("діряве відро"). Суть цього алгоритму полягає в тому, що на шляху потоку встановлюється буфер, вихідний потік якого постійний і узгоджений з можливістю приймача. Якщо буфер переповнюється, пакети втрачаються.

Втрата пакетів - річ малоприємна, але це блокує процеси, які можуть привести до колапсу сегменту або усієї мережі.

Там, де втрата пакетів небажана, можна застосувати гнучкіший алгоритм.

Алгоритм "маркерне відро". Алгоритм token bucket (маркерне відро) припускає наявність в буферному пристрої (чи програмі) деякої кількості маркерів. При вступі на вхід буфера пакетів маркери використовуються для їх транспортування на вихід. Подальша передача даних на вихід залежить від генерації нових маркерів. Пакети, що надходять ззовні, тим часом накопичуються в буфері. Таким чином, повної гарантії відсутності втрат ми не маємо і тут. Але алгоритм token bucket дозволяє передавати на вихід "щільні" групи пакетів обмеженої чисельності (по числу маркерів), знижуючи в деяких випадках вірогідність втрати. Якщо буферний пристрій "змонтований" усередині ЕОМ відправника, втрат можна уникнути зовсім, блокуючи передачу при заповненні буфера. Як у одному, так і в іншому алгоритмі мірою передаваної інформації може бути не пакет, а n-байт (де n - деяке обумовлене заздалегідь число).

1.3. Алгоритми RED і WRED

Одним з можливих підходів при вирішенні проблеми перевантаження являється алгоритм RED (Random Early Detection). RED дозволяє маршрутизатору відкидати пакети, навіть коли в черзі ще є місце.

Алгоритм WRED використовує зважене значення довжини черги як чинник, що визначає вірогідність відкидання пакету. У міру зростання середньої довжини черги росте вірогідність відкидання пакетів. Якщо середня довжина черги менше певного порогу, вірогідність відкидання пакету дорівнює нулю. Невеликі кластери пакетів можуть успішно пройти через фільтр RED. Більші кластери можуть зазнати втрат. Це призводить до того, що TCP-сесії з великими відкритими вікнами мають велику вірогідність відкидання пакетів.

Головною метою алгоритму RED являється виключення ситуації, коли декілька TCP-потоків перевантажуються майже одночасно і потім синхронно починають процедуру відновлення.

Алгоритм RED дозволяє організувати черги таким чином, що їх розмір відстежує осциляції величини RTT (Real-time tactics).

RED намагається збільшити число коротких перевантажень і уникнути довгих. Завдання RED полягає в тому, щоб повідомити відправника про можливість перевантаження, і відправник повинен адаптуватися до цієї ситуації.

Існує модифікація алгоритму обробки черг WRED (Weighted Random Early Detection), широко використовувана в маршрутизаторах. Цей алгоритм застосовується і при реалізації DiffServ і гарантованій переадресації (AF), коли рішення про обробку пакету приймається в кожному транзитному вузлі незалежно (PHB - Per Hop Behavior). При цьому може враховуватися код DSCP (Differential Service Code Point) IP-заголовка [5].

Алгоритм WRED зручний для адаптивного трафіку, який формується протоколом TCP, оскільки тут втрата пакетів призводить до зниження темпу їх посилки відправником. Алгоритм WRED привласнює пакетам не IP-типу нульовий пріоритет, підвищуючи вірогідність їх втрати. Є можливість конфігурації WRED і WFQ для одного і того ж інтерфейсу.

WRED корисний для будь-якого вихідного інтерфейсу, де очікується перевантаження. Коли приходить пакет, обчислюється середнє значення довжини черги. Якщо вичислене значення менше мінімального порогу черги (T_1), пакет, що приходить, ставиться в чергу. Якщо вичислене значення лежить між мінімальним порогом черги для цього типу трафіку і максимальним порогом для інтерфейсу (T_2), пакет ставиться в чергу або відкидається залежно від вірогідності відкидання, встановленої для цього типу трафіку. І, нарешті, якщо усереднений розмір черги більше максимального порогу, пакет безальтернативно відкидається (див. рис. 1.3).

WRED статистично відкидає більше пакетів для сесій з великими потоками. З цієї причини TCP-відправники великих потоків будуть вимушені більшою мірою знизити потік, чим відправники малого трафіку.

Ні ATM-інтерфейси маршрутизаторів, ні ATM-комутатори не надають гарантій QoS для UBR віртуальних каналів. Маршрутизатор CISCO може специфікувати тільки пікову швидкість передачі (PCR) для UBR-канала.

Маршрутизатор автоматично задає параметри, які треба використовувати в обчисленнях WRED. Середнє значення довжини черги визначається на основі попереднього значення і поточного розміру черги, згідно

$$\alpha = \text{current\_queue\_size} \times 2 - \text{n}. \qquad (2.1)$$

де n - експоненціальний ваговий чинник, що конфігурується користувачем.

Велике значення цього чинника згладжує піки і провали значення довжини черги. Середня довжина черги не мінятиметься швидко. Процес WRED не відразу почне відкидати пакети при перевантаженні, зате продовжить відкидання, навіть коли перевантаження вже немає (черга вже скоротилася нижче за мінімальний поріг). Див. рис. 1.4 (крива з більшою амплітудую відповідає реальній довжині черги, друга, гладша крива відображує варіацію усередненої довжини черги).

При цьому можливе осцилювання довжини черги, що у багатьох випадках небажано, оскільки призводить до неефективного використання буфера і збільшення дисперсії часу доставки даних.

Коли n стає занадто великим, WRED не реагуватиме на перевантаження. Пакети посилатимуться або відкидатимуться так, як якби WRED не працював.

Для малих значень n середнє значення довжини черги практично співпадає з поточною її величиною, що викликає значні флуктуації середнього значення. В цьому випадку реакція WRED на перевищення довжини черги стає практично миттєвою. Якщо значення n занадто мале, WRED надзвичайно чутливий до флуктуацій трафіку, що може знижувати пропускну спроможність. Слід враховувати, що на практиці флуктуації трафіку мають дисперсію, у декілька разів перевершуючу Гауса.

Вірогідність відкидання пакету залежить від мінімального порогу (T_1), максимального порогу (T_2) і маркерного деномінатора вірогідності P_C. Коли середня довжина черги вище T1, RED починає відкидати пакети. Частота відкидань збільшується лінійно у міру зростання довжини черги до тих пір, поки усереднений розмір черги не досягне максимального порогу (T_2).

Маркерний деномінатор вірогідності PC дорівнює долі пакетів, що втрачаються, коли середня довжина черги дорівнює максимальному порогу. Наприклад, якщо P_C дорівнює 1/512, один з 512 пакетів відкидається, коли середня довжина черги досягає максимального порогу (див. рис. 11.2). Коли усереднена довжина черги перевищує T_2, відкидаються усі пакети. Мінімальний поріг T1 слід вибрати досить високим, щоб максимізувати використання каналу. Значення P_C можна задати і більше 1, але в цьому випадку довжина черги ніколи не досягне T_1.

У системах, де управління трафіком здійснюється з використанням зворотного зв'язку, можна досягти більшої ефективності. Одним з механізмів подолання перевантажень являється управління дозволом (admission control). Суть методу полягає в тому, що при реєстрації перевантаження не формуються більш ніяких віртуальних з'єднань до тих пір, поки ситуація не покращає. Альтернативним варіантом може служити рішення, де формування нового з'єднання дозволяється, але при цьому маршрутизація здійснюється так, щоб обійти вузли, в яких виявлено перевантаження (рис. 1.5).

На рис. 1.5 (верх) показаний приклад мережі з двома вузлами, що характеризуються перевантаженням (помічені чорним кольором). Припустимо, що необхідно прокласти віртуальний канал з вузла А у вузол Б. З графа маршрутів віддаляються переобтяжені вузли, після чого здійснюється прокладення шляху. У нижній частині рисунку жирними лініями показаний новий віртуальний канал.

Ще більш універсальним рішенням, придатним для роботи зі встановленням з'єднання і без, являється посилка пакетів блокування (choke Packets). Маршрутизатор зазвичай контролює завантаженість усіх своїх зовнішніх каналів L, яка може набувати значень від 0 до 1. Коли L досягає деякого порогового значення, відправникові посилається пакет блокування. При обчисленні L слід використовувати яку-небудь методику усереднювання, щоб уникнути занадто частих блокувань.

Коли відправник отримує пакет блокування, він повинен зменшити трафік, що посилається одержувачеві, на задане число відсотків. Оскільки на шляху до місця призначення може бути багато вузлів, це викличе серію пакетів блокування. Відправник повинен ігнорувати пакети блокування протягом деякого часу після отримання першого такого пакету. Після закінчення цього періоду відправник прослуховує канал упродовж аналогічного часу, чекаючи отримання нових пакетів блокування. Якщо такий пакет приходить, значить, канал все ще переобтяжений і відправник знову повинен знизити темп посилки пакетів. Якщо упродовж періоду прослуховування не приходить нових пакетів блокування, відправник може збільшити потік знову [6].

ЕОМ може знижувати трафік, коригуючи свої параметри, наприклад ширину вікна або темп передачі, на виході пристрою типу "діряве відро". Зазвичай перший блокуючий пакет зменшує потік удвічі, наступний на - 0,25 і так далі. Збільшення потоку також виконується аналогічними кроками. Існує велике число варіантів алгоритму управління потоком з використанням пакетів блокування. Параметром, який контролюється і визначає умову відправки пакету блокування, може служити довжина черги або міра заповнення буфера.

Практично усі описані алгоритми запобігання або подолання перевантаження припускають, що відправникові невідома міра заповнення буферів по дорозі і

він шукає оптимум методом проб і помилок. Звідси слідує висновок: треба прагнути до варіантів, де відправник отримує повну інформацію про міру заповнення буферів і про темп їх заповнення уздовж усього маршруту транспортування.

Ситуація перевантаження не завжди керується однозначно. Наприклад, при вступі на вхід пакетів від трьох джерел можлива ситуація, коли приймач посилає блокуючі пакети усім відправникам, а відгукнеться скороченням потоку тільки один. В результаті цей вузол, який "грає за правилами" (як це часто буває і в житті), опиняється в програші. У 1987 році Нагле був запропонований алгоритм fair queuing (справедлива черга, див. відповідний розділ WFQ вище). У цьому алгоритмі маршрутизатор організовує незалежні черги для пакетів, що поступають від різних джерел. Коли вихідний канал маршрутизатора виявляється вільним, він переглядає черги циклічно і відправляє черговий пакет. В результаті при n чергах після закінчення такого циклу переглядів-посилок буде послано по одному пакету з кожної черги. Цей алгоритм використовується в деяких ATM -перемикачах. Слід зауважити, що цей алгоритм дає переваги тим вузлам, які посилають довші пакети. Демерс (Demers) та ін. в 1990 році запропонували удосконалення алгоритму. У цьому варіанті організовується циклічний перегляд черг не попакетно, а побайтно. Система послідовно сканує черги і визначає положення кінців пакетів. Першими вирушають коротші пакети. Для ілюстрації пропонується розглянути рис. 1.6.

Пакети на рисунку мають від трьох до дев'яти октетів. Порядок пересилки октетів показаний в лівій частині рисунку. У відсутність вступу нових пакетів кадри, записані в буфер, будуть передані в порядку, представленому в правій частині рисунку. Особливістю цього алгоритму являється рівність пріоритету усіх каналів.

При передачі даних на великі відстані з великими значеннями RTT ефективність використання методу блокуючих пакетів знижується. Поки блокуючий пакет дійде через ряд проміжних вузлів до відправника, на вхід одержувача поступить велике число пакетів, які не лише посилять ситуацію перевантаження, але і можуть викликати втрату якоїсь їх долі, що у свою чергу може зажадати повтор-

ної пересилки кадрів, що йшли за ними. Для підвищення ефективності часто застосовується схема, при якій блокуючі пакети впливають на усі маршрутизатори по шляху свого дотримання. В цьому випадку зниження потоку можна чекати вже через час, рівний RTT до вузла, найближчого до одержувача пакетів. Така схема вимагає, щоб усі проміжні вузли мали досить місткі буфери, інакше можливі втрати.

У протоколі TCP використовується алгоритм управління трафіком, що називається "ковзаюче вікно". Тут розмір вікна, яке визначає число сегментів, що посилаються без отримання підтвердження, варіюється залежно від наявності втрат пакетів. При великій вірогідності втрати система переходить в режим, коли черговий пакет не посилається до тих пір, поки не буде підтверджено отримання передування.

При серйозних перевантаженнях, коли втрати стають значними, порушується механізм обчислення значень RTT і тайм-аутів, що може призводити до тяжко передбачуваних наслідків.

Слід звернути увагу, що в протоколі UDP якого-небудь механізму керування трафіком не передбачено. З цієї причини для мультимедійних завдань, слід передбачати інші, наприклад ICMP-способи пригнічення перевантаження (хоча це часто неприйнятно і рішення слід шукати в залученні методів резервування або DiffServ) [7].

Якщо інші способи випробувані, а перевантаження не зникло, маршрутизатор починає відкидати пакети, що приходять, які вже не може обробити. Найпростіше - це випадковий вибір відкинутих пакетів. Але це не краща тактика. У разі пересилки мультимедійних даних перевагу слід віддавати останнім отриманим пакетам, а "старі" пакети викидати. При передачі файлів, навпаки, "старий" пакет має пріоритет, адже якщо його відкинути, доведеться повторно передавати не лише його, але і усі наступні пакети. Деякі методи передачі зображення вимагають передачі час від часу усього кадру з наступною пересилкою тільки фрагментів, де сталися зміни. У таких умовах втрата пакету, що становить базовий кадр, менш бажана. Схожі обставини можуть виникати і в інших додатках. Можна позначати

пакети, привласнюючи їм певні рівні пріоритетів, що дозволить усвідомлено приймати рішення про відкидання того або іншого пакету в умовах перевантаження. У перспективі проблема може бути розв'язана на чисто комерційній основі: компонента трафіку, помічена як високопріоритетна, оплачуватиметься за вищим тарифом. У деяких мережах певна кількість пакетів об'єднується в групи, що утворюють повідомлення. Якщо один осередок такого повідомлення викинутий, усе повідомлення буде повторно переслано.

2. МОДЕЛІ ДИФЕРЕНЦІЙОВАНИХ ТА ІНТЕГРОВАНИХ ПОСЛУГ

2.1. Опис моделі DiffServ

У основі моделі DiffServ лежить визначення стандартизованого байта DSCP, колишнього байта типу обслуговування (TOS) в IPv4 і байта класу трафіку (Traffic Class octet) в IPv6 R. Від значення занесеного в цей байт залежатиме рішення про просування пакету з потоку даних в кожному проміжному вузлі на шляху пакету. Для пакетів, що передаються у зворотному напрямі потрібна окрема конфігурація устаткування. Для кожного з напрямів обміну конфігурація QoS виконується незалежно. Постачальники послуг (ISP) використовують модель DiffServ для надання своїм клієнтам набору пропозицій QoS залежно від вимог, що пред'являються клієнтом, до якості обслуговування. Клієнт може вибрати необхідні вимоги до QoS, встановивши для кожного свого пакету відповідне значення DSCP. Значення мітки DSCP визначає послідовність рішень про просування пакету в кожному проміжному вузлі мережі постачальника Інтернет послуг. Постачальник послуг може заздалегідь обумовити із замовником профіль послуг, в якому буде написано відповідність між інтенсивністю трафіку і рівнем обслуговування цього трафіку. Якщо виділена смуга буде перевищена клієнтом для якогось з рівнів обслуговування, то для незапланованих пакетів якість послуг буде негарантовано. Модель DiffServ визначає базові механізми (мітка DSCP і PHB), на основі яких виконується обслуговування пакетів. Використовуючи ці механізми, можна побудувати набір послуг. Кожна послуга може забезпечувати наступні характеристики: пропускну спроможність, затримку при передачі пакету, відсоток втрати пакетів в одному напрямі (при передачі уздовж мережевого маршруту). Після того, як тип послуги для маркованого пакету визначений, приймається рішення про просування пакету (PHB політика) для кожного вузла мережі, який підтримує цю політику. Кожній PHB політиці ставиться у відповідність значення мітки DSCP. Вузли все-

редині DiffServ домена визначають PHB політикові для пакетів на підставі тих, що зберігаються в них DSCP міток.

Прикордонні вузли DiffServ домена формують трафік, що надходить в домен.

Формування трафіку полягає в наступному: проводиться класифікація пакетів і обмеження що входить в DiffServ домен трафіку. В результаті класифікації прикордонний вузол DiffServ домена визначає для кожного пакету відповідну цьому пакету PHB-політику і ставить в заголовку пакету значення поля DSCP (це виконується відповідно до профілю трафіку). PHB політика, відповідна певному класу трафіку, залежить від цілого ряду умов [8]:

1) від інтенсивності вхідного потоку даних для цього класу трафіку;

2) від розподілу ресурсів для цього класу трафіку;

3) від відсотка втрат пакетів в потоці.

На рис. 2.1 приведена схема моделі QoS в DiffServ домені.

Класифікатор пакетів вибирає пакет з потоку і аналізує частину заголовка пакету. Класифікація пакетів може виконуватися на основі поля DSCP, або інших полів заголовка. Після класифікації пакету йому приписується відповідний клас трафіку. Маркування пакетів полягає в записі або перезаписі поля DSCP залежно від класу трафіку, до якого відноситься пакет.

Функція обмеження трафіку перевіряє, чи відповідає трафік заданому профілю. Однією з причин перевантаження часто є імпульсні завантаження сегменту мережі або мережевого пристрою. З цієї причини будь-які заходи (наприклад, буферизація), які можуть вирівняти потік пакетів, безумовно поліпшать ситуацію (наприклад, traffic shaping в мережах ATM).

У TCP ж з його вікнами імпульсні завантаження зумовлені. Для систем без зворотного зв'язку вирішення проблеми вирівнювання швидкості передачі даних може бути знайдене за допомогою алгоритму leaky bucket (“діряве відро”). Суть цього алгоритму полягає в тому, що на шляху потоку встановлюється буфер, витікаючий потік якого постійний і узгоджений з можливістю приймача. Якщо бу-

фер переповнюється, пакети втрачаються. Втрата пакетів річ мало приємна, але це блокує процеси, які можуть привести до колапсу сегменту або усієї мережі.

Там, де втрата пакетів небажана, можна застосувати гнучкіший алгоритм - Token Bucket ("маркерне відро"). Алгоритм token bucket припускає наявність в буферному пристрої (чи програмі) деякої кількості маркерів. При надходженні пакетів на вхід буфера, маркери використовуються для їх транспортування на вихід. Подальша передача даних на вихід залежить від генерації нових маркерів. Пакети, що надходять ззовні, тим часом накопичуються в буфері. Таким чином, повної гарантії відсутності втрат ми не маємо і тут.

Алгоритм token bucket дозволяє в деяких випадках знизити вірогідність втрати пакету. Якщо буферний пристрій "змонтований" усередині EOM-відправника, втрат можна уникнути зовсім, блокуючи передачу при заповненні буфера. Як у одному, так і в іншому алгоритмі мірою передаваної інформації може бути не пакет, а n-байт (де n деяке обумовлене заздалегідь число). У системах, де управління трафіком здійснюється з використанням зворотного зв'язку, можна досягти більшої ефективності.

Одним з механізмів подолання перевантажень є управління дозволом (admission control). Суть методу полягає в тому, що при реєстрації перевантаження не формуються більш ніяких віртуальних з'єднань до тих пір, поки ситуація не покращає. Альтернативним варіантом може служити рішення, де формування нового з'єднання дозволяється, але при цьому здійснюється маршрутизація так, щоб обійти вузли, в яких виявлено перевантаження.

Після маркування пакетів функція відкидання пакетів видаляє ті пакети, які не задовольняють параметрам заданого профілю трафіку. Профіль трафіку описаний в SLA - договір надання послуг з сервіс провайдером. Трафік, що посилається користувачами понад договір, може бути відкинутий сервіс провайдером.

На сьогодні стандартизовані деякі значення мітки DSCP (політики AF і EF). Стандартне значення (значення за умовчанням) мітки DSCP визначене як 000 000. Класи міток DSCP зворотньосумісні з кодами IP пріоритету див. табл. 2.1 [9].

Існують дві основні політики PHB - це PHB політика негайної передачі (EF PHB) і PHB політика гарантованої доставки (AF PHB). Для цих політик виділено два класи EF (expedited forwarding) і AF (assured forwarding).

EF служить для маркування пакетів, які вимагають термінової передачі. EF пакети отримують найбільш високий рівень обслуговування. Значення мітки DSCP, що встановлюється в EF пакетах рівне 101110. Трафік, що належить до EF класу, зазнає при проходженні DiffServ домена малу затримку, низький рівень втрат і гарантовану смугу пропускання. Політика EF застосовується для даних таких додатків, як передача голосу (VoIP) і додатків відеоконференцій. Основна причина затримок пакетів і дисперсії затримок пакетів - це виникнення великих накопичених черг. Черги виникають в переобтяжених ділянках мережі. Причиною виникнення перевантаження в мережі є довготривале перевищення інтенсивності вхідного потоку даних над інтенсивністю вихідного потоку. Для зменшення затримки пакетів можна обмежувати максимальну інтенсивність вхідного потоку даних значенням мінімальної інтенсивності вихідного потоку даних. Це можна робити двома способами або вести буфер, у разі переповнювання якого усі пакети, що знову поступають, втрачатимуться, або організувати контроль за рівнем заповнення буфера і відкидати пакети залежно від рівня заповнення буфера. Другий спосіб дає можливість через появу відкинутих пакетів, повідомити відправника, щоб він знизив швидкість передачі пакетів. Як ще один спосіб скорочення затримки при передачі пакетів можна порекомендувати використовувати спільно з DiffServ технологію MPLS, в цьому випадку на порядки скорочуються розміри маршрутних таблиць і скорочується час доставки комутованих пакетів.

PHB EF задає обмеження на вихідну смугу трафіку, а інтенсивність потоку трафіку, що входить, повинна контролюватися формувачами трафіку, які реалізовані в пограничних маршрутизаторах DiffServ домена. Однією з вимог політикою PHB EF є недопущення утворення черги пакетами, що входять, отже, смуга витікаючого потоку має бути більшій або рівнішій смузі потоку трафіку, що входить. Маршрутизатори дозволяють виділяти на своїх інтерфейсах ресурси, необхідні

для забезпечення певної смуги для витікаючого трафіку. При передачі EF трафіку через переобтяжений сегмент мережі, резервування смуги під трафік здійснюється за рахунок застосування різних механізмів обслуговування черг таких, як CBWFQ, WRR. При цьому EF трафіку призначається вага, яка відповідає смузі пропускання, що перевищує інтенсивність EF трафіку, що входить. EF трафік також можна перенаправляти в пріоритетну чергу, але при цьому нестача ресурсів каналу торкнеться усіх інших потоків витікаючих через переобтяжений інтерфейс. Щоб уникнути монопольного захоплення смуги EF трафіком задають величину максимальної смуги, що обробляється за допомогою пріоритетної черги, увесь надлишковий EF трафік відкидається.

AF політика служить для задавання декількох рівнів надійності доставки IP-пакетів отриманих від клієнта. У AF PHB політиці закладена можливість надання різних рівнів обслуговування для кожного з чотирьох класів AF трафіку. Кожному AF класу відповідає свій власний буфер. Це дозволяє ефективно управляти смугою пропускання. Кожен AF клас містить три рівні пріоритету відкидання пакетів (низький, середній і високий). Це дозволяє для кожного AF класу конфігурувати механізм управління чергою (RED)

AF PHB політика визначає чотири рівні обслуговування. У кожному з цих рівнів знаходиться три пріоритети відкидання пакетів (див. табл. 2.2).

Таблиця 2.2

Чотири рівні обслуговування AF PHB політики

Пріоритет відкидання	Клас 1	Клас 2	Клас 3	Клас 4
Низький	001010	010010	011010	100010
Середній	001100	010100	011100	100100
Високий	001110	010110	011110	100110

2.2. Модель гарантованого обслуговування IntServ

Приведемо, як порівняння, короткий опис моделі IntServ (Integrated Services). Як засіб інформування мережі про потреби різних потоків трафіку в моделі IntServ використовується сигнальний протокол RSVP - протокол резервування ресурсів. Протокол RSVP використовується ЕОМ користувача для того, щоб зарезервувати для додатка необхідний рівень QoS (наприклад, певний рівень смуги пропускання). Ініціатива резервування йде від самого користувача (відправника). Для порівняння, в моделі DiffServ виділення ресурсів відбувається завчасно на устаткуванні сервіс провайдера. RSVP використовується також маршрутизаторами для доставки запитів QoS усім вузлам уздовж шляху інформаційного потоку. Також він використовується для установки і підтримки необхідного рівня послуг. RSVP-запити дозволяють резервувати певні мережеві ресурси, які потрібні, щоб забезпечити необхідний рівень QoS уздовж усього маршруту передачі даних. RSVP резервує ресурси тільки для одного з напрямів трафіку і тільки на вимогу одержувача. Протокол RSVP намагається зарезервувати ресурси для потоку даних на кожному з вузлів, використовуваних для передачі цього потоку. RSVP працює поверх протоколів мережевого рівня IPv4 або IPv6. Протокол RSVP сигналізує про запити резервування ресурсів по доступному в мережі шляху, що маршрутизується [10].

При визначенні шляху для даних і управляючого трафіку RSVP, покладаються на протокол маршрутизації, який використовується в мережі. Після того, як інформація протоколу маршрутизації адаптується до змін в топології мережі, запити резервування протоколу RSVP переносяться на новий шлях. RSVP резервування завжди виконуватися по одному і тому ж одноадресному шляху або по багатоадресному дереву. Якщо виходить з ладу канал зв'язку, маршрутизатор повинен повідомити про цьому RSVP, щоб RSVP-повідомлення, які генерує RSVP, передавалися їм по новому шляху.

Пакети, що йдуть від додатка на ЕОМ-відправника до додатка на ЕОМ-одержувача, формують окремий потік. Для управління доступом потоку до ресурсів каналу, вимоги потоку представляються у вигляді параметрів об'єкту FlowSpec.

RSVP одержувачі можуть передавати по RSVP мережі наступні вимоги до потоку: середню швидкість передачі даних, максимальний розмір сплеску і параметри QoS (регульованого навантаження або гарантована бітова швидкість).

2.3. Спільне використання механізмів IntServ і DiffServ

Існує пропозиція по спільному використанню технологій DiffServ і IntServ . Пропонується використовувати IntServ в пограничних вузлах мереж, а усередині мереж використовувати DiffServ. IntServ дозволяє передати вимоги QoS через мережу, що складається з набору різнорідних мереж. Щоб відбулося резервування ресурсів через різнорідну мережу, необхідно, щоб кожен вузол підтримував протокол RSVP.

За допомогою сигнального протоколу RSVP, PHB політики повідомляють пограничні вузли DiffServ доменів про вимоги до якості обслуговування з боку RSVP потоків. За допомогою протоколу RSVP можна встановити відповідність між запитами до якості обслуговування RSVP потоків і класами послуг DiffServ. Наприклад, гарантованому обслуговуванню RSVP може бути поставлена у відповідність DiffServ послуга негайної передачі пакетів (EF). В цьому випадку сигналізація про якість обслуговування є добре масштабованим рішенням у великих мережах, оскільки протокол RSVP виконується тільки в пограничних вузлах DiffServ домена, див. рис. 2.1.

Процес установки відповідності між резервуванням ресурсів протоколу RSVP і DiffServ класами проводиться на межі DiffServ мережі. Пограничні додатки, заздалегідь оповіщені про існуючу політику, можуть використовувати RSVP щоб повідомити мережі свої вимоги до якості обслуговування. Декілька резервувань ресурсів протоколу RSVP можуть бути об'єднані в одне агреговане велике резервування. Агреговане резервування ресурсів є потужним, повільно регульованим станом резервування, який дозволяє понизити об'єм сигнальної інформації в базовій мережі. Аналогічно випадку звичайного резервування ресурсів, прото-

кол RSVP може встановити відповідність між агрегованим резервуванням і DiffServ -класом.

Не секрет, що неможливо розгорнути протокол RSVP (чи будь-який новий протокол) в усьому Інтернет одночасно. Більше того, протокол RSVP ймовірно ніколи не буде розгорнутий повсюдно. RSVP повинен гарантувати коректну роботу додатків, коли два RSVP маршрутизатора об'єднані один з одним через мережеву область, що не підтримує цей протокол. Звичайно, проміжна мережева область, позбавлена підтримки RSVP, не здатна здійснювати резервування ресурсів.

Протокол RSVP може працювати і через не підтримуючі його мережеві області. Якщо така область має достатню смугу пропускання, вона не сильно вплинемо на якість послуг, що надаються. Як підтримуючі RSVP так і не підтримуючі RSVP маршрутизатори переадресують повідомлення RSVP PATH за адресою місця призначення, використовуючи свої локальні таблиці маршрутизації. Таким чином, на маршрутизацію повідомлень RSVP PATH, не чинить вплив наявність проміжних маршрутизаторів не підтримуючих RSVP. Коли повідомлення RSVP PATH проходить через мережеву область, не підтримуючу RSVP, воно, прямуючи до наступного вузла, підтримуючого RSVP, несе в собі IP-адресу останнього маршрутизатора, підтримуючого RSVP. Повідомлення RSVP RESV переадресовується безпосередньо наступному RSVP -маршрутизатору на шляху до відправника.

Про спільне застосування технологій IntServ і DiffServ, детальніше можна дізнатися в .

У IntServ постійно треба вирішувати проблему приналежності пакету тому або іншому субпотоку, для цього необхідно проаналізувати велике число полів заголовка, а іноді і поля даних. У DiffServ аналізується тільки поле DSCP.

Крім того, якщо використовувати DiffServ у поєднанні з технологією MPLS, то розмір маршрутних таблиць скоротиться на порядок і комутація пакетів ще сильніше прискориться в порівнянні з IntServ.

Технологія IntServ може знайти додатки на рівні локальних мереж.

Простота реалізації DiffServ не забезпечує простоту і легкість його використання. Як уже згадувалося, DiffServ будується на базових механізмах, якими є поле DS (де зберігається DSCP мітка) і PHB політика (це локальна логіка мережевого вузла, то - як вузол повинен відреагувати на пакет із заданим значенням DSCP мітки). Можливості QoS залежать від побудованої політики PHB. Кожному значенню DSCP мітки відповідає налаштування двох алгоритмів в мережевому пристрої - алгоритму диспетчера (WRR, PQ, WFQ, LLQ, CBWFQ) і алгоритму активного управління чергою (WRED, RED) [11].

2.4. Алгоритми активного управління чергами

Один із способів забезпечення якості обслуговування - це боротьба з перевантаженнями в мережі. Уперше термін "управління перевантаженнями (congestion control)" ввів в 1980 році Ван Якобсон. Алгоритм по боротьбі з перевантаженнями Ван Якобсон реалізував в протоколі TCP. У наш час колишніх способів боротьби з перевантаженнями стало недостатньо. Кількість користувачів Інтернет і різних мережевих застосувань збільшується з кожним днем, мережа потребує засобів, які б забезпечили підтримку як існуючих, так і таких, що з'являються застосувань і служб. На сьогодні Інтернет може забезпечити усього лише негарантовану доставку даних. Негарантована доставка даних не припускає надання яких-небудь гарантій, що стосуються часу і самого факту прибуття пакету в пункт призначення. Слід зазначити, що відкидання пакетів відбувається тільки у момент перевантаження мережі.

Як правило, передавані по мережі, пакети розрізняються на основі п'яти полів заголовка IP, які однозначно визначають потік даних, - адреса джерела IP-пакету, адреса призначення IP-пакету, поле протоколу IP, порт джерела і порт призначення. Потік інформації складається з пакетів, згенерованих додатком, що виконується на комп'ютері-джерелі, і що призначаються для передачі додатку, що виконується на комп'ютері-приймачі. Пакети, що належать одному потоку, мають однакові значення усіх п'яти полів в заголовку IP-пакету. Системи ядра IP мережі

повинні мати можливість диференціювання і обслуговування різних типів мережевого трафіку залежно від вимог, що пред'являються ними.

Вирішенням цієї проблеми займаються протоколи транспортного рівня. Можливість надати потоку певну якість по затримці і смузі сильно залежить від протоколу транспортного рівня, а також від протоколів транспортного рівня фонових потоків. На сьогодні в мережі переважає плавний трафік від TCP потоків, але так само велику частку сучасного трафіку складають потокове відео і аудіо дані реального часу.

Мультимедіа трафік в сучасних додатках як транспорт використовує протокол RTP/UDP. Наведемо простий приклад: Нехай через маршрутизатор одночасно проходить декілька потоків TCP і мультимедіа-UDP потоків. Сталося переповнювання буфера маршрутизатора, почали втрачатися пакети. TCP джерела відразу знижують швидкість передачі, смуга займана ними звужується. UDP джерела не контролюють доставку пакетів і в UDP немає алгоритму запобігання перевантаженням. Смуга звільнилася від TCP буде зайнята UDP потоками. Цей приклад показує несправедливість, що виникає відносно TCP трафіку. Виникає завдання справедливого ділення ресурсів маршрутизатора між потоками, що належить не лише різним класам обслуговування, але і передаваним посереднецтвом різних протоколів транспортного рівня.

Виникає проблема: Як передавати мультимедіа не на шкоду іншим потокам? QoS покликаний забезпечити крізну гарантію якості передачі даних, і заснований на системі правил контроль за засобами підвищення продуктивності IP-смережі, такими як: механізм розподілу ресурсів, комутація, маршрутизація, механізми обслуговування черг і механізми відкидання пакетів. В цілому ж оптимальні умови функціонування усіх додатків включають вимоги до виділення певних мережевих ресурсів в межах смуги пропускання, затримки і числа втрачених пакетів.

Для передачі аудіо і відео даних вимагаються різні вимоги до QoS. Практично кожен мережевий інтегратор стикається з проблемою передачі різних типів інформації по одній мережі (це може бути звукова інформації, відео, і прості да-

ні). Для передачі відеоданих потрібна висока пропускна спроможність і стабільний час затримки при передачі. Для уникнення ефекту "снігу" і інші спотворення зображення потрібний стаціонарний потік даних. При інтерактивній передачі звуку вимагається менше пропускної спроможності каналу, чим при передачі відео, але потрібна мала затримка через мережу, інакше виникає "луна-камера". Передача файлів вимагає високої пропускної спроможності, але на відміну від більшості інших видів мережевого трафіку, найменш чутлива до тривалих і непостійних затримок в мережі.

У мережевого інтегратора виникає завдання про надання необхідної якості до відео, аудіо і потокам даних як на рівні опорної мережі інтегратора, так і на рівні останньої милі, наприклад для клієнтів які заплатили за QoS і вимагають особливого обслуговування. Ті ж складнощі виникають на стику мереж мережевих провайдерів. Для забезпечення необхідних вимог до різних потоків даних використовуються дві методики QoS:

- управління перевантаженням (congestion management);
- запобігання перевантаженням (congestion avoidance).

Перший метод заснований на привласненні квот і пріоритетів потокам, і у разі перевантаження, потоки отримають якість обмежену їх квотою і пріоритетом (наприклад WFQ).

Другий метод обмежує розмір черги, сигналізуючи джерелам даних про необхідність зменшити швидкість передачі інформації (наприклад WRED). У IOS - операційної системі компанії “Cisco” ці методи настроюються як окремо, так і в поєднанні один з одним.

У публікаціях R проводиться дослідження переваг WRED по відношенню до технології Tail Drop - старої технології управління перевантаженнями. У R приведений звіт про експериментальне порівняння цих двох технологій, учені доводять неефективність вживання WRED замість Tail Drop, коли в WRED конфігуровані параметри за умовчанням.

Ведеться активне дослідження в області методів що обмежують розмір черги. Можна перерахувати лише деякі, найбільш відомі модифікації алгоритму

RED: WRED, GRED (Gentle RED) R, DRED (Dynamic RED), SRED (Stabilized RED) , ARED (Adaptive RED) R, RED - PD R.

Також існує ряд робіт у яких автори намагаються знайти оптимальний набір параметрів WRED [12].

У роботі наводяться результати моделювання системи із зворотним зв'язком "WRED і TCP Reno", виводяться складні аналітичні вирази для значень оптимальних параметрів WRED. Аналітична модель будується для TCP сесії, що довго триває. Метою роботи являлося загальне дослідження механізмів управління перевантаженнями в Інтернет, зокрема досліджувався механізм управління перевантаженнями RED. Як один з результатів, був виявлений факт виникнення осциляції довжин черг як наслідки роботи алгоритму RED. Приводилися рекомендації по реалізації RED в мережевих пристроях.

Робота лягла в основу безлічі інших робіт побудованих на поліпшенні і ускладненні первинної роботи. У роботі , що базується на публікаціях і , теж виконується моделювання WRED для TCP Reno і тривалих сесій. Досліджується вплив кожного параметра WRED на роботу системи WRED -TCP. Мета цієї роботи полягала в знаходженні таких параметрів WRED, щоб звести до мінімуму осциляції довжин черг, виявлених в попередній роботі. У роботі , що базується в основному на публікації , виконується моделювання WRED і GRED і виконується їх порівняння. Перевіряється запропонований оптимальний набір параметрів з роботи і робиться висновок, що цей набір не дає ніякої переваги у використанні WRED в порівнянні з Tail Drops. Як виведення статті витікає, що замість RED слід використовувати його модифікований варіант GRED.

Не дивлячись на значний об'єм публікацій по цій темі, залишається проблема вибору налаштувань параметрів для алгоритму WRED. Усі дослідники WRED згодні з тим, що вплив алгоритму на якість передачі потоків сильно залежить від правильного завдання його параметрів, але і досі немає зрозумілої інструкції, як на практиці вибирати значення цих параметрів.

У роботі виконано модельне дослідження алгоритмів AQM. Алгоритми AQM (Active queue management) застосовуються не лише для запобігання перева-

нтаженням, але і для цілей диференційованого обслуговування пакетів, як, наприклад, в архітектурі DiffServ. Відомо, що процедура коректного завдання параметрів алгоритму AQM складна і дослідження в області AQM показали, що якість роботи AQM сильно залежить від миттєвого стану мережі. У роботі приведено детальне модельне дослідження алгоритму Adaptive RIO (A - RIO). Алгоритм A - RIO є гібридом алгоритмів Adaptive RED (A - RED)R і RIO . У роботі досліджуються переваги алгоритму A - RIO над старими алгоритмами, також досліджується питання осциляції довжини поточної черги. A - RIO може використовуватися в AF сервісах для створення послуг з гарантованою затримкою при передачі пакету. Моделювання виконане в середовищі NS - 1.

У роботі виконано дослідження оптимальної форми кривою відкидання пакетів для RED. Результати досліджень в і відрізняються. Ми привели статті, в яких пропонуються ті або інші зміни у вже наявному алгоритмі, а потім в них на прикладі моделювання в середовищі NS - 2 доводяться переваги зроблених змін.

У роботі розглянута процедура аналітичного отримання функції вірогідності відкидання пакету для алгоритму RED. Розглянутий там підхід заснований на системі рівнянь що враховують розмір TCP вікна, а також вимоги стійкості системи. Стабільність розуміється в сенсі пригнічення осциляції довжини черги в стаціонарному режимі роботи. У кінці виводиться вид залежності вірогідності відкидання пакету від поточної довжини черги. Причому при виводі використовуються вимоги стійкості системи, якщо на практиці вдасться добитися виведеної в статті залежності вірогідності відкидання пакету від довжини поточної черги (кубічна залежність), то в стаціонарному випадку осциляції довжини черги не буде.

У роботі R автори запропонували рекомендований ними набір параметрів RED, вони виходили із здорового глузду і експерименту. Людини, перед якою стоїть завдання конфігурувати RED в мережі сервіс провайдера, не обрадує думку про те, що спочатку йому доведеться провести серію експериментальних вимірів, щоб виявити який набір налаштувань RED дає кращий виграш в продуктивності. Робота R - являється засадничою роботою для усього напряму AQM.

У роботі R автори будують аналітичну модель роботи RED для плавного і переривчастого трафіку в припущенні, що трафік описується пуасоновським процесом. Проте така модель занадто проста і неправдоподібна. Приклад невідповідності їх моделі трафіку реальному трафіку - процес пригнічення перевантаження в TCP.

У роботі R автори досліджували динаміку поведінки RED, але модель вхідного трафіку в їх роботі була вибрана штучно, трафік замінювався процесом MMPP (Markov modulated Poisson Process).

У цій роботі не висуваються ніяких моделей відносно вхідного трафіку, а за допомогою моделювання в NS - 2 і експерименту виводимо рекомендації по установці параметрів WRED.

Найбільш перспективними технологіями для заміни WRED є ARED R і GRED R, але скільки часу пройде до їх реалізації в мережевих пристроях сказати важко.

Окрім згаданих способів існує напрям, який займається поліпшенням наявних мережевих протоколів, з тим, щоб поліпшити якість роботи цих протоколів. Пропонуються нові модифікації протоколів TCP (Tahoe, Reno, Vegas, T/TCP, SACK, NTCP) і UDP (DCCP). В сумі усі ці технології покликані розв'язати проблему QoS, але для їх реалізації вимагається адаптація програмного забезпечення у користувачів і в мережевих пристроях Інтернет.

Далі ми розглянемо алгоритм WRED, оскільки цей алгоритм реалізований практично в усіх сучасних маршрутизаторах, а інші його модифікації лише бурхливо обговорюються і не мають практичної реалізації в мережевих пристроях. До позитивних сторін алгоритму WRED можна віднести:

1) Средньозважена довжина черги для TCP з'єднань підтримується на низькому рівні, відповідні затримки при передачі теж низькі.

2) WRED на відміну від Tail Drops добре справляється з трафіком, в якому є сплески.

3) WRED допомагає позбавитися від "глобальної синхронізації TCP джерел" Опишемо параметри, які беруть участь в роботі алгоритму.

2.5. Робота алгоритму WRED і його параметри

Маршрутизатор, що підтримує механізм RED, постійно обчислює значення середньозваженої довжини черги [13].

У роботі досліджувалася можливість впливати на якість послуг для різних потоків за допомогою алгоритму WRED. WRED (модифікація RED для декількох потоків) встановлювався в "вузькому місці" каналу. У каналі, через вузьке місце пропускалися декілька потоків, кожен потік належав окремому класу обслуговування, тобто для кожного потоку маршрутизатор застосовував унікальний набір параметрів RED, визначений значенням DSCP мітки потоку. DSCP мітки встановлювалися для кожного потоку відправником.

Основна ідея WRED полягає в тому, щоб підтримувати середньозважену довжину черги (тобто середнє число пакетів в буфері маршрутизатора) на низькому рівні. WRED-маршрутизатор відкидає пакети, що входять, з вірогідністю, яка пропорційна середньозваженій довжині черги. Перевага WRED, в тому, що у нього простий алгоритм роботи, і він вже реалізований практично в усіх маршрутизаторах "Cisco".

Приведемо детальний опис алгоритму запобігання перевантаженням RED R. Міркування, що приводяться, вірні для пакетів з однаковими мітками DSCP, нижче наводиться алгоритм RED в тому вигляді, як він реалізований в маршрутизаторах "Cisco" і в програмі моделювання NS - 1. Для кожного пакету, що прибуває, обчислюється середньозважена довжина черги Q(t).

$$Q(t+\Delta) = (1 - w_q)Q(t) + w_q q(t) \qquad (2.1)$$

де t - час, Δ - приріст часу між послідовними вступами пакетів, q(t) - поточне значення довжини черги.

Параметри алгоритму RED, що міняються:

Q(t) — середньозважена довжина черги;

q_time — час, коли сталося обнулення довжини черги;

count — число пакетів з моменту останнього відкинутого пакету.

Фіксовані параметри:

w_q — ваговий коефіцієнт;

T_1 — нижній поріг середньозваженої довжини черги;

T_2 — верхній поріг середньозваженої довжини черги;

p_c — максимальне значення вірогідності відкидання пакету, після якої пакети починають відкидатись з вірогідністю 1 $\left(0 < p_c \le 1\right)$.

Допоміжні параметри:

p_a — поточна вірогідність відкидання пакету;

q — поточна довжина черги;

t — поточне значення часу;

f(t) — лінійна функція часу t.

Якщо $T_1 \le \overline{Q}(t) \le T_2$, обчислюється вірогідність p_a (див. визначення нижче) відкидання пакету. Якщо $T_2 < \overline{Q}(t)$ то пакет безумовно відкидається.

Періоди, коли пакетів не поступало - періоди простою теж враховуються при обчисленні середньозваженої довжини черги. Позначимо m - число пакетів, яке могло бути передане в мережу за час простою. Після періоду простою алгоритм вважає, що ці m пакетів в цей період поступили в порожню чергу. $\overline{Q}(t)$ міняється від T_1 до T_2, вірогідність відкидання — лінійно зростає від 0 до p_c.

$$p_b(t) \leftarrow \frac{p_c\left(\overline{Q}(t) - T_1\right)}{\left(T_2 - T_1\right)} \tag{2.2}$$

Остаточна вірогідність відкидання (з урахуванням періодів простою):

$$p_a(t) \leftarrow \frac{p_b}{1 - \text{count} \cdot p_b} \tag{2.3}$$

Ініціалізація алгоритму:

$\overline{Q} \leftarrow 0$

$\text{count} \leftarrow -1.$

Для кожного новоприбулого пакету, обчислюється $Q(t+\Delta)$

if $q > 0$

$$\overline{Q} = (1 - w_q)\overline{Q} + w_q q(t)$$

else

$$m = f(\text{time} - q\_\text{time})$$

$$\overline{Q} = (1 - w_q)^m \overline{Q}$$

if $T_1 \le \overline{Q}(t) \le T_2$.

Обчислюємо вірогідність відкидання.

$$p_b(t) \leftarrow \frac{p_c\left(\overline{Q}(t) - T_1\right)}{(T_2 - T_1)} \quad (\text{див. рис. 1.3});$$

$$p_a(t) \leftarrow \frac{p_b}{1 - \text{count} \cdot p_b}.$$

З вірогідністю p_a відкинути пакет, що поступив count=0.

else if $T_2 \le \overline{Q}(t)$ відкинути пакет count $= 0$

else count $= -1$

when $q = 0$

$q - \text{time} = \text{time}$.

Чим більше значення w_q, тим сильніше вклад попереднього значення довжини поточної черги при обчисленні середньозваженого значення черги, і тим гірше згладжування коливань довжини черги. У той же час не слід вибирати значення w_q занадто малим. В цьому випадку RED не зможе досить швидко реагувати на виникнення перевантаження. Наслідки роботи механізму RED для потоку А - це випадкова вибірка пакетів, що поступили на вхід, з потоку А, які були відкинуті RED. Має місце твердження: потік з великою часткою втрачених пакетів ймовірно має велику частку в загальному вхідному потоці (при рівних частках в загальному витікаючому потоці).

Існує два типи втрат: випадкові (random), і вимушені (forced) втрати. Говорять, що втрачений пакет був вимушено відкинутий, якщо це сталося через переповнювання FIFO буфера або якщо середньозважена довжина черги, вичислена RED, перевищила поріг T_1. У інших випадках пакет вважається "випадково" відкинутим.

У разі вірної конфігурації маршрутизатора, більшість відкинутих пакетів повинна належати типу "random drops" R. Якщо ця вимога не дотримується, то при переповнюванні буфера або перевищенні порогу T_2, RED відкидає усі пакети, що входять, поки не звільниться місце в буфері, або не зменшиться значення середньозваженої довжини черги.

Коли середньозважена довжина черги перевищує нижній поріг T_1, це свідчить про зародження перевантаження, RED починає застосовувати "метод випадкової вибірки" R для вибору з потоку вхідних пакетів, тих, що треба відкинути. У R описано два варіанти роботи алгоритму RED. У пакетному режимі, тобто для заданої довжини середньозваженої черги, пакет, що знову поступив, має вірогідність відкидання незалежну від його довжини в байтах. У байтовому режимі вірогідність відкидання пакету є функцією від довжини пакету в байтах. У маршрутизаторах Cisco застосовується комбінована метрика втрат.

Комбінована метрика втрат: для вибраного інтервалу часу і для вибраного потоку як метрика втрат береться відношення числа втрачених пакетів до загального числа втрачених пакетів за той же часовий інтервал.

Для WRED в байтовому режимі, таким чином, певна метрика задає долю потоку в загальному потоці, що входить, в байтах/секунду. Для WRED в пакетному режимі ця метрика задає частку потоку в пакетах/секунду [14].

Як правильно вибрати інтервал часу для обчислень по комбінованій метриці втрат описано в R. Припустимо, що усі перехідні процеси завершилися і втрати вийшли на стаціонарний рівень. Нехай усього за період спостереження відкинуто n пакетів, а вірогідність покидька пакету, що приходить, постійна і дорівнює p.

Якщо для і-го потоку доля вихідної смуги p_i те $S_{i,n}$ - число відкинутих пакетів з і-го потоку, $S_{i,n} = n \cdot p_i$.

Вірогідність того, що потік отримає в с разів більше втрачених пакетів в порівнянні з $S_{i,n}$:

$$P\left(S_{i,n} \geq c\, p_i\, n\right) \leq \left[\left(\frac{1}{c}\right)^{c \cdot p_i} \left(\frac{1-p_i}{1-c \cdot p_i}\right)^{t-c \cdot p_i} \right]^n . \qquad (2.4)$$

Цей результат R показує, що в стаціонарному випадку вірогідність того, що втрати пакетів перевищать значення встановлене виходячи з частки витікаючої смуги пропускання, прагнути до нуля. Це означає, що доля потоку, що перевищує виділену квоту смуги у витікаючому каналі буде втрачена, а вірогідність того, що на додаток до цієї частки загубляться ще пакети з цього потоку прагне до нуля (у стаціонарному випадку).

До виходу на стаціонарний режим можливі коливання середньозваженої довжини черги і як наслідок виникнення непотрібних втрат. Шляхом правильного підбору параметрів WRED можна уникнути цих непотрібних втрат.

На жаль, досі не існує однозначних рецептів, як вірно виставити налаштування WRED. Виробник устаткування "Cisco" лише рекомендує користуватися установками за умовчанням.

Як показав досвід, модельні дослідження не завжди співпадають з практикою. Незважаючи на заявлену реалізацію WRED в маршрутизаторах Cisco ми стикаємося з недоліком інформації про багато інших особливостей маршрутизатора, які побічно впливають на результат роботи WRED. У результаті ми вимірюємо відгук маршрутизатора на вхідний трафік, який відрізняється від відгуку, обчисленого за допомогою середовища моделювання NS-1. NS-2 широко використовується для тестування нових протоколів, а також для перевірки їх спільної роботи із старими протоколами. Усі реалізовані в NS - 2 протоколи в точності відповідають теоретичній моделі. Мінус NS - 2 в тому, що моделі коректні з точки зору теорії не завжди відповідають поведінці реальних мереж.

Крім перерахованих складнощів виникають проблеми при використанні RED, а саме: коли потоки передаються за допомогою TCP, то RED, викликаючи втрати, примушує джерело знизити швидкість передачі і тим самим здійснюється зворотний зв'язок між відправником і маршрутизатором, чий інтерфейс переобтяжений. Але останнім часом через глобальну мережу активно взаємодіють додатки реального часу такі як потокове відео (streaming video), IP-телефонія, відео конференції, та мережеві ігри. Ці додатки швидко розвиваються і вимагають усе більш швидкісних каналів. Причому тривалість таких з'єднань теж збільшується. Як правило, усі перераховані додатки використовують як транспортний протокол - UDP. Поєднання RED і UDP менш ефективно, чим RED і TCP. Не дивлячись на те, що TCP добре контролює послідовність доставки пакетів і їх черговість, він не може гарантувати своєчасність доставки, тобто TCP вносить непередбачувану затримку. Із сказаного виходить, що потрібний новий транспортний протокол, який дозволяв би контролювати черговість і затримку при доставці пакетів. Такий протокол зараз проходить стандартизацію. Він називається "Datagram Congestion Control Protocol (DCCP)" .

Однією з цілей DCCP було максимальне полегшення для UDP додатків переходу на DCCP, коли він буде впроваджений. Щоб полегшити це, DCCP був спроектований з мінімальною надмірністю, як з точки зору розміру заголовка пакету, так і з позиції завантаження ЦПУ машин партнерів. У DCCP була включена мінімальна функціональність, при збереженні можливості включення нових функцій, таких як FEC (forward error correction), псевдо надійність та множинні потоки, які можуть бути додані поверх DCCP, якщо буде потрібно.

Можливі різні механізми управління перевантаженням для різних додатків. Наприклад, ігри реального часу можуть вимагати швидкого використання усієї доступної смуги пропускання, тоді як потокове середовище може використовувати компроміс між швидкістю відгуку і стабільною, менш імпульсивною передачею. Різка зміна потоку може викликати неприйнятні збої, такі як відчутні паузи або клацання). Протокол DCCP, таким чином, пропонує додатку вибір одного з декількох механізмів управління перевантаженням. Однією з альтернатив є TCP-

подібне управління перевантаженням, скорочення удвічі вікна перевантаження у відповідь на втрату пакету. Додатки, що використовують механізм управління перевантаженням, швидко реагуватимуть на зміни доступної смуги, але повинні витримувати різку зміну вікна перевантаження, що типово для TCP. Другу альтернативу представляє механізм управління швидкістю передачі, дружній по відношенню TCP (TFRC) . Це алгоритм управління перевантаженням, що базується на рівнянні, мінімізує різкі зміни швидкості передачі.

Протокол DCCP дозволяє також ненадійному трафіку без проблем використовувати технікові ECN. Ядро UDP API не може дозволити додаткам рахувати UDP пакети, адаптованими до ECN, оскільки API не може гарантувати, що додаток здатний коректно детектувати або реагувати на перевантаження. Ядро DCCP API позбавлене такого недоліку, оскільки має вбудований механізм управління перевантаженням. Протокол DCCP має схожий з TCP алгоритм пригнічення перевантажень. Відмінність DCCP від TCP в тому, що в DCCP немає повторної передачі пакету. Таким чином, новий протокол є модифікацією протоколу UDP, і в перспективі використовуватиметься для передачі даних реального часу [15].

У експерименті на маршрутизаторі ми можемо настроювати WRED. Різниця в тому, що WRED дозволяє запускати на маршрутизаторі декілька незалежних копій RED. Якщо для декількох потоків задані квоти від загальної смуги пропускання, то для кожного з потоків можна конфігурувати свій набір параметрів RED. Тому експеримент зводиться до стандартної схеми RED, коли через пристрій проходить один потік із заданою на вході і на виході смугою.

3. ЗАБЕЗПЕЧЕННЯ QOS ЗА ДОПОМОГОЮ АЛГОРИТМІВ АКТИВНОГО УПРАВЛІННЯ ЧЕРГАМИ В СЕРЕДОВИЩІ NS-2

3.1. Моделювання мережевого каналу в середовищі NS-2

У цьому розділі описані результати застосування програми NS-2 для моделювання тестового каналу. Проблема оптимізації настроювальних параметрів WRED витікає з необхідності забезпечити потокам даних необхідне значення затримки при передачі пакетів і необхідний відсоток втрат пакетів. У сучасному Інтернеті перевантаження каналів являється звичайним явищем. Але, не дивлячись на те, що більшість з нас працює через переобтяжені канали, ми все ж можемо продовжувати працювати в мережі. Залежно від тривалості перевантаження, маршрутизатор може поміщати пакети в буфер відправки, або відкидати їх. При використанні механізму пригнічення перевантажень WRED, процес буферизації пакетів тягне коливання затримки доставки пакету. Також відбуваються додаткові втрати пакетів, викликані коливанням середньозваженої довжини черги, яка розраховується в процесі роботи WRED [16].

Мета моделювання тестового каналу полягала в пошуку для працюючого каналу його аналога, написаного із застосуванням програми моделювання NS-2. Щоб не перенавантажувати зайвим трафіком вже працюючий канал, можна побудувати програмну модель тестового каналу. Відкалібрувати модель каналу так, щоб при однакових рівнях завантаження тестового каналу, модель давала значення втрат і затримок пакетів близькі до експериментальних значень. Коли модель тестового каналу побудована і відкалібрована, можна проводити виміри по оптимізації настроювальних параметрів WRED на моделі, не вносячи перешкод до роботи мережевого каналу.

При дослідженні можливості забезпечення якості ми провели ряд модельних розрахунків. У перерахованих нижче пунктах знаходиться схема модельних досліджень. Мета цього моделювання-розробка додаткових засобів виміру і аналі-

зу параметрів QoS. У роботі досліджувалася можливість застосування модельних методів оцінки параметрів якості обслуговування.

1) Проведені дослідження алгоритмів RIO-C і WRED для різних моделей призначення порогів. Було також досліджено вплив алгоритмів-диспетчерів на якість обслуговування, що надавалася. Встановлені оптимальні межі роботи алгоритмів WRED і RIO, при яких використання цих алгоритмів являється найбільш ефективним.

2) Був конфігурований пакет NS-2. До вбудованих в NS-2 функціям були додані утиліти для розрахунку і відображення значень середньозважених черг і вірогідності відкидання пакетів. Це було необхідно для вивчення і інтерпретації поведінки нашої тестової мережі.

3) Отримані залежності довжин черг від часу, довжин середньозважених черг від часу і вірогідності відкидання пакетів від часу. Також виконаний аналіз цих залежностей і вироблені рекомендації по установці параметрів алгоритмів WRED і RIO для досягнення оптимальної роботи (для моделі).

4) Було виконано підбір параметрів моделі віртуальної мережі, так, щоб вона адекватно відбивала роботу тестової мережі. Було виконано повномасштабне моделювання для широкого набору параметрів, були отримані умови, при яких модель і експеримент на реальній мережі поводяться так само.

У разі одночасного проходження великого числа потоків через мережу сервіс провайдера виникає проблема: як при обмежених ресурсах оптимізувати затримку доставки, часовий розкид і втрати. Такі вимоги істотні, наприклад, у разі IP-телефонії, відеоконференцій, управління устаткуванням в реальному масштабі часу.

Для вирішення цих проблем існують технології IntServ і DiffServ. IntServ реалізується у рамках протоколу RSVP і не припускає формування декількох субпотоків з різними рівнями послуг. Це дослідження присвячено DiffServ. Передбачається, що усі налаштування алгоритмів працюють в межах одного DiffServ домена. (DiffServ домен - це набір прилеглих вузлів, конфігурованих з однаковою політикою PHB (Per Hop Behavior)).

Було поставлено завдання знаходження оптимальних параметрів WRED, які у разі перевантаження забезпечували б призначеному для користувача потоку гарантовану смугу і затримку.

При дослідженні формувалися три стаціонарні потоки пакетів (див. рис. 3.1), які мали DSCP мітки. На підставі даних про проходження цих потоків через DiffServ домен (затримки, втрати, дисперсія затримок) вибираються оптимальні значення для налаштування конфігурації маршрутизатора, при яких у разі перевантаження потік користувача отримає передбачувану смугу і затримку.

У дослідженнях використовувався транспортний протокол UDP. Як параметри, що характеризують потік, застосовувалися:

Вірогідність втрати пакетів;

Час доставки пакету (RTT/2);

Смуга пропускання, доступна потоку.

Потоки позначалися мітками DSCP рівними 10, 12 та 13.

Віртуальний відправник посилав одержувачеві три UDP потоку пакетів із швидкістю 920 кбіт/с кожен (що створює загальний потік в 2,76 Мбіт/с). Швидкість відправки пакетів потоків була постійною протягом усього сеансу передачі. Пакети мали довжину 1400 байт, тривалість посилки потоків (моделювання) складала 30 секунд. Віртуальний канал між "Маршрутизатор 1" і "Маршрутизатор 2" (аналог ATM каналу) мав пропускну спроможність 2 Мбіт/с. Сумарний потік даних відправника перевищував пропускну спроможність каналу. Експеримент тривав 40 секунд. Щоб вивчити, як здійснювалася конкуренція між різними потоками в каналі, ми завантажували канал так, щоб нестача смуги була менше 30% пропускної спроможності. З кожною міткою DSCP була зв'язана певна черга. Введемо позначення:

- DSCP 10 - відповідала черга 1;
- DSCP 12-2;
- DSCP 14-2.

Максимальні довжини середньозважених черг дорівнювали 40. Максимальний розмір пакетного буфера для кожної з черг дорівнював 40 пакетам. Підкрес-

лимо, що середньозважена черга і пакетний буфер - це різні об'єкти, нижче ми це проілюструємо.

При моделюванні використовувався алгоритм CBWFQ, який дозволяє виділити кожній черзі долю смуги вихідного інтерфейсу віртуального маршрутизатора [17].

Для черги 1 надавалося 41%, для черги 2-34%, а для черги 3-25% смуги пропускання каналу. При постановці в чергу працював алгоритм WRED (табл. 3.1).

T_1 - нижній поріг відкидання пакетів;

T_2 - верхній поріг відкидання пакетів;

p_c - максимальна вірогідність відкидання = 1.0/c (c > 1 - параметр, використаний для завдання максимальної вірогідності відкидання пакетів);

w_q - параметр усереднювання при обчисленні середньозваженого значення довжини черги.

У маршрутизаторі існує дві причини відкидання пакетів:

- Idrops - пакети, що відкидаються через перевищення потоком квоти;
- edrops - відбувається через переповнювання буфера черги, edrops-пакети, відкинуті WRED (при постановці в чергу).

У сучасних маршрутизаторах часто застосовується алгоритм RIO, який використовує той же механізм, що і RED, але для його конфігурації служать два набори параметрів, один для пакетів, що належать профілю (профіль це умова приналежності до потоку, наприклад, значення DSCP мітки або певне значення полів IP заголовка в пакеті), інший-для тих, що не належать профілю. При вступі нового пакету маршрутизатор перевіряє, відповідає пакет профілю або ні. Якщо пакет відповідає профілю, маршрутизатор обчислює avg_in (середньозважена довжина черги для вхідних пакетів), якщо пакет непрофільний, то маршрутизатор обчислює avg_total, середньозважена суму довжин черг профільного і непрофільного потоків. Вірогідність відкидання профільного пакету залежить від avg_in, а вірогідність відкидання непрофільного пакету залежить від avgtotal.

При моделюванні використовувалися наступні значення квот для черг (табл. 3.2).

Саме ці квоти в основному визначають рівень втрат Idrops. Пояснимо, чому квоти були вибрані саме так, а не інакше. Ми хочемо, щоб із зменшенням значення DSCP, потік отримував кращу якість обслуговування, тобто менше втрачався і менше затримувався в буфері. Кожен з потоків, що відправляються в мережу, мав однакову смугу, позначимо її "A". Тоді сумарний потік даних від трьох потоків - "3A". Нехай вихідна смуга каналу рівна "B".

У разі, якщо B<A, то про хорошу якість говорити не доводиться. Оскільки в цьому випадку, ні про яку якість говорити не можна: найважливішому потоку не вистачає смуги, а іншим двом потоком зовсім не залишиться смуга у вихідному каналі. Їх можна було б взагалі не передавати, все одно вони будуть відкинуті.

Якщо B>3A те усі потоки будуть передані одержувачеві без втрат і затримок і алгоритми обслуговування черг не використовуватимуться маршрутизатором.

Інтерес представляє випадок, коли A<B<3A. Для потоку з DSCP 10 виділяється квота смуги свідомо достатня, щоб потік не втрачався. Інші два потоки втрачатимуться і для них спрацює процедура управління чергою.

На рис. 3.2 показані часові залежності смуги пропускання, виділеної під кожен потік. Ця залежність допоміжна і ілюструє лише те, що потоки отримали правильну долю смуги (див. табл. 3.2).

Алгоритми, які розпоряджаються долею пакетів при вилученні з черги, називаються диспетчерами (schedulers). В сумі рівноважні рівні пропускної спроможності (рис. 3.2) дають 2 Мбіт/с.

На рис. 3.2. приведені часові залежності затримки поширення пакету (RTT/2) для кожного з потоків. Для потоків з DSCP=12 і 14 спостерігається викид в області 7-17 с. Цей викид пов'язаний з буферизацією пакетів в черзі і роботою алгоритмів WRED і Tail Drops. Найбільш пріоритетний потік має DSCP=10, отже, його пакети проводять в черзі менше часу. Потокам з DSCP=12 і 14 не вистачає смуги пропускання, і вони вимушені довше перебувати в черзі.

Проблема оптимізації налаштувань параметрів алгоритму WRED пов'язана з допустимими вимогами додатка до граничних величин втрат і затримок пакетів. З вимог кожного конкретного застосування виходить відповідь на питання: значення якого з двох параметрів QoS повинен забезпечувати WRED в першу чергу.

Збільшення затримки потоків з DSCP=12 і 14 відбувається через роботу алгоритму WRED. Середньозважена довжина черги обмежена значенням 40 (T_2=40), а WRED починає відкидати пакети, коли середньозважена довжина черги перевищує поріг T_1 .

На рис. 3.4 [18] показані зміни довжин черг залежно від часу. При першому погляді на рис. відразу виникає питання: Чому досягнувши максимального порогу від 0 до 10 секунд, поточна довжина черги починає убувати і осцилювати біля нового порогу, відмінного від величини максимальної глибини черги? Відповідь полягає в правильному підборі параметра усереднювання w_q.

Для потоку з DSCP=14 в період часу з 0 до 3 с. буфер починає заповнюватися, затримка росте лінійно (див. рис. 3.4), далі поточне значення довжини черги стає рівним максимально допустимому значенню (середньозважена довжина черги ще не встигла досягти порогу відкидання) і пакети, що приходять, починають відкидатися при спробі поставити їх в чергу (Tail Drops). Етап Tail Drops триває з 3 до 13 с.

На цьому етапі ми спостерігаємо викид, причиною якого являється алгоритм CBWFQ (див. рис. 3.3). Потоки отримують різну смугу, і це сильно впливає на час перебування в черзі. Після 13-ої секунди розмір середньозваженої черги виходить на плато (див. рис. 3.4), і пакети починають відкидатися алгоритмом WRED.

Розглянемо еволюцію довжини черги для DSCP=14 (див. рис. 3.4). Для пояснення поведінки черги розглянемо зміну довжини середньозваженої черги залежно від часу для потоку з DSCP=14 (див. рис. 3.5). Горизонтальні лінії означають пороги при налаштуванні алгоритму WRED: для DSCP 10 T_1 = 35 T_2 = 40, для DSCP 12 T_1 = 30, T_2 =40, для DSCP 14 T_1= 20, T_2 =40.

У момент 8 секунд від початку експерименту середньозважена довжина черги з DSCP=14 досягла першого порогу - 20 пакетів, у момент часу 14 секунд середньозважена довжина черги вийшла на плато. По вичисленому середньозваженому значенню довжини черги визначається вірогідність відкидання пакету. Отже, вірогідність відкидання пакету теж вийшла на плато, і така, що підтримує поточну довжину черги нижче за максимальний поріг. У цей же момент часу - 13 секунд спостерігається найбільша затримка (див. рис. 3.3 для потоку з DSCP=14). На плато середньозважена довжина черги знаходиться між мінімальним і максимальним порогом.

Вибір параметрів WFQ і WRED сильно позначаються на поведінці втрат і затримок. На рис. 3.6 приведена часова залежність загальних втрат, які складаються з edrops і Idrops. Слід мати на увазі, що чим менше значення p_c (менша доля пакетів відкидається алгоритмом WRED), тим більше пакетів буде відкинуто на виході буфера через обмеженості квоти [19].

Моделювання показало, що при некоректному виборі параметрів, наприклад чинника усереднювання, можна отримати досить широкі варіації довжини черги, які у свою чергу можуть в рази збільшити дисперсію RTT. Показане моделювання продемонструвало наслідки неоптимального підбору параметрів і наші можливості спостереження за цими параметрами в середовищі моделювання NS-2.

Нижче приведені результати дослідження залежності усередненої довжини черги від параметра w_q для трьох потоків. Параметри потоків представлені нижче в табл. 3.3. Кожен з вхідних потоків мав інтенсивність 0.92 Мбіт/с, а смуга пропускання вихідного потоку дорівнювала 1,9 Мбіт/с.

3.2. Дослідження залежності усередненої довжини черги від параметра p_c для трьох потоків

У цьому експерименті досліджується вплив параметра p_c на поведінку алгоритму WRED. Спочатку приводяться графіки для значення p_c рівного 0.9, далі приводяться графіки для значення p_c рівного 0.1.

Таблиця 3.4.

Залежність усередненої довжини черги від параметра p_c

DSCP	T_1	T_2	p_c	w_q
10	20	80	0.9	0.002
12	50	80	0.9	0.002
14	70	80	0.9	0.002

З порівняння рис. 3.13 і 3.16 можна зробити висновок, що збільшення значення p_c впливає на час виходу системи в стан рівноваги, але цей вплив дуже малий.

3.3. Аналіз осциляції довжин черг

Дослідження еволюції довжин черг в процесі передачі даних по каналах з гарантією якості обслуговування являється вкрай важливим. Від довжини черги і її флуктуації (чи осциляції) залежить значення RTT і його дисперсія. Ці параметри визначають якість передачі мультимедійних даних. Крім того, від довжини черги (від міри заповнення буфера) залежить і вірогідність втрати пакетів.

Метод усереднювання довжини черги пакетів WRED (див. формулу (3.1)) з неминучістю повинен породжувати осциляційні перехідні процеси, що ми і спостерігаємо. Враховуючи той факт, що RTT визначається в основному часом перебування в чергах (окрім супутникових каналів), такі осциляції можуть призводити

до збільшення дисперсії значення RTT. Важливо визначити, від яких параметрів і яким чином залежить рівноважне значення довжини черги, амплітуда осциляції і час їх загасання. Для оцінки цих характеристик нами були проведені модельні і експериментальні дослідження, а також був зроблений висновок аналітичної залежності середньозваженої довжини черги від часового кроку. Отримані формули і розрахунки дозволяють оптимізувати параметри віртуального каналу з метою максимально можливого демпфування осциляції черги. Отримані формули і написані програми дозволяють швидко оцінити якість віртуального каналу, не завантажуючи його допоміжним трафіком. Нижче представлений аналітичний опис моделі. У цій моделі передбачалося, що пакети поступають з постійною швидкість λ, а швидкість видалення пакетів з черги рівна μ. (з урахуванням квоти, що задається для цього потоку) [20].

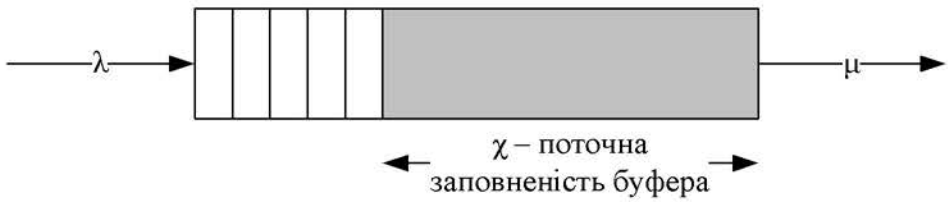

Рисунок 3.19. Ілюстрація до механізму WRED

Розглянемо модель, що ілюструє швидкість заповнення буфера для випадку, коли пакети на вхід буфера поступають з постійною бітовою швидкістю. У цій моделі розглядається рівномірна часова шкала.

Нехай поточне заповнення буфера рівне χ, а відповідна цьому заповненню середньозважена довжина черги рівна F(x)=len, тоді рішення про відкидання пакету прийматиметься за наступною схемою:

Вірогідність відкидання пакету міняється лінійно згідно із законом.

$P_{loss}(len)=0$, для $len \le T_1$

$$P_{loss}(len)=\frac{len - T_1}{T_2 - T_1}, \text{ для } T_1 \le len \le T_2$$

$P_{loss}(len)=1$, для $T_2 \le len$

$[1-P_{loss}(len)]$ - вірогідність того, що пакет пройде і не загубиться.

Рішення для поточної довжини черги в інтервалі від 0 до T_1*, де T_1* - наповненість буфера, яка досягається в мить, коли середньозважена довжина черги стає рівною T_1 - нижньому порогу.

Параметр усереднювання $w_q \subset [0,1]$:

$$\begin{cases} Q_{av}^{i+1} = (1 - w_q) Q_{av}^{i} + Q^{i} w_q \\ Q^{i+1} = (\lambda - \mu) \Delta t \cdot (i+1) \\ Q^{K} = T_1^{*} \\ Q^{0} = 0 \\ Q_{av}^{0} = 0 \end{cases}. \tag{3.1}$$

Прослідкуємо по кроках еволюцію системи. З граничних умов ми отримаємо часовий крок сітки для фіксованого числа кроків K.

$$\begin{cases} \mathbf{Q^{K} = T_1^{*}} \\ \Downarrow \\ \mathbf{Q^{i+1} = (\lambda - \mu) \Delta t \cdot K} \\ \mathbf{\Delta t = \dfrac{T_1^{*}}{(\lambda - \mu) \Delta t \cdot K}} \\ \mathbf{i = 0} \\ \mathbf{Q_{av}^{1} = (1 - w_q) Q_{av}^{0} (= 0) + Q^{0} w_q (= 0) = 0} \\ \mathbf{Q^{1} = (\lambda - \mu) \Delta t \cdot 1 = (\lambda - \mu) \Delta t} \\ \mathbf{i = 1} \\ \mathbf{Q_{av}^{2} = (1 - w_q) Q_{av}^{1} (= 0) + Q^{1} w_q = Q^{1} \cdot w_q} \\ \mathbf{Q^{2} = (\lambda - \mu) \Delta t \cdot 2} \\ \mathbf{i = 2} \\ \mathbf{Q_{av}^{3} = (1 - w_q) Q_{av}^{2} + Q^{2} w_q = (1 - w_q) Q^{1} \cdot w_q + Q^{2} \cdot w_q = Q^{1} \cdot w_q (3 - w_q)} \\ \mathbf{Q^{3} = Q^{1} \cdot 3} \\ \end{cases} \tag{3.2}$$

З приведених формул видно, що функція залежності середньозваженої довжини черги від номера часового кроку - суворо монотонна, причому її вид сильно залежить від параметра w_q.

Далі приведемо графіки, отримані для випадків з різними значеннями параметра w_q. Графіки були побудовані за допомогою середовища моделювання NS-2.

Для маленького розміру буфера (якщо розмір буфера і поріг T_1* приблизно рівні), треба ставити великі значення w_q (w_q=0.9 на рис. 3.20: по осі Y відкладено число пакетів, а по X - число кроків), оскільки середньозважена довжина черги майже лінійно залежить від поточної кількості пакетів в черзі.

Якщо пакетний буфер великий в порівнянні з нижнім порогом T_1*, то можна використовувати маленьке значення (wq=0.1 на рис. 3.21), Але ми бачимо, що довжина середньозваженої черги у момент досягнення довжини поточної черги рівня T_1* дорівнює 2.

Значить алгоритм WRED почне відкидати пакети приблизно через $10 \cdot t_{T1}$, де t_{T1} - час заповнення поточного значення черги від 0 до T_1*.

Далі досліджуємо осциляції середньозваженої довжини черги на плато. У момент виходу на плато, середньозважена довжина черги $\overline{Q}_0 = T_2$. Під "плато" позначається область значень середньозваженої довжини черги, де вона слабо міняється.

Коли $\overline{Q}_0 = T_2$, WRED відкидає усі пакети, що поступають на вхід черги, черга зменшується за рахунок відправки в мережу пакетів, що вже буферизують в ній.

Закон зміни довжини Q(t) $\overline{Q}(t+dt) = (1-w_q)\overline{Q}(t) + w_q \overline{Q}(t)$, що еквівалентно $\overline{Q}^{i+1} = (1-w_q)\overline{Q}^i + w_q \overline{Q}^i$.

За одиницю часу 1/μ - час відправки в мережу 1 пакету поточна довжина буфера M зменшується на 1.

Теорема 1. Для будь-яких n>0 справедлива формула

$$\overline{Q}^{n} = \left(1 - w_q\right)^n \overline{Q}^{0} + w_q \sum_{k=1}^{n} \left(Q^0 - \left(k-1\right)\right)\left(1 - w_q\right)^{n-k}, \tag{3.3}$$

де $\overline{Q} = \overline{Q}^{0}$ - початкове значення середньозваженої довжини черги; $Q=Q^0$- початкове значення поточної довжини черги.

Доказ.

Поточна довжина черги на плато змінюється згідно із законом $Q^n = Q^0$-n (з властивості спустошення черги).

Середньозважена довжина черги міняється згідно із законом, описаному формулою

$$\overline{Q}^{n} = \left(1 - w_q\right)\overline{Q}^{n-1} + w_q \overline{Q}^{n-1}. \tag{3.4}$$

Нехай формула 3.3 справедлива для $n=n^*$-1. Покажемо, що вона буде справедлива і для $n=n^*$. З формули (3.4) [21]

$$\begin{aligned}
&\overline{Q}^{n^*} = \left(1 - w_q\right)\overline{Q}^{n^*-1} + w_q \overline{Q}^{n^*-1} = \\
&= \left(1 - w_q\right)\left[\left(1 - w_q\right)^{n^*-1}\overline{Q}^{0} + w_q \sum_{k=1}^{n^*-1}\left(Q^0 - \left(k-1\right)\right)\left(1 - w_q\right)^{n^*-k}\right] + w_q\left(Q^0 - n^*\right) = \\
&= \left(1 - w_q\right)^{n^*}\overline{Q}^{0} + w_q \sum_{k=1}^{n^*}\left(Q^0 - \left(k-1\right)\right)\left(1 - w_q\right)^{n^*-k}.
\end{aligned} \tag{3.5}$$

Оскільки формула (3.3) справедлива для n=1 і ми довели, що з її справедливості для довільного n*-1 слідує її справедливість для n*, формула (3.3) справедлива для будь-кого n. Теорема доведена.

На рис. 3.22 представлені залежності варіації довжин черг для інформаційних потоків з мітками DSCP=10, 12 і 13.

Червоним кольором на рисунку позначений відрізок, де працює формула (3.3). Виведемо залежність для ділянки, позначеної синім кольором (формула 3.8). Коли середньозважена довжина черги стає нижче T_2, WRED починає пропускати

в буфер нові пакети, що приходять від відправника. За момент відправки в мережу 1 пакету 1/μ в буфер поступить-пакетів λ/μ. З вірогідністю

$$P_c^i = \frac{\overline{Q}^{i-1} - T_1}{T_2 - T_1} p_c \qquad (3.6)$$

пакет на вході в буфер буде відкинутий (де p_c - максимальна вірогідність відкидання WRED, після того, як, P_c^i стане рівним p_c, пакети відкидаються з вірогідністю 1). Середнє число пакетів, що поступили в буфер, в і момент часу:

$$\Sigma = \frac{\lambda}{\mu}\left(1 - \frac{\overline{Q}^{i-1} - T_1}{T_2 - T_1} p_c\right). \qquad (3.7)$$

Отримаємо аналітичний вираз для залежності середньозваженої довжини черги від часового кроку (виділена ділянка залежності на рис. 3.22).

Теорема 2. Для будь-яких n>0 справедлива формула

$$\overline{Q}^n = \left(1 - w_q\right)^n \overline{Q}^0 + w_q \sum_{k=1}^{n} \left(Q^0 - (k-1)\right)\left(1 - w_q\right)^{n-k} + w_q \frac{\lambda}{\mu} \sum_{k=1}^{p=n-1} \left(1 - w_q\right)^{p-k} \sum_{i=1}^{k} P_c^i, \qquad (3.8)$$

де $\overline{Q} = \overline{Q}^0$ - початкове значення середньозваженої довжини черги; $Q=Q^0$- початкове значення поточної довжини черги.

Доказ.

Коли середньозважена довжина черги лежить між нижнім і верхнім порогами WRED, поточна довжина черги змінюється згідно із законом $Q^n = \left(Q^0 - n\right) + \frac{\lambda}{\mu} \sum_{i=1}^{n-1} P_c^i$ (з властивості спустошення і заповнення черги). Середньозважена довжина черги міняється згідно із законом описаному формулою (3.4).

Нехай формула 3.8 справедлива для випадку n^*-1. Покажемо, що вона буде справедлива і для випадку n^*. З формули (3.4)

$$
\begin{aligned}
&\overline{Q}^{n^*} = \left(1 - w_q\right)\overline{Q}^{n^*-1} + w_q \overline{Q}^{n^*-1} = \\
&= \left(1 - w_q\right)\left[\left(1 - w_q\right)^{n^*-1}\overline{Q}^{0} + w_q \sum_{k=1}^{n^*-1}\left(Q^0 - (k-1)\right)\left(1 - w_q\right)^{n^*-k} + \right. \\
&\left. + w_q \frac{\lambda}{\mu} \sum_{k=1}^{p=n-2}\left(1 - w_q\right)^{p-k}\sum_{i=1}^{k} P_c^i \right] + w_q\left(Q^0 - n^* + w_q \frac{\lambda}{\mu}\sum_{k=1}^{n^*-1} P_c^k\right) = \\
&= \left(1 - w_q\right)^{n^*}\overline{Q}^{0} + w_q \sum_{k=1}^{n^*}\left(Q^0 - (k-1)\right)\left(1 - w_q\right)^{n^*-k} + w_q \frac{\lambda}{\mu}\sum_{k=1}^{p=n-1}\left(1 - w_q\right)^{p-k}\sum_{i=1}^{k} P_c^i .
\end{aligned}
\tag{3.9}
$$

Оскільки формула (3.8) справедлива для n=1 і ми довели, що з її справедливості для довільного n^*-1 слідує її справедливість для n^*, формула (3.8) справедлива для будь-кого n. Теорема доведена.

ВИСНОВКИ

Сукупність проведених досліджень в магістерській кваліфікаційній роботі складають вирішення завдань з забезпечення гарантованої якості обслуговування в мережі Інтернет.

По результатах роботи зробимо наступні висновки:

1. Поняття якості обслуговування виникає, коли для роботи додатка вимагається гарантоване виділення ресурсів мережевого каналу. Потік даних від додатка виділяється на основі набору ознак із загального потоку даних. До виділеного потоку повинні застосовуватися правила, які забезпечать йому необхідні умови проходження через канал сервіс провайдера і локальну мережу.

2. Кінцевою метою регулювання трафіку і запобігання перевантаженням є встановлення відповідності між темпом передачі і можливостями прийому. Причиною перевантаження може бути не лише обмеженість розміру буфера, але і недостатня пропускна спроможність якоїсь ділянки каналу. З урахуванням цієї обставини кожен відправник формує два вікна: вікно одержувача і вікно перевантаження.

3. Основними характеристиками продуктивності мережевого з'єднання є смуга пропускання, затримка при передачі пакетів (RTT), дисперсія RTT, рівень втрат пакетів.

4. Для боротьби з накопиченням черг застосовуються наступні методи:

- Черги FIFO;
- Пріоритетне обслуговування черг (PQ);
- Звичайне обслуговування черг (CQ);
- Справедливі черги (WFQ);
- Черги з малою затримкою (LLQ);
- Інтегрована служба IntServ і диференційована служба DifServ.

5. У випадку неможливості уникнення накопиченням черг застосовуються наступні методи:

- Алгоритм "діряве відро";
- Алгоритм "маркерне відро";
- Алгоритми RED і WRED.

6. Алгоритм WRED на сьогоднішній день є найбільш досконалим алгоритмом для підтримання працездатності мережі в годину найбільшого навантаження та доставки високопріорітетних пакетів. Хоча втрати низькопріорітетних пакетів все одно присутні.

7. За допомогою програми NS-2 проведені дослідження по ефективності роботи алгоритму WRED. Використання моделі для визначення настроювальних параметрів WRED, які забезпечують потоку даних необхідне значення затримки, дисперсії затримки, і відсоток втрачених пакетів, дозволяє відмовитися від проведення вимірів в реально працюючому мережевому каналі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. - 3-е изд., - С.Пб.: Питер, 2006. - 957 с.

2. Конахович Г. Ф., Чуприн В. М. Сети передачи пакетных данных. - К.: МК-Пресс, 2006. - 272 с.

3. Шринивас Вегешна Качество обслуживания в сетях IP. - М.: Вильямс, 2003. - 368 с.

4. Кучерявый Е. А. Управление трафиком и качество обслуживания в сети Интернет. - С.Пб.: Наука и техника, 2004. - 336 с.

5. Столлингс В. Современные компьютерные сети. - СПб.: Питер, 2003. -783 с.

6. K. Ramakrishnan, S.Floyd, D.Black, 'The Addition of Explicit Congestion Notification (ECN) to IP", IETF RFC 3168, September 2001

7. Vishal Misra, WeiBo Gong, Don Towsley, "Fluidbased Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", In Proceedings of ACM SIGCOMM 2000, pages 151-160, Aug. 2000.

23. M. Handly, S. Floyd, J. Padhye, and J. Windmer, "TCP friendly rate control (TFRC): protocol specification", RFC 3448, Jan. 2003

8. Yavatkar, R., Pendarakis, D. and R. Guerin, "A Framework for Policy-Based Admission Control", RFC-2753, January 2000.

9. ATM, http://book.itep.ru/4/43/atm_435.htm

10. Braden B., Clark D., Crowcroft J., Davie B., Deering S., Estrin D., Floyd S., Jacobson V., Minshall G., Partridge C, Peterson, L.,Ramakrishnan K., Shenker S., Wroclawski J., Zhang L., Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309,2001.

11. Erich Plasser, Thomas Ziegler, "An Analytical RED Function Design Guaranteeing Stable System Behavior"

12. Julio Orozco, David Ros, Jos_e Incera, Rodolfo Cartas "A Simulation Study of the Adaptive RIO (A-RIO) Queue Management Algorithm", 2004

13. Plasser E., Ziegler T., On the Non Linearity of the RED Drop Function, ICCC 2002, Mumbai, India, August 2002

14. S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Internetworking, V7, N4. August 2001.

15. The NS-2 network simulator (ver.2) LBL, http.7/www-mash.CS.Berkeley.edu/ns

16. S. Floyd, "Discussion on setting RED parameters", http://www.aciri.org/floyd/red.html, Nov. 2005

17. http://www.cisco.com/warp/public/121/atm_vbrshape.shtml

18. RFC 2001 - TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms

19. ANSI/IEEE Std 802.1D. Standard for local and metropolitan area networks: Media access control (MAC) bridges, 2005

20. Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DSField) in the IPv4 and ff v6 Headers", RFC 2474, December 2007.

21. Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 2008.

22. Ratul Mahajan, Sally Floyd, and David Wetherall, Controlling High-Bandwidth Flows at the Congested Router, 9th International Conference on Network Protocols (ICNP), November 2001

23. Ratul Mahajan and Sally Floyd, Controlling High Bandwidth Flows at the Congested Router ICSI Tech Report TR-01-001, April 2001

24. E. Kohler, M. Handley, S. Floyd, and J. Padhye. Datagram Congestion Control Protocol, 2003.

25. M. Handley, S. Floyd, J. Padhye, and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification, RFC 3448, Proposed Standard, January 2003.

26. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 2007.

27. Ziegler T,, Fdida S., Brandauer G, A quantitative Model of RED with TCP Traffic, IEEE/ACM IWQoS, May 2008, Karlsruhe

28. Firoiu V., Borden M., A Study of Active Queue Management for Congestion Control, Infocom 2005

29. Feng W., Kandlur D., Saha D., Shin K., A Self-Configuring RED Gateway, Infocom 2003

30. Ziegler T., On Averaging for Active Queue Management Congestion Avoidance, IEEE ISCC-2002

31. D. Clark, W. Fang, Explicit Allocation of Best-Effort Packet Delivery Service,IEEE/ACM Transactions on Networking 6 (4) (2004) 362-373

32. Understanding Buffer Misses and Failures, http://cisco.com/en /US/products/hw/modules/ps2643/products_tech_note09186a0080 093fc5.shtml

33. Bernet, Y. et al, "A Framework For Integrated Services Operation Over Diffserv Networks". Internet Engineering Task Force, Request for Comments (RFC) 2998, November 2000.