

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кваліфікаційна наукова
праця на правах рукопису

ПОПОВСЬКА КАТЕРИНА ОЛЕГІВНА

УДК 621.391

ДИСЕРТАЦІЯ
МЕТОДИ ОПТИМІЗАЦІЇ ПРОЦЕСУ ФРАГМЕНТАЦІЇ КОНТЕНТУ В
ПРИНГОВИХ ФАЙЛООБМІННИХ МЕРЕЖАХ

Спеціальність: 05.12.02 – телекомунікаційні системи та мережі
05 «Технічні науки»

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Поповська К.О.

Науковий керівник: Москалець Микола Вадимович, кандидат технічних наук,
доцент

Ідентичність всіх примірників дисертації засвідчую:
Вчений секретар спеціалізованої вченої ради

/О.Б. Ткачова/

Харків – 2017

АНОТАЦІЯ

Поповська К. О. Методи оптимізації процесу фрагментації контенту в пірингових файлообмінних мережах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.12.02 «Телекомунікаційні системи та мережі». – Харківський національний університет радіоелектроніки, – Харків, 2017.

Дисертаційна робота присвячена вирішенню актуальної наукової задачі, що полягає в оптимізації процесу фрагментації запрошених файлів децентралізованої пірингової мережі P2P і аналізу якості процесу формування файлів при різному режимі їх реплікації і в умовах стаціонарності і нестаціонарності роботи мережі. Оптимізація забезпечує покращення якості надання послуг при передачі медіаконтенту. Для вирішення задачі розроблена дискретно-неперервна математична модель фрагментації. З використанням мультиагентної моделі вирішено задачу оптимізації відбору кращого з m - запропонованих фрагментів, задачу оптимізації процесу фрагментації з використанням динамічного програмування, і в дещо скороченій формі з використанням задачі лінійного програмування. Представлено приклади розв'язання даних оптимізаційних задач, проведена порівняльна характеристика методів при одних і тих самих умовах. Показано, що для конкретизації файлів метод лінійного програмування дає кращий результат при паралельному скачуванні декількох фрагментів. Отримана статистика процесу скачування фрагментів для оптимізованої ситуації і без неї. Зроблено висновок, що час скачування в оптимізованих умовах майже в три рази скорочується. Одночасно з цим зменшується дисперсія часу скачування, що дозволяє мінімізувати втрати фрагментів при їх реплікації. Проведено аналіз різних методів планування фрагментів і показано, що втрати фрагментів ростуть пропорційно бітрейту трафіку. Разом із тим, ці втрати можуть бути мінімізовані з урахуванням медіаобізнаності мережі при обліку структури відеоконтенту.

Запропоновані рішення щодо оптимізації процесу завантаження забезпечують мінімізацію часу скачування і втрату фрагментів в умовах великих навантажень.

Ключові слова: пірингові, фрагментація, скачування, реплікація, контент, файлообмінні мережі.

ABSTRACT

Popovska K.O. Optimization methods of content fragmentation process in peer-to-peer file-sharing networks. – Qualification research work as a manuscript.

The thesis for a candidate degree (Ph.D.) in specialty 05.12.02 – telecommunication systems and networks. – Kharkiv National University of Radio Electronics. – Kharkiv, 2017.

The dissertation is devoted to the solution of the actual scientific problem, which consists in the development of a discrete-continuous mathematical model for the process of fragmentation of downloadable files in the decentralized P2P peer-to-peer network and analysis of the quality of the process of file generation under different mode of replication both under stationary conditions and non-stationary network operation. Using the multiagent model, the problems of optimizing the selection of the best of the proposed fragments are solved, the task of optimizing the fragmentation process using dynamic programming is solved, and in a somewhat reduced form using the linear programming problem. Examples of the solution of these optimization problems are presented, the comparative characteristics of the methods under the same conditions are made. It is shown that with the concretization of files, the linear programming method gives the best result when several fragments are downloaded in parallel. The statistics of the process of downloading fragments for an optimized situation and without it has been obtained. It is shown that the download time in optimized conditions is almost three times reduced. At the same time, the dispersion of the download time is reduced, which allows to minimize the loss of fragments during their replication. The analysis of various methods of fragment planning has been carried out and it is shown that the loss of fragments grows in proportion to the bit rate of

traffic. However, these losses can be minimized taking into account the media awareness of the network when taking into account the structure of video content. The proposed solutions for optimizing the download process ensure the minimization of the download time and the loss of fragments under conditions of high loads.

Keywords: peer-to-peer, fragmentation, downloading, replication, content.

Список публікацій здобувача:

1. Теплицкая С.Н. Анализ качества оценки случайных величин / С.Н. Теплицкая, Е.О. Поповская, Я.Т. Хусейн // Радиотехника: Всеукр. межвед. науч.- техн. сб. – 2010. – Вып. 163. – С. 204-208.
2. Saburova S.A. Prospects for service platform pre – IMS / S.A. Saburova, E.V. Bondar, E.O. Popovskaia // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2010. – Вып. 163. – С.13 -19.
3. Мельникова Л.И. Управление перегрузкой сетевых ресурсов с использованием процедуры Калмана-Бьюси / Л.И. Мельникова, С.Н. Горяева, Е.О. Поповская // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2009. – Вып. 159. – С. 35-41.
4. Поповская Е.О. Анализ качества двумерного рекурсивного фильтра / Е.О. Поповская, Б.С. Тур, Я.Т. Хуссейн // Восточно-Европейский журнал передовых технологий. – 2012. – Том 1. – № 9(55). – С. 26-28.
5. Saburova S.A. Development of services in the IP multimedia subsystem / S.A. Saburova, E.O. Popovska, Y.T. Hussein // Радиотехника: Всеукр. межвед. науч.- техн. сб. 2009. – Вып. 159. — С. 229 - 233.
6. Москалец Н.В. Модели одноранговых пиринговых сетей для передачи потоков IPTV/ Н.В. Москалец, Е.О. Поповская // "Международный научно-исследовательский журнал". – 2014. – Ч. 2. – № 4(23). – г. Екатеринбург. – С. 46-50.
7. Ощепков М.Ю. Разработка математических моделей одноранговых сетей с передачей IPTV потоков / М.Ю. Ощепков, Е.О.Поповская // Труды Северо-Кавказского филиала Московского технического университета связи и

информатики. – Ростов-на-Дону: ПЦ "Университет" СКФ МТУСИ'2014. – Ч.1. – С. 304 - 307.

8. Ощепков М.Ю. Математическое моделирование пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // «Science and world» International scientific journal. – 2014. – № 3(7). –Vol.1. – С. 195 – 197.

9. Москалец Н.В. Оптимизация суммарного взвешенного времени обслуживания в пиринговой сети / Н.В. Москалец, Е.О.Поповская, К.А. Тарасов // Международный научно-исследовательский журнал. – 2016. – № 6-2(20). – С. 110-116.

10. Москалец Н.В. Функциональная модель управления в сетях пиринговой связи / Н.В. Москалец, Е.О.Поповская // Радиоэлектроника и информатика. – ХНУРЕ. – 2017. –№1. – С. 48-52.

11. Москалец Н.В. Фрагментация TV-контента в P2P-сетях с помощью процедуры динамического программирования / Н.В. Москалец, Е.О.Поповская // Зв`язок. – 2017. – №2(126). – С.14-17.

12. Поповская Е.О. Анализ модели трафика реального времени и данных / Сб. науч. тр. 4-го Международного радиоэлектронного форума "Прикладная радиоэлектроника" МРФ-2011– Х.: АНПРЭ, Украина, Харьков. – 2011. – С. 130–131.

13. Поповская Е.О. Математические модели одноранговых сетей // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – 2014. – Том 4. – С. 79 – 80.

14. Oshchepkov M. Ways for organization of IPTV services over peer-to-peer networks / M. Oshchepkov, K. Popovskaya // Modern problems of radio engineering, telecommunications and computer science, Proceedings of the XII-th International Conference TCSET' 2014, February 25- March 1, Slavske. – P. 568.

15. Способы организации услуги IPTV поверх пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // 10-я Международная молодежная научно-техническая конференция РТ-2014 «Современные проблемы радиотехники и телекоммуникаций». – Севастополь. –12-17 мая 2014. – С. 95.

16. Moskalets N. Mathematical scheduling models of IPTV peer to peer networks / N. Moskalets, E. Popovskaya // *Матеріали 3-ї Міжнародної наукової конференції «Інформація, комунікації, суспільство»*. – Славське. – 21-24 травня. – 2014. – С. 20-21.
17. Linnik E.V. Optimization of the total weighted holding time in a p2p network / E.V. Linnik, K.O. Popovskaya // *Материалы IV международной научно-практической конференции «Фундаментальные и прикладные науки сегодня»*. – 20-21 октября 2014 г. – North Charleston, USA, Том 3. – С. 160 -163.
18. Popovsky V.V. Mathematical model of P2PTV-network dynamics / V.V. Popovsky, K.O. Popovska // *2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2014)*, October 14-17, 2014. – Kharkiv, Ukraine. – С. 23 - 25.
19. Москалец Н.В. Методы управления потоком фрагментов в пиринговых сетях / Н.В. Москалец, Е.О. Поповская // *Збірник матеріалів Міжнародної науково-технічної конференції «Проблеми телекомунікацій»*. – Київ: НДІ телекомунікацій ІТС КПІ ім. Ігоря Сікорського, 2017. – С.1-3.
20. Kuzminich I. Minimization process of the total weighted holding time in a P2P network / I. Kuzminich, K. Popovska // *2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2016)*, October 4-6, 2016. – Kharkiv, Ukraine. – С. 154 – 155.
21. Popovsky V. Searching optimization method for requested media data in peer-to-peer network / V. Popovsky, K. Popovska // *2017 Fourth International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2017)*, October 10-13, 2017. – Kharkiv, Ukraine. – P.1- 4.

ЗМІСТ

Вступ.....	10
Розділ 1. Файлообмінні пірингові мережі.....	18
1.1. Мережі доставки контенту CDN (Content Delivery Network).....	18
1.2. Загальні відомості про однорангові файлообмінні мережі.....	23
1.2.1. Визначення та властивості P2P-систем.....	25
1.2.2. Правила файлового обміну в пірингових мережах.....	28
1.3. Типи організації файлообмінних мереж.....	30
1.3.1. Централізовані мережі.....	30
1.3.2. Децентралізовані мережі.....	31
1.3.3. Частково децентралізовані (гібридні) мережі.....	32
1.3.4. Особливості децентралізованих технологій.....	33
1.3.5. Безпека пірингових P2P-мереж.....	35
1.3.6. Побудова пірингових мереж.....	36
1.4. Области застосування файлообмінних P2P-мереж.....	39
1.5. Процес пошуку ресурсів в пірингових P2P-мережах.....	44
1.5.1. Централізована файлообмінна система Napster.....	44
1.5.2. Розподілена децентралізована система: Gnutella.....	46
1.5.3. Гібридна система: FastTrack/Kazaa.....	48
1.5.4. BitTorrent.....	50
1.6. Порівняльна характеристика процесів пошуку в неструктурованих і структурованих P2P-мереж.....	51
1.6.1. Порівняння і обговорення методів пошуку файлів.....	51
1.7. Висновки до першого розділу	53
Розділ 2. Аналіз архітектури пірингових P2P мереж.....	56
2.1. Склад і основні функції елементів P2P-мережі.....	56
2.1.1. Загальні відомості про архітектуру P2P-мереж.....	56
2.1.2. Порядок взаємодії елементів в P2P-мережі.....	58
2.2. Архітектура структурованої системи на основі кластерів.....	62
2.3. Узагальнена архітектура програмного забезпечення P2P-мережі.....	66

2.3.1. Модуль планувальника.....	66
2.3.2. Модуль оверлею.....	67
2.3.3. Контрольний модуль.....	68
2.3.4. Рівень передачі повідомлень.....	68
2.4. Методи формування і послідовності відеофрагментів.....	69
2.5. Структура і топологія оверлею.....	72
2.6. Структура мультимедійних файлів.....	74
2.6.1. Системи на основі фрагментів.....	74
2.6.2. Системи на основі смуг.....	76
2.7. Висновки до другого розділу.....	78
Розділ 3. Розробка математичних моделей пірингових мереж.....	81
3.1. Математична модель динаміки пірингової P2PTV-мережі.....	81
3.1.1. Структура алгоритму управління пірингових мереж.....	81
3.1.2. Математична модель динаміки P2PTV-мережі.....	83
3.1.3. Аналіз динаміки пірингової мережі.....	86
3.2. Оптимізація процесу відбору скачуваних файлів у пірингових мережах.....	90
3.3. Аналіз стану контенту на вузлах P2P-мереж.....	96
3.4. Висновки до третього розділу	100
Розділ 4. Методи управління дискретно - неперервною моделлю скачування фрагментів.....	102
4.1. Опис функціональної математичної моделі дискретно-безперервного процесу скачування фрагментів.....	107
4.2. Оптимізація процесу управління в пірингової мережі.....	109
4.3. Оптимізація управління потоком фрагментів.....	111
4.3.1. Математична модель потоку фрагментів.....	111
4.3.2. Приклад рішення задачі динамічного програмування.....	115
4.4. Висновки до четвертого розділу	119
Розділ 5. Оптимізація процесу скачування фрагментів на основі дискретно-неперервної моделі.....	121

5.1. Оптимізація сумарного зваженого часу обслуговування в піринговій мережі.....	121
5.2. Синтез процедури мінімізації сумарного зваженого часу обслуговування в піринговій мережі.....	126
5.2.1. Математична постановка і рішення задачі цілочисельного лінійного програмування.....	127
5.3. Аналіз процедури мінімізації сумарного зваженого часу обслуговування...	131
5.4. Висновки до п'ятого розділу.....	139
Розділ 6. Аналіз методів планування інформаційних сегментів для потокового пірингового відео.....	141
6.1. Моделі планування потокового відео.....	141
6.2. Аналіз якості відео з урахуванням особливостей планування відео-контенту.....	144
6.3. Висновки до шостого розділу.....	150
Висновки по роботі.....	151
Список використаних джерел.....	153
Додатки.....	164

ВСТУП

Актуальність теми. Піринговими (Peer-to-peer: P2P) називаються однорангові оверлейні мережі, призначені для обміну файлами практично будь-яких розмірів, аж до десятків гігабайт.

До P2P належать мережеві телекомунікаційні технології, що забезпечують можливість обміну мультимедійними, обчислювальними й іншими інформаційними ресурсами між декількома рівноправними користувачами, підключеними до Інтернету.

Для вирішення зазначених завдань створюється однорангове (peer-to-peer) обчислювальне середовище, яке дозволяє окремим елементам мережі взаємодіяти без допомоги серверів. Кожен користувач вносить свій вклад у вигляді файлів, дискового простору, процесорного часу.

Принцип «peer-to-peer» дозволяє створювати суттєві обчислювальні потужності, об'єднуючи безліч звичайних персональних комп'ютерів.

З появою в 1999 році протоколу обміну музичними файлами Napster і в 2001 BitTorrent, пірингові мережі набули популярності і впевнено утримують першість за обсягом трафіку, що передається мережею Інтернет. Особливе визнання отримав обмін відеофайлами і передача потокового відео. Великого комерційного успіху набули проекти Gnutella, PPLine, PPStream, I2P, JXTA та ін.

Можна виділити два основних покоління файлообмінних мереж (file-sharing). Перше покоління – це гібридні конструкції, що поєднують сервери з P2P-маршрутизацією. Друге покоління – це децентралізовані архітектури, які отримали в цей час пріоритетний розвиток. Водночас, протягом останніх років з'явилися нові напрямки анонімізованих P2P-мереж, таких як Freenet, I2P, які можуть організуватися в третє покоління.

В основі технології лежить принцип децентралізації: всі вузли в мережі P2P рівноправні, що забезпечує такі переваги технології P2P перед клієнт-серверних підходом: відмовостійкість при втраті зв'язку з декількома вузлами мережі, збільшення швидкості отримання даних за рахунок копіювання одночасно з

декількох джерел, можливість поділу ресурсів без «прив'язки» до конкретних IP-адрес, величезна потужність мережі в цілому та ін. На сьогодні в Інтернеті більше половини всього трафіку припадає на трафік файлообмінних P2P-мереж, а розміри найбільших із них подолали межу в мільйон одночасно працюючих вузлів, які розподіляють петабайти інформації. Загальна кількість зареєстрованих учасників файлообмінних мереж P2P у всьому світі складає більше 100 млн.

Наразі P2P-мережі в основному використовуються не тільки для роздачі файлів, існує ще багато інших областей, де аналізована технологія теж успішно застосовується, – це телебачення й аудіотрансляції, паралельне програмування, розподілене кешування ресурсів для розвантаження серверів, розсилка повідомлень і статей, підтримка системи доменних імен, індексування розподілених ресурсів і їх пошук, резервна реплікація і створення стійких розподілених сховищ даних, обмін повідомленнями, створення систем, стійких до атак типу «відмова в обслуговуванні», поширення програмних модулів та ін. Є значна кількість клієнтських програм для обслуговування P2P-мереж як комерційних, так і з відкритим кодом. Постійно йде робота із вдосконалення протоколів і збільшення функціональності систем. Очевидно, коли клієнтське програмне забезпечення для P2P буде інтегровано з операційними системами, зросте й додаткова якість. Так, вже зараз великі компанії виявляють інтерес до P2P. Наприклад, компанія Sun розробляє протокол для доступу до основних P2P-мереж із кишенькових комп'ютерів і смартфонів, а компанія Microsoft створила свої реалізації P2P-протоколів Scribe і Pastry. З вирішенням проблеми покращення технологій пірингових мереж, з пошуком нових прикладних рішень працюють сьогодні великі колективи і відомі вчені, серед яких слід назвати В.Столлінгса, Е.Таненбаума, Н.Чепмена і др.

Нині існує безліч програм, спрямованих як на створення нових методів організації пірингових мереж, так і модернізацію вже відомих. Поряд з цим існує велика кількість важливих завдань, що вимагають уваги, серед яких: підвищення якості та достовірності наданого контенту, захист від хакерських атак, ботів, вірусів. Особливу значимість набуває задача пошуку потрібного контенту,

швидко й ефективно знаходження найбільш релевантних відгуків на запити від сид-вузлів до всієї мережі. Крім цих головних йде пошук нових рішень і для суміжних завдань. Однією з найбільш проблематичних є задача фрагментації файлів, яка впливає на затримку надання послуг і з якою пов'язані втрати окремих фрагментів.

Отже, тема дисертаційної роботи є актуальною та практично важливою.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційні дослідження проводились відповідно до наступних програм та нормативно-правових документів: Закон України “Про Основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки” від 09.01.2007 р., № 537-V, Закон України “Про телекомунікації” від 18.11.2003 р., № 1280-IV, Державна науково-технічна програма «Створення перспективних телекомунікаційних систем і технологій», планів наукової, науково-технічної діяльності Харківського національного університету радіоелектроніки. Матеріали дисертації були використані у звіті про науково-дослідну роботу № 308 “Нитка-3” номер держреєстрації 0116U0000662. Крім того, напрям досліджень пов'язаний із планами університету та кафедри телекомунікаційних систем, де виконувалась робота, а також тематикою науково-дослідних робіт цієї кафедри.

Мета досліджень: підвищення якості та достовірності надання контенту в файлообмінних P2P-мережах.

Задачами досліджень є:

1. Аналіз динамічного стану пірінгових мереж при передачі файлів.
2. Вибір найбільш перспективних технологій управління пірінговими мережами.
3. Розробка математичних моделей пірінгових мереж в режимі формування файлів.
4. Обґрунтування методів вирішення задачі оптимізації процесу фрагментації файлів.
5. Вибір методів рішення задачі мінімізації кількості відібраних фрагментів на кожному із кроків їх формування.

6. Синтез і аналіз оптимальної процедури побудови файлу при різних методах їх фрагментації.

7. Порівняльний аналіз методів оптимізації формування файлів на основі динамічного і цілочисельного лінійного програмування.

Аналіз якості функціонування алгоритмів фрагментації при різноманітних методах планування файлів в умовах різного навантаження і при різних швидкостях передачі контенту.

Об'єкт дослідження: процес обміну файлами в децентралізованих неструктурованих оверлейних мережах.

Предмет дослідження: синтез і аналіз стану фрагментації файлообмінних пірингових мереж і методів щодо забезпечення якості надання послуг.

Методи досліджень. Для синтезу моделі фрагментації файлів була використана динамічна рекурсивна дискретно-неперервна математична модель випадкового процесу, заданого на інтервалі скачування файлів. Для вирішення задач оптимізації було використано методи теорії розкладів, а також теорії динамічного програмування. Оптимізаційні задачі вирішено за допомогою цілочислового лінійного програмування. Для аналізу продуктивності мережі були використані інтегро-диференційні моделі. Для моделі аналізу якості роботи пірингової мережі застосовано методи аналізу і синтезу диференційних моделей скачування і завантаження контенту. Для аналізу якості планування контенту залучено стандартні методики P2PTV Sim3, SSSim.

Наукова новизна отриманих результатів:

З використанням доступних інформаційних джерел, численних результатів інших дослідників і отриманих особисто автором аналітичних і експериментальних даних, для дисертаційних досліджень обрано технологію відбору і доставки фрагментів через Інтернет на базі децентралізованих неструктурованих пірингових P2P-мереж, що володіють більшою масштабованістю і надійністю щодо клієнт-серверних структур в умовах динамічних навантажень. З аналізу стану цих мереж отримані наступні результати, які представлені в дисертаційній роботі:

1. Вперше для процесів фрагментації скачуваних файлів запропонована динамічна дискретно-неперервна модель, що дозволяє враховувати суттєві відмінності в динаміці оверлею, положенні фрагментів, в умовах їх доставки, викликаних незалежним характером сид-роздавальних пірів. З використанням даної моделі вирішена задача оптимізації процедури пошуку і відбору кращих із запропонованих фрагментів.

2. Вперше на основі розробленої методики розподілу ресурсів пірингової мережі проведено аналіз продуктивності відомих алгоритмів реплікації файлів і показано, що кількість кроків пошуку потрібного ресурсу безпосередньо залежить від співвідношення загального числа вузлів і загального для мережі обсягу пам'яті для зберігання файлів. За результатами досліджень, більш якісним, порівняно з уніфікованим і пропорційним, виявився алгоритм реплікації квадратного кореня.

3. Вперше розроблено динамічну модель оцінки якості пірингової мережі і методику аналізу умов функціонування при різній стратегії завантаження пірів у вигляді нормованого співвідношення швидкостей зачачування (download) і завантаження (upload) контенту. Показано, що в стаціонарних умовах показник якості зростає пропорційно числу активних пірів. Водночас, в умовах інтенсивної динаміки, при масовому підключенні нових пірів, монотонність зростання якості мережі погіршується, що вимагає застосування обмежувальних заходів. Активним пірам рекомендується підтримувати щедрю альтруїстичну стратегію, що виражається в підтримці високого темпу пропозицій на завантаження.

4. Вперше з використанням запропонованої дискретно-неперервної моделі методом динамічного програмування поставлена і вирішена задача оптимізації процесу фрагментації скачуваних файлів по критерію мінімуму втрат, пов'язаних з упорядкуванням фрагментів та їх затримкою та обмеженим передуманням, що забезпечило скорочення часу на формування і доставку файлів.

5. Вперше з урахуванням медіаданих щодо алгоритмів кодування і структури відеокадрів (медіаобізнаності мережі) з використанням методів теорії розкладів отримано рішення задачі фрагментації, яке зводиться до задачі цілочисельного лінійного програмування і показано, що при використанні критерію мінімуму

середньозваженого часу скачування для заданих файлів вдається отримати більш якісний результат в порівнянні з загальним рішенням на основі динамічного програмування.

6. З використанням стандартних методик P2PTV SIM3 і SSSIM проведено аналіз якості відео з урахуванням особливостей планування контенту за схемою LUC / RUC (останній корисний фрагмент; випадковий корисний пір). Аналіз втрат фрагментів показав, що ці втрати зростають зі збільшенням бітрейту [Мб/с] і при менших затримках. Дослідження якості відтворення за критеріями пікового відношення сигнал/шум PSNR, середньоквадратичної похибки MSE й індексу структурної подібності SSIM дозволяють вибрати і вдосконалити відповідний режим стану мережі. Показано, що при плануванні з урахуванням особливостей відеоструктури (медіа-обізнаності мережі) якість відтворення значно поліпшується.

Практична значимість роботи полягає в тому, що:

Завдяки запропонованим методам фрагментації вдається в три рази скоротити сам процес фрагментації.

Використання запропонованих методів оптимізації формування файлів у пірингових мережах дозволяє підвищити якість послуг, що надаються шляхом упорядкування процесу завантаження фрагментів. За рахунок скорочення в 2,2 рази дисперсії часу цих формувань, скорочується кількість втрачених файлів, що підвищує оперативність доступу. Надається кількісна оцінка введеного показника якості роботи мережі в нестационарних умовах функціонування. Вводиться показник “егоїзму” або “альтруїзму” активного піра, який впливає на ефективність роботи мережі. Доведено, що планування файлообмінних процесів доцільно проводити з урахуванням медіаструктур, що призводить до зменшення втрат фрагментів.

Наукове значення: запропоновані в дисертації моделі та методи процесу фрагментації забезпечують підвищення якості при наданні послуг в пірингових мережах P2P при скачуванні файлів. Результати наукових досліджень можуть бути рекомендовані для вдосконалення технології передачі відеоконтенту в

мережі Інтернет. До того ж ці рекомендації можуть бути поширені й на інші децентралізовані мережі, які інтенсивно поширюються, зокрема на мережу електронних розрахунків bitcoin. Матеріали дисертаційної роботи використані в навчальному процесі на кафедрі інфокомунікаційної інженерії ХНУРЕ в курсі “Основи телебачення та радіомовлення”, а також результати представлені в навчальному посібнику “Методи наукових досліджень в телекомунікаціях”, Том 1, Харків, Компанія Сміт, 2013 р., розділ 10.7 при аналізі вибору методів планування, для мінімізації втрат фрагментів.

Особистий внесок здобувача. Усі основні наукові результати представлені в дисертаційній роботі й отримані автором самостійно. Зокрема, в роботі [12] запропоновано і проаналізовано ряд функціональних математичних моделей пірингових мереж, у статті [13] розроблено два види математичних моделей пірингових мереж. Перша модель може бути використана при плануванні мережі. Друга модель відображає динамічні властивості пірингової мережі й може бути використана для відображення і організації контролю завантаження і перерозподілу ресурсів мережі. У статті [14] проаналізовані можливі варіанти топології і математичні моделі пірингових мереж. У статті [21] запропоновано комплексний критерій оптимальності процесу фрагментації відеофайлів. У статті [22] автором розроблено та досліджено функціональну модель управління в піринговій мережі. У статті [23] запропоновано метод оптимізації фрагментації TV-контенту за допомогою динамічного програмування.

Апробація. Основні результати дисертації представлені та схвалені на: XVIII Міжнародному молодіжному форумі «Радиоэлектроника и молодежь в XXI веке», 2014; Modern problems of radio engineering, telecommunications and computer science, Proceedings of the XII th International Conference TCSET' 2014, February 25 - March 1, Slavske; 10-й Міжнародній молодіжній науково-технічній конференції RT-2014 «Современные проблемы радиотехники и телекоммуникаций», Севастополь, 12-17 мая 2014; 3-й Міжнародній науковій конференції «Інформація, комунікації, суспільство 2014», 21-24 травня, Славське; IV Міжнародній науково-практичній конференції “Фундаментальные и прикладные

науки сьогодні”, 20-21 октября 2014 г., North Charleston, USA; 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T’2014), october 14-17,2014, Kharkiv, Ukraine.

Публікації. Основні положення і результати дисертаційної роботи опубліковано у 21 науковій праці: 3 статті у закордонних фахових журналах [11, 12, 14], 11 статей у фахових науково-технічних журналах та збірках наукових праць [6-16]. Апробація результатів дисертації проходила в ході 10 доповідей на міжнародних науково-технічних конференціях [17-26], з них апробації на конференціях [19-21, 23, 25, 26], які проходили під егідою IEEE та індексуються в міжнародних наукометричних базах Scopus та IEEE Xplore Digital Library.

Структура та обсяг дисертації. Дисертація складається зі вступу, шести розділів, висновків, списку використаних джерел, що включає 111 найменувань, і двох додатків. Текст дисертації викладено на 178 сторінках, включає 41 рисунок і 7 таблиць.

РОЗДІЛ 1

ФАЙЛООБМІННІ ПІРИНГОВІ МЕРЕЖІ

Файлообмінна мережа – збірна назва однорангових оверлейних пірингових (peer-to-peer) мереж для спільного використання файлів, заснованих на засадах рівноправності, які беруть участь в обміні файлами, тобто кожен учасник одночасно є і клієнтом, і сервером. Користувачі файлообмінної мережі викладають файли в так звану «розшарену» (англ. Share – ділитися) директорію, вміст якої доступний для скачування іншим користувачам.

Загальний принцип роботи пірингових мереж наступний: клієнтська програма передає в мережу список файлів, які вона може надати для скачування і які хоче отримати сама. До того ж, якщо пошук відповідних партнерів здійснюється за допомогою сервера, а самі дані скачуються безпосередньо або, якщо пряме з'єднання не може бути встановлене, за посередництва сервера, тобто реалізується централізована модель. Якщо ж одночасно виконуються функції і клієнтів, і серверів, посылаючи запити один одному, реалізується децентралізована організація. Разом із тим мережі можуть бути одночасно і централізованими, і децентралізованими, тобто реалізується змішаний (гібридний) тип структури.

1.1. Мережі доставки контенту CDN (Content Delivery Network)

CDN – це територіально розподілена мережа відео і Web-серверів, що використовуються для того, щоб доставляти контент інтернет-користувачам з найкращою якістю. Особливу популярність CDN отримала в 2005-2006 р.р. при значному поширенні Інтернет-відео та, зокрема, із зростанням популярності відеохостингу YouTube, який активно використовується і в даний час.

Хостинг означає, що постачальник контенту вирішує бути розпорядником, управляти і підтримувати власний контент на сервері або банку серверів,

незалежно від будь-якої третьої сторони, такий як, наприклад, постачальники послуг мереж доставки контенту.

Більшість відеохостингів використовує класичну клієнт-серверну архітектуру, або її модифікації, які реалізують централізовану схему зберігання і доставки контенту (рис.1.1.)



Рис. 1.1. Клієнт-серверна модель на основі CDN

Основним недоліком централізованого потокового мовлення є необхідність створення потужних відеосерверів і каналів передачі відео з високою пропускнуою здатністю, які повинні підтримувати роботу потокового відео для мільйонів користувачів у реальному часі. Чим більше глядачів одночасно вибирають певний канал, тим більше навантаження на сервер і канали передачі відео. В результаті виникають проблеми зі швидкістю і якістю передачі відеопотоку (зависанням кадрів). На відміну від клієнт-серверної моделі мовлення CDN у пірінговій моделі телебачення якість і швидкість передачі відеопотоку поліпшується зі збільшенням кількості глядачів, тобто, чим більше глядачів, тим вища якість телебачення. З цієї причини на сучасному етапі досить використовуване й затребуване децентралізоване Internet-TV або пірінгове телебачення, яке засноване на тих же принципах, що й робота торрентів (торрент – стрімкий потік).

Використання торрент-технологій дозволяє плеєру глядача завантажувати фрагменти відео не з централізованого CDN-сервера, а вибирати необхідні фрагменти з кешу (де вони тимчасово зберігаються) комп'ютерів сусідів, які переглядають це відео. Торрент-технології значно розвантажують сервери для розміщення відеофайлів. Перспективними моделями мовлення також є гібридні, побудовані на основі клієнт-серверної моделі CDN і пірингової моделі телебачення P2P. Гібридна модель забезпечує можливість одночасної передачі відеофрагментів клієнту як з централізованого сервера (за протоколом HTTP), так з і комп'ютерів клієнтів через P2P мережу (з мережних протоколів BitTorrent). У такій гібридній моделі HTTP-сервери служать тільки в якості серверів підтримки, а увесь основний трафік децентралізовано розподіляється між споживачами через P2P-мережу.

Гібридна модель мовлення P2P CDN реалізована, наприклад, на основі технологій Ace Stream (Рис.1.2). ACE Stream HD – це кроссплатформний програмний додаток. У пакет програмного забезпечення Ace Stream Media включені серверні й клієнтські програми (як настільна програма, так і плагіни для браузерів): ACE Stream HD, плагін для веб-браузерів Ace Stream P2P Multimedia Plug-in, мультимедійний плеєр ACE Player HD і TS Magic Player – додаткове розширення для браузерів Mozilla Firefox, Google Chrome і Opera. Платформа забезпечує відтворення відео в режимах: iVoD і Live Stream.



Рис. 1.2. Модель ACE Stream P2P

До іншого типу гібридної моделі відносяться: PeerCDN, FlockPlay, Viblast, Peer5, Swarmify, StreamRoot, в яких використовуються технології WebRTC. У цих моделях застосовується клієнт-серверна модель на основі CDN і пірингова модель телебачення на основі технології WebRTC. Для організації розподіленої CDN в цих проектах браузері (піри) працюють як сервери (PeerServer) для інших браузерів через пірингові P2P-канали з використанням технології WebRTC.

Необхідно відзначити, що WebRTC є частиною проекту HTML5. WebRTC забезпечує доступ браузера до веб-камери і мікрофону користувача засобами HTML5 і JavaScript без додаткових плагінів і розширень.

Мережі доставки контенту – це оверлейні мережі, тому вони складені з дзеркальних серверів (званих також «дублюючими серверами»), які є ні чим іншим, як кінцевими системами (це обмежує зв'язки TCP/IP). Дзеркальні сервери, на додачу до Інтернету, ефективно використовують мережевий рівень обслуговування Інтернету. Дзеркальні сервери розміщуються в центрах даних постачальників інтернет-послуг (ISP). Таким чином, мережі доставки контенту складені з оверлейної мережі великого числа дзеркальних серверів, розподілених в Інтернеті.

Є деякі особливості в процедурі доставки.

По-перше, мережі CDN мають інфраструктуру «колективного користування», засновану на тому, що дзеркальні сервери, що належать CDN-постачальнику послуг, зазвичай «розділяються» між численними постачальниками контенту. Ці постачальники контенту є клієнтами постачальників послуг мереж доставки контенту. По-друге, оцінка наданого контент-провайдером постачальника послуг мереж доставки контенту вимірюється обсягом контенту, завантаженого з дзеркальних серверів, і в піковій пропускній спроможності вони повинні обслуговувати контент від дзеркальних серверів, тому що під час відсутності постачальника послуг мережі доставки контенту їх власна інфраструктура повинна буде зробити те ж саме.

Типове використання мереж CDN представлено на рис.1.3. Сервер джерела може знаходитись далеко, наприклад, на іншому континенті, а кілька дзеркальних серверів встановлюється в інших регіонах.

Контент з сервера джерела (СД) копіюється «0» на дзеркальні сервери (ДС). Коли клієнти запитують контент сервера джерела «1», останній повертає веб-сторінку, в яку впроваджені посилання, які містять символічні імена дзеркальних серверів «2», що відсилають на статичний контент. Далі це ім'я зводиться сервером мережі (СМ) постачальника CDN до IP-адреси ЗС, найближчого до клієнта «3». Потім клієнт встановлює зв'язок з найближчим ДС і завантажує зображення і відео.

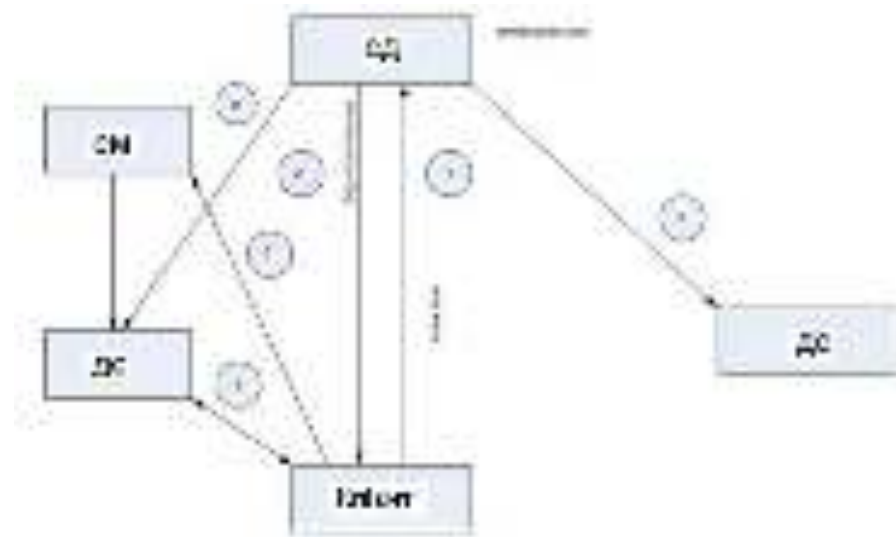


Рис.1.3. Схема доставки відеоконтенту CDN за допомогою дзеркальних серверів (ДС). СД – сервер-джерела, СМ – сервер-мережі

Одна з головних труднощів для постачальників послуг мережі доставки контенту – спроможність знайти найближчий дзеркальний сервер, відповідний клієнту. Щоб ідентифікувати місце знаходження клієнта, зазвичай використовується IP-адреса клієнта.

На швидкість завантаження веб-сторінки і її вміст сильно впливає те, наскільки далеко користувач знаходиться від сервера. Це відбувається через те,

що при використанні технології TCP/IP, яка застосовується для поширення інформації в мережі Інтернет, затримки при передачі інформації залежать від кількості маршрутизаторів, що знаходяться на шляху між джерелом і споживачем контенту. Розміщення контенту між декількома серверами за допомогою засобів CDN скорочує мережевий маршрут передачі даних і робить завантаження сайту швидшим з точки зору користувача.

Використання CDN знижує кількість хопів, що істотно збільшує швидкість скачування контенту з мережі Інтернет (рис.1.4). Кінцеві користувачі відчувають меншу затримку при завантаженні контенту, відсутність різких змін швидкості завантаження і високу якість потоку даних. Встановлена стабільність дозволяє операторам CDN доставляти відеоконтент в форматі HD, забезпечувати швидке завантаження файлів великих розмірів або організувати відеомовлення з високою якістю сервісу (QoS) і низькими витратами на мережу.

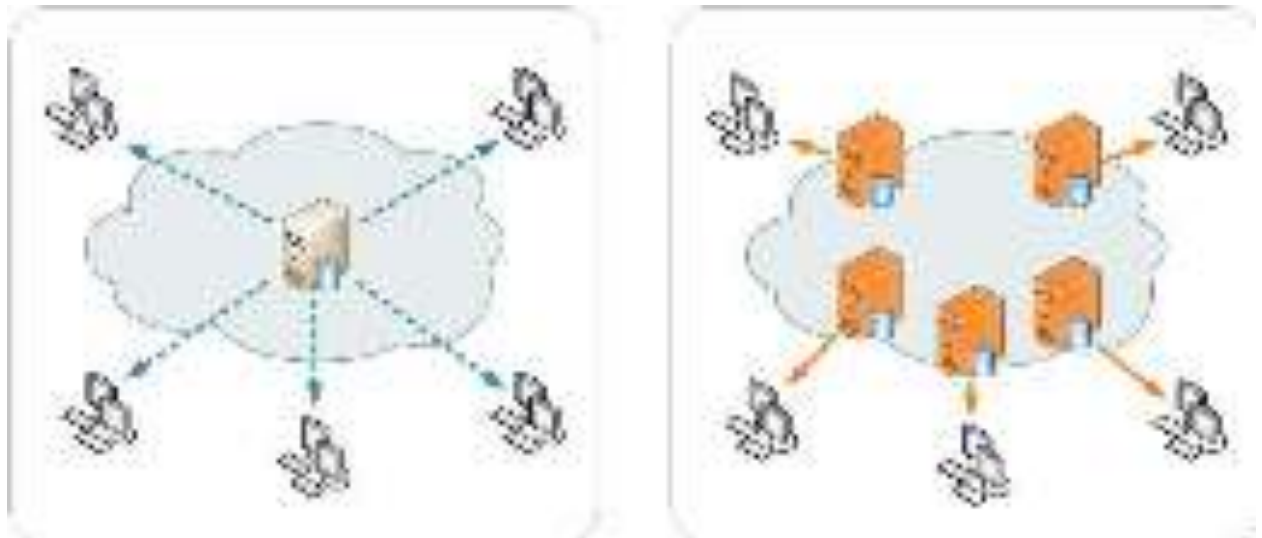


Рис.1.4. Одноточкове (зліва) та дистрибуція засобами CDN (справа)

1.2. Загальні відомості про однорангові файлообмінні мережі

За останні роки сильно зросла соціальна роль Інтернету, з'явився ряд важливих додатків, заснованих на децентралізованій архітектурі мережі з

одночасним підключенням мільйонів користувачів до змісту загального характеру. При цьому виключаються сервери для посередництва між кінцевими системами, на яких працюють додатки, а їх поведінка в мережі описується як оверлейна мережа, в якій рівнонаправлені протоколи формують віртуальну мережу на основі фізичної мережі.

Соціальна роль систем P2P далеко виходить за межі «мереж загального користування», їх роль на цей час можна трактувати як масово розподілені платформи для зберігання і добування інформації.

P2P - Service Model – це певна модель в Service-Oriented Peer-to-Peer Service (SOPPS) [22], яка представлена в Європейському дослідницькому проекті Market Management of Peer-to-Peer Services (MMAPPS). Дана модель описує різні типи сервісів, які пір може запропонувати іншим пірам, а також інтерфейси, за допомогою яких вони можуть бути доступні.

Зокрема, дана модель обслуговування, що представлена в [52], містить логічну структуру, розділена на три різних рівні, які представлені в висхідному порядку (рис.1.5):

1. Рівень оверлею, де здійснюється з'єднання та від'єднання піру, маршрутизація, балансування навантаження, самоорганізація накладеної мережі.
2. Рівень фінансового обліку, який забезпечує монетизацію, виконання процедур розрахунків для кредитування і дебетування користувачів.
3. Рівень ринку, який включає в себе більшість послуг, що отримують кінцеві користувачі: цілеутворення, аукціон, інтелектуальна власність, поширення контенту, довіра і репутація, та ін.

Ці модулі надають різні послуги для інших модулів, і представляються у вигляді групи протоколів. Додатки можуть використовувати функції та операції, запропоновані на різних рівнях. Кожен модуль визначається в умовах функцій, які вони експортують в інші частини структури [76]:



Рис. 1.5. Логічні рівні P2P сервісної моделі

За звітом німецької компанії Iroque, частка інтернет-трафіку, що припадає на P2P мережі, становить від 49% на Близькому Сході і до 84% в Європі. У міру поширення широкосмугового Інтернету кількість користувачів файлообмінних мереж швидко збільшується.

1.2.1. Визначення та властивості P2P-систем

Наведемо відомі визначення [22], що охоплюють поняття розподілу ресурсів, самоорганізації, децентралізації і взаємозв'язку:

Визначення для P2P-мережі

«Розподілена мережева архітектура може бути названа peer-to-peer-мережею, якщо учасники поділяють частину власних апаратних ресурсів (обчислювальна потужність, ємність сховища, пропускна здатність мережі, принтери). Ці загальні ресурси необхідні для надання послуг і контенту, пропонованих мережею (наприклад, спільне використання файлів або спільне використання робочих просторів для спільної роботи).

Мережа P2P є розподіленою системою, що містить взаємопов'язані вузли, здатні до самоорганізації в мережеву топологію з метою поділу ресурсів, таких як контент, цикли процесора, пристрої зберігання і смуга пропускання, що

адаптується до відмов і змінному числу вузлів, підтримуючи при цьому прийнятний рівень зв'язності й продуктивності без необхідності в посередниках або підтримки глобального центрального сервера.

Згідно з більш широким визначенням, P2P є класом додатків, які використовують ресурси: жорсткі диски, цикли процесора, контент, доступні на краю хмари Інтернету. У той же час це визначення підходить і для систем, які застосовують для свого функціонування централізовані сервери (таких як *seti@home*, системи миттєвих повідомлень або навіть сумнозвісна мережа Napster), а також різні додатки з області grid-обчислень (гратчастих обчислень).

Визначення оверлейної мережі [26]:

«Віртуальна або логічна мережа на рівні додатків, у якій кінцеві точки мають доступну адресу і забезпечують взаємодію, маршрутизацію і обмін повідомленнями між кінцевими точками. Оверлейні мережі часто використовуються в якості основи для розгортання нових мережевих послуг або для забезпечення топології маршрутизації, недоступної в базовій фізичній мережі. Багато peer-to-peer-систем являють собою оверлейні мережі, які працюють по Інтернету».

Визначення для оверлея.

Оверлей – це такий спосіб використання оперативної пам'яті, при якому в одну і ту ж ділянку пам'яті, звану оверлейним буфером, поперемінно, в міру потреби, завантажуються різні оверлейні (такі, що перекриваються) модулі. При цьому всі оверлейні модулі в готовому до роботи вигляді зберігаються на диску, а в оперативній пам'яті в кожен момент знаходиться лише один активний модуль і, можливо, невелике число неактивних.

Нехай, наприклад, програма (рис. 1.6) складається з головної частини MAIN і двох модулів А до В, а LM, LA і LB – відповідно довжина головної частини і обох модулів, причому $LA > LB$. Тоді неоверлейна програма займе в пам'яті $LM + LA + LB$ байт, в той час як оверлейна програма, – лише $LM + LA$ байт.

При виконанні оверлейної програми в пам'ять спочатку завантажуються головна частина і один з модулів, наприклад, модуль А. Якщо в процесі

виконання програми зустрінеться звернення до модуля В, програма призупинить свою роботу, з диска в оверлейний буфер буде завантажений модуль В (модуль А при цьому частково знищується), після чого програма продовжить свою роботу. Якщо в подальшому зустрінеться звернення до А, точно таким же чином буде завантажений модуль А, причому завантаження потрібних модулів в оверлейний буфер здійснюється автоматично і програмісту не потрібно про це дбати.

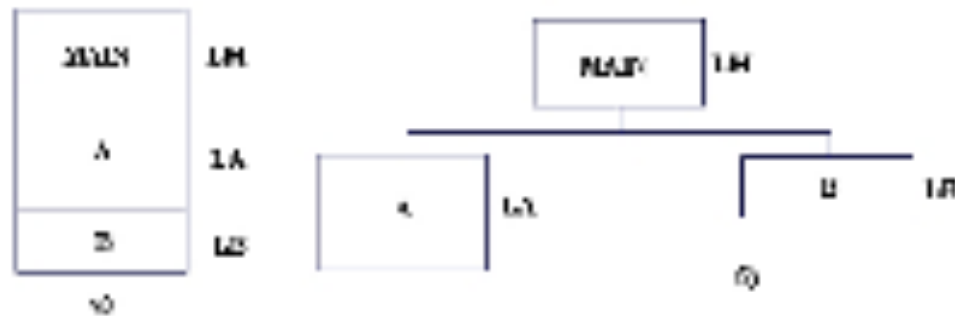


Рис.1.6. Приклад структури програми: а) неоверлейна; б) оверлейна

Описаний механізм виявляє головну перевагу оверлейної структури: обсяг оперативної пам'яті, займаної оверлейною програмою, визначається довжиною її головної частини і найбільшого з модулів, що перекриваються, в той час як при неоверлейній структурі в цей обсяг входить сумарна довжина всіх модулів. Чим більше в програмі оверлейних модулів і чим менше довжина найбільшого з них, тим більший виграш в пам'яті дає оверлейна структура. Однак зовсім очевидний і головний недолік таких структур: на кожне завантаження оверлейного модуля з диска в оверлейний буфер потрібен додатковий час, тому оверлейна програма буде виконуватися з меншою швидкістю.

Розподіл контенту.

До цієї категорії належить більшість сучасних P2P-мереж, що включають системи та інфраструктури, розроблені для поділу цифрової аудіовізуальної інформації та інших даних між користувачами. Спектр таких систем для розподілу контенту починається від відносно простих додатків для прямого поділу файлів і простягається до більш складних, які створюють розподілені

середовища зберігання, що забезпечують безпеку і ефективні організацію, індексацію, пошук, оновлення та вилучення даних. Як приклад можна навести мережу Cool Streaming, PPStream, PPLive, Gnutella, Kazaa, Freenet і Groove. Надалі сфокусуємо увагу саме на цьому класі мереж.

Децентралізація поведінки P2P системи визначається колективними діями пірів, в системі також відсутня центральна контрольна точка. Однак деякі системи забезпечують безпеку системи P2P з використанням центрального сервера входу в систему.

Вузли симетричності приймають рівні ролі в роботі P2P системи. У багатьох проектах ця властивість послаблюється за допомогою спеціальних ролей пірів, таких як суперпіри або ретранслятори (дзеркальні сервери).

Автономна участь піру в P2P системі визначається локально, для системи P2P немає єдиного адміністративного контексту.

Самоорганізація системи P2P збільшується в міру використання ефективних методів потрібних файлів і операцій кожного піру.

Балансування навантаження. Піри використовують різномірний ресурс (пропускну здатність, обчислювальну потужність, простір для зберігання і реплікації файлів. Система повинна бути в змозі балансувати навантаження серед пірів.

Масштабованість – це попередня умова функціонування P2P-систем з мільйонами вузлів, що працюють одночасно. Це означає, що ресурси, використовувані кожним піром, демонструють швидкість росту в залежності від розміру оверлею, який менший лінійного. Це також означає, що час відгуку не збільшується більш, ніж лінійно залежно від розміру оверлею.

1.2.2. Правила файлового обміну в пірингових мережах

Для того, щоб споживач міг стати повноправним учасником тієї чи іншої файлообмінної мережі, необхідно встановити на свій комп'ютер спеціальну

програму-клієнт, за допомогою якої і буде здійснюватися обмін файлами. Причому для різних мереж використовується різне програмне забезпечення.

Однією з основних сфер застосування P2P-мереж є обмін файлами. Незважаючи на переваги такого застосування, наприклад, високу швидкість поширення запитів і надійність самої системи, існують і недоліки. Одним із таких можна назвати складність пошуку саме тих користувачів, у яких є потрібний в даний момент часу файл і в той же час знаходяться в мережі в активному стані. Здолати ці труднощі вдалося за допомогою використання гібридних мереж, де все ж існують виділені сервери, на які покладено функції координації роботи, пошуку існуючих машин в мережі, а також визначення їх поточного статусу («в мережі» або «не в мережі»).

У пірінгових мережах є певний набір правил, яких необхідно дотримуватися і намагатися не порушувати. В основі P2P-мереж лежить принцип рівноправності, з якого випливає, що файли потрібно не тільки завантажувати, але і обов'язково ними ділитися. Це найголовніше правило, порушувати яке не можна. Якщо ж потрібно тільки отримувати інформацію, то в такому випадку можна звернутися до множинних файлообмінних серверів типу DepositFiles, Rapidshare, Letitbit, Vip-Files і інших, що використовують клієнт-серверну архітектуру. Однак, безкоштовно завантажувати дані з таких серверів вийде тільки після перегляду реклами і на низькій швидкості.

Беручи участь в обміні файлів в пірінгових мережах бажано більше віддавати інформації, ніж отримувати або як мінімум робити це в рівних частинах. У теорії масового обслуговування є аналог цієї вимоги: потік інтенсивності обслуговування повинен перевищувати потік заявок, тобто:

$$\frac{\lambda}{\mu} \leq 1.$$

Якщо обсяг завантажуваної інформації буде перевищувати кількість інформації, що віддається, в кілька разів, швидше за все до такого учасника будуть застосовані штрафні санкції, що не дозволяють отримувати дані до тих пір, поки користувач не почне ними ділитися з іншими.

Таким чином, слід дотримуватися наступних правил:

- Після того, як закінчилося скачування файлу, не потрібно його відразу видаляти. Поки є можливість, слід продовжувати тримати його на вашому комп'ютері, щоб дати можливість іншим завантажувати його.
- Не слід переміщати скачані файли відразу після їх завантаження, інакше вони стануть недоступними для інших користувачів.
- Без особливої необхідності не варто занижувати швидкість вихідного з'єднання. Природно, це є певним навантаженням на комп'ютер, проте при низькій швидкості віддача обсягу інформації, одержуваного від вас іншими користувачами, буде незначною.
- Слід пам'ятати, що ваші інтереси можуть не збігатися з інтересами інших користувачів, так що для такого випадку тримайте завжди на своєму комп'ютері кілька популярних файлів, які будуть цікавими багатьом.

1.3. Типи організації файлообмінних мереж

1.3.1. Централізовані мережі

За структурою файлообмінні мережі поділяються на централізовані й децентралізовані. Централізовані мережі – мережі з сервером. На рис.1.7 представлена модель централізованої мережі P2P. Незважаючи на те, що кожен учасник файлообмінної мережі є і клієнтом, і сервером, необхідна інфраструктура для об'єднання розрізнених між собою учасників у певну спільноту. У централізованих файлообмінних мережах службову інформацію зберігають індексаційні сервери.

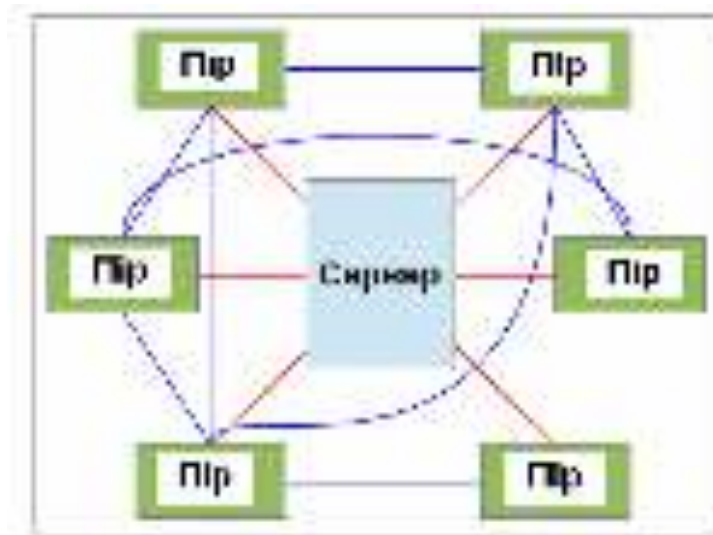


Рис.1.7. Централізована модель P2P-мережі

Перевагою такої мережі є відносна простота програмування та невеликий обсяг службової інформації, переданої на сервери. Відомо, що централізовані мережі більш організовані й продуктивні, однак менш надійні, оскільки не функціонують без сервера. Так, основною причиною закриття мереж Napster, Audiogalaxy і WinMX була ненадійність централізованих файлообмінних мереж.

Локальні файлообмінні мережі, такі як Bit Torrent-трекери, Direct Connect (DC ++), теж належать до централізованих мереж, однак припускають використання великої кількості серверів різного масштабу (так званих «трекерів» або «хабів»). Незважаючи на «локальність», існують досить великі файлообмінники світового масштабу: The Pirate Bay, великі DC ++ хаби і т.д. [28]

1.3.2. Децентралізовані мережі

Децентралізовані файлообмінні мережі функціонують без індексаційних серверів (рис.1.8). Хоча обсяг переданої службової інформації в них більше, їх надійність набагато вища. До децентралізованих мереж відносяться Gnutella, Overnet, Kad та ін. Залежно від рівня децентралізації системи можуть бути розділені на неструктуровані й структуровані. У структурованих мережах

розташування файлів відоме, що значно прискорює їх пошук. Неструктуровані мережі більш стійкі до динаміки стану пірів та узгоджені з динамікою самого Інтернету.

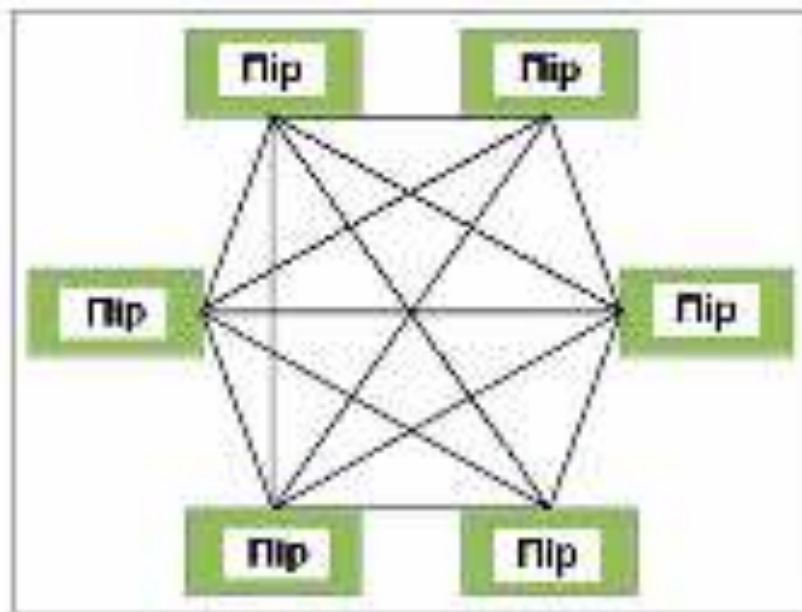


Рис.1.8. Узагальнена схема однорангової взаємодії

Файлообмінники третього покоління (повністю децентралізовані мережі), такі як Overnet і Kad Network (для клієнтів типу eMule), Gnutella, DHT (і magnet посилання) в Bit Torrent, а також японські WinNY, Share і Perfect Dark. Через свою анонімність і повну децентралізованість особливо популярні Gnutella, Пікрелейтед. Найбільший успіх і популярність пірингові децентралізовані мережі мають в Японії через масову доступність швидкого і майже безкоштовного Інтернету при забороні на пром без цензури і копіразму в країні. Різні сервіси всередині i2P також потрапляють до цієї категорії.

1.3.3. Частково децентралізовані (гібридні) мережі

Крім чистих P2P-мереж, знаходять застосування також гібридні мережі, в яких є сервери, які використовуються для координації роботи, пошуку або надання інформації про існуючі комп'ютерні мережі та їх статус (on-line, off-line і

т.д.). Гібридні мережі поєднують швидкість централізованих мереж і надійність децентралізованих завдяки гібридним схемам із незалежними індексаційними серверами, що синхронізують інформацію між собою.

Гібридні схеми поєднують швидкість централізованих мереж і надійність децентралізованих. У них використовуються незалежні індексаційні сервери, що постійно синхронізують інформацію між собою. Таким чином, при виході з ладу одного з них, мережа продовжує функціонувати. До частково децентралізованих файлообмінних мереж відносяться eDonkey2000, OpenNap.

Частково централізовані мережі з багатьма серверами. Файлообмінники другого покоління, наприклад, eDonkey, KaZaA успішно використовуються у файлообміні.

Даркнети – пирингові мережі останнього покоління, що використовують спеціальні прийоми анонімізації, засновані на обмеженні безпосередніх з'єднань деякою кількістю визначених користувачем так званих «друзів» (принцип F2F: friend-to-friend), що теоретично дозволяє уникнути будь-яких деанонімізуючих атак. Тим часом, принцип F2F має на увазі велику кількість посередників між кінцевими вузлами, що негативно позначається на продуктивності мережі. Приклади: FreeNet (в опціональному режимі F2F), GNUnet (в опціональному режимі F2F), Retro Share [29].

1.3.4. Особливості децентралізованих технологій

В основі технології децентралізації лежить принцип рівноправності всіх вузлів в мережі P2P, що забезпечує переваги технології P2P перед клієнт-серверних підходом:

- відмовостійкість при втраті зв'язку з декількома вузлами мережі;
- збільшення швидкості отримання даних за рахунок копіювання одночасно з декількох джерел;
- можливість поділу ресурсів без «прив'язки» до конкретних IP-адресів;
- величезна потужність мережі в цілому та ін.

Зараз на технології P2P заснована величезна кількість популярних мережеских сервісів – від простого обміну файлами до мовного та відеозв'язку, можна виробляти розподілені обчислення, що дозволяють задіяти віддалені комп'ютери користувачів для виконання складної обробки даних [30].

В Інтернеті більше половини всього трафіку припадає на трафік файлообмінних мереж, а розміри найбільших із них перевалили за позначку в мільйон одночасно працюючих вузлів, що містять петабайт інформації. Загальна кількість зареєстрованих учасників мереж P2P у всьому світі складає близько 300 млн [31].

Мережа P2P – це множина вузлів (комп'ютерів, смартфонів та ін.), об'єднаних у єдину систему і взаємопов'язаних за допомогою власного протоколу, що забезпечує можливість створення і функціонування мережі рівноправних вузлів. Протоколом визначається логічна топологія мережі, механізм підключення до неї і відключення від неї вузлів, алгоритм їх взаємодії, вирішення завдань корекції помилок, регламентування форматів повідомлень, прохання про надання послуг і відгуків, маршрутизація в умовах постійного підключення і відключення вузлів. Протоколи P2P (в моделі стека мережеских протоколів TCP/IP) відносяться до прикладного рівня семирівневої моделі взаємодії.

Таким чином, P2P-мережа є накладеною (overlay), що функціонує поверх Інтернету і використовує існуючі транспортні протоколи.

Велике зростання популярності децентралізування мереж P2P обумовлене привабливим набором мережеских характеристик даної технології – це розподіленість мережі, що самоорганізується, яка забезпечує такі переваги, як простота і дешевизна реалізації та підтримка роботи мережі, її відмовостійкість і масштабованість, збільшення швидкості копіювання і колосальна потужність мережі в цілому. Кожен новий вузол збільшує потенційну продуктивність і пропускну здатність мережі. Розосередження ресурсів, що відбулося завдяки виникненню пірінгових мереж, призвело до фактичного зникнення витрат, пов'язаних з підтриманням гігантської централізованої інфраструктури.

1.3.5. Безпека пірингових P2P-мереж

Використання відкритих розподілених P2P-систем має не тільки плюси, а й мінуси, пов'язані з особливостями забезпечення безпеки [32].

Отримати контроль над настільки розгалуженою і великою структурою, або використовувати прогалини в реалізації протоколів для власних потреб – бажана мета для зловмисників. Захистити розподілену структуру набагато складніше, ніж централізований сервер. Настільки величезну кількість ресурсів важко шифрувати/розшифровувати, тому більша частина інформації про IP-адреси і ресурси учасників зберігається і пересилається в незашифрованому вигляді. При перехопленні зловмисник не тільки отримує власне інформацію, але також дізнається і про вузли, на яких вона зберігається. Інша проблема – можливість підробки ID серверів і вузлів. При відсутності механізму перевірки достовірності службових повідомлень, що пересилаються, існує можливість фальсифікації сервера або вузла, що призведе до компрометації всієї мережі або її частини.

Незважаючи на широкі можливості, які дає використання P2P-технології, в файлообмінних мережах знаходить поширення і неліцензійний контент. Тут і виникає основна колізія між правовласниками й інтернет-користувачами. Правовласники намагаються законними методами вести боротьбу з P2P-мережами. Разом з тим, припинити поширення файлів в децентралізованій піринговій мережі технічно неможливо – для цього буде потрібно фізично відключити від мережі всі машини, на яких розташовані ці файли, а таких машин може бути досить багато – в залежності від популярності файлу їх число може досягати сотень тисяч.

У всіх розвинених країнах з кожним роком посилюється законодавство в області захисту авторських і суміжних прав. Зараз під дію антипіратських законів в США, Німеччині та деяких інших країнах потрапляють як власники серверів, що забезпечують роботу P2P-мереж, так і самі користувачі, що скачують і роздають піратський контент.

1.3.6. Побудова пірингових мереж

Парадигми, що лежать в основі роботи пірингових систем, поділяються на два класи: деревовидні системи (tree-push) і мережеві системи (mesh-pull). У першому випадку вузли утворюють дерево в оверлейному шарі, і контент, що передається, проходить від джерела до кінцевих абонентів.

На рис.1.9 представлена схема з кодуванням субпотоків (MDS) і поділом потоку на чотири субпотоки (4 дерева). Літерами G1, G2 і G3 позначені покоління вузлів-партнерів (peer), які беруть участь у доставці відеоданих інших вузлів.

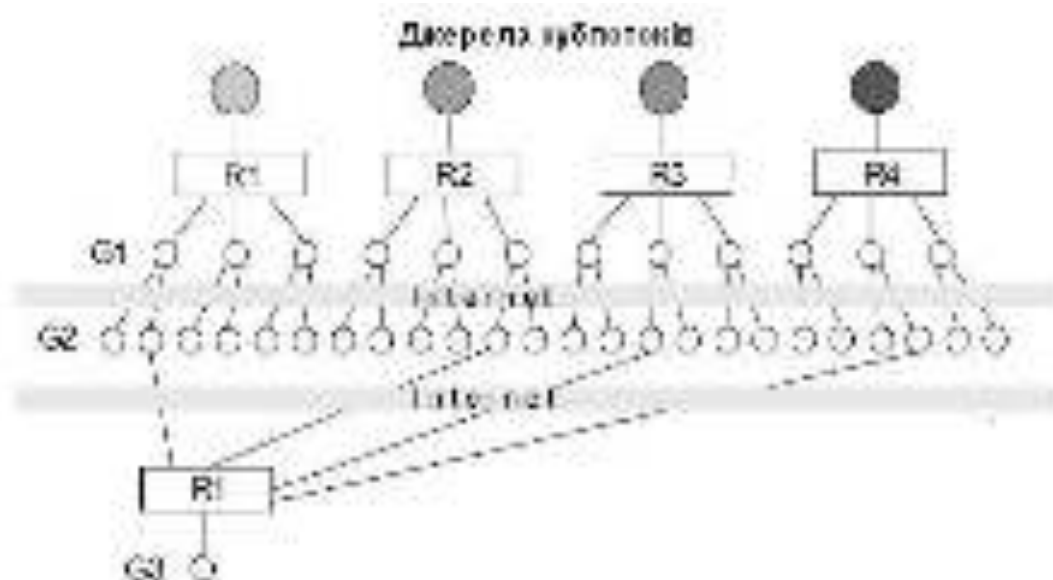


Рис. 1.9. Деревоподібна топологія з чотирма деревами

Другий підхід характерний для тих систем, де вузли в оверлейному шарі утворюють віртуальну сітку і скачують один у одного фрагменти файлів.

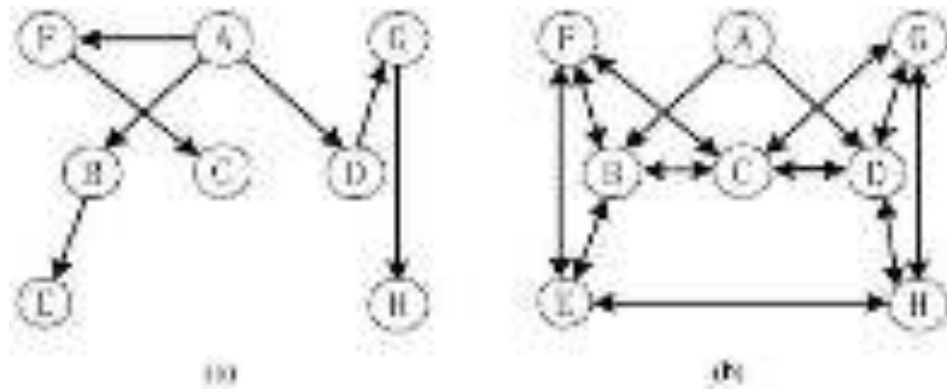


Рис. 1.10. Топологія решітки GBS (Gossip Based Streaming)

При сітчастій топології потік даних ділиться на невеликі модулі, наприклад, відповідні одній секунді фільму. Ці модулі клієнт отримує від вузлів сітки випадковим чином або згідно з деякою закономірністю.

На рис.1.10 показана схема роботи алгоритму Gossip Based Streaming. Вузлом-джерелом первинного відеопотоку тут є вузол А. Еволюція графа доставки відеофрагментів залежить від того, які вузли направляють запити першими. Розмір сегментів зазвичай постійний. Припустимо, що вузол А може обслужити не більше трьох вузлів (обмеження вихідної смуги). При розвитку дерева доставки враховується вихідна смуга вузлів, яка передбачається більшою, ніж потрібно для одного відеопотоку, і випадковий характер вибору партнера – джерела даних. Вузли інформують партнерів (не тільки сусідів) про наявність у них відеосегментів. Будь-який із вузлів може стати відправником, якщо у нього є потрібний сегмент і йому надійшов запит на цей сегмент. Для будь-якого з відеосегментів можна відстежити його шлях від А до вузла редистриб'ютера. Для такої схеми характерна велика затримка відтворення. Причина в тому, що при M кроках маршруту доставки сегмента затримка $\tau = Mx(T_{пер} + T_{диал})$, де $T_{пер}$ – час передачі сегмента між суміжними вузлами (~ 1 с), $T_{диал}$ – час оповіщення про наявність сегмента і обслуговування запиту.

Використання деревовидної парадигми дає такі переваги, як:

1. Велика масштабованість.

2. Відмовостійкість.
3. Зниження середнього навантаження на канали передачі даних.
4. Немає необхідності введення додаткових елементів в інфраструктуру мережі.

У той же час є і недоліки:

1. Істотна затримка початку відеотрансляцій. Незадовільна синхронізація надходження трафіку на різних вузлах, що позначається на якості відеоконтенту;
2. Необхідність зберігання частин даних у себе. Створення політики управління буфером пам'яті;
3. Є можливість недоотримати деякі фрагменти даних (через їх відсутність на сусідніх вузлах), що веде до погіршення якості відтворення у сід-приймача.

У сучасних системах P2PTV, де використовується друга парадигма побудови мережевої архітектури у вигляді mesh-pull лежить припущення, що відеодані розділені на фрагменти і розподілені по вузлах системи. Разом із цим, існує кілька серверів – постачальників нових відеоданих у мережу, які постійно працюють. Всі вузли – учасники мережі обмінюються один з одним буферними картами (buffer maps) – файлами, в яких зазначено наявність тих чи інших відеофрагментів у вузлах мережі, які можуть знадобитися в поточний період часу. В результаті може бути розглянута наступна модель: є мережа вузлів, з'єднаних каналами зв'язку з різними пропускними здатностями. У кожен момент часу з'являються пари кореспондуючих вузлів між якими необхідно передати потік певної величини протягом кінцевого проміжку часу. Після закінчення часу заявка знімається і звільняється ресурс мережі.

Відповідно завдання полягає в передачі максимального потоку між множинними парами вузлів. Така постановка задачі також відома, як багатопродуктове завдання. При цьому потрібно дотримуватися безперервності відеотрансляції, що полягає в небажаності комутації нового шляху для існуючих маршрутів передачі відеоданих.

1.4. Области застосування файлообмінних P2P-мереж

Існує багато областей, де успішно застосовується P2P-технологія, наприклад, паралельне програмування, кешування даних, резервне копіювання даних [33]. Завдяки таким характеристиками, як живучість, відмовостійкість, масштабованість, пірингові мережі знаходять все більше застосування в системах управління виробництвами й організаціями (наприклад, P2P-технологія сьогодні застосовується в Державному Департаменті США). В даному випадку можливий вихід з ладу частини вузлів або серверів, що несуттєво впливає на керованість всієї системи. Загальновідомо, що система доменних імен (DNS) в мережі Інтернет також фактично є мережею обміну даними, побудованою за принципом P2P.

Реалізацією технології P2P є також популярна в даний час система розподілених обчислень GRID. Ще одним прикладом розподілених обчислень може служити проект `distributed.net`, учасники якого займаються легальним зломом криптографічних шифрів, щоб перевірити надійність.

Розглянемо кілька найбільш популярних пірингових мереж [34].

KaZaA – це змішана пірингова мережа, в якій присутні елементи як централізованої, так і децентралізованої мережі. Існує кілька центральних вузлів, до яких підключаються клієнти. Програма підтримує закачування і може закачувати різні фрагменти файла у різних користувачів. Є можливість попереднього перегляду не повністю завантажених файлів. KaZaA насамперед орієнтована на обмін музикою й відеофільмами, але підтримує і інші формати. У списку знайдених файлів відображається повна інформація про файл: довжина, бітрейт, виконавець, назва композиції, а для відео – розмір екрану і деяка інша корисна інформація.

Napster

Napster став першою широко використовуваною службою P2P для обміну музикою. До появи Napster користувачі Інтернету тільки пасивно управляли підключеними комп'ютерами, наприклад, переглядали новини або перевіряли

електронну пошту. З ростом популярності Napster звичайні користувачі Інтернету почали відкривати свої ПК, щоб активно вносити ресурси і відігравати більш важливі ролі в Інтернеті.

У порівнянні з подальшими додатками P2P, Napster використовує простий, але дуже ефективний механізм для спільного використання і пошуку файлів у мережі. Щоб взяти участь у мережі Napster, новим користувачам необхідно зареєструватися на сервері Napster і опублікувати список файлів, яким вони хочуть поділитися. Щоб знайти спільний файл в мережі, користувачі можуть відправити запит в сервер Napster і отримати список «постачальників», що розміщують файли, відповідні запиту. Передача файла відбувається без участі сервера Napster. Запитуваний файл передається безпосередньо між запитом і постачальником, як показано на рис.1.11.

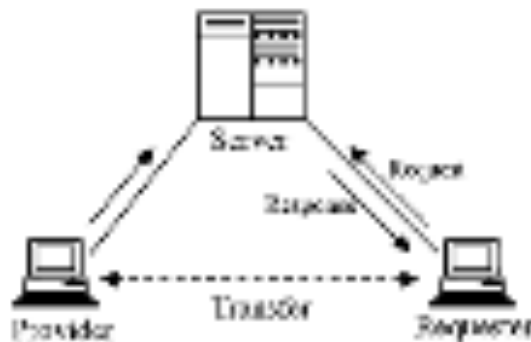


Рис. 1.11. Приклад постачальника в мережі Napster

Gnutella

Файлообмінна мережа Gnutella, перша повністю децентралізована пірингова мережа, була розроблена в 1999 році компанією NullSoft. Але через деякий час America Online, якій належала ця компанія, об'єдналася із звукозаписною компанією і, як наслідок, переглянула своє ставлення до цього проекту. Незважаючи на всі докладені з боку AOL зусилля, мережа Gnutella не припинила свого існування і продовжила роботу.

Супровід і розвиток проекту взяла на себе група ентузіастів. Їх добровільним, хоча не завжди скоординованим зусиллям Інтернет-спільнота зобов'язана виникненню більш двох десятків клієнтів для мережі Gnutella, з яких найбільш відомий Morpheus. До головних недоліків Gnutella належить повільний і часто неточний пошук, що є платою за децентралізований принцип роботи.

eDonkey2000

Мережа eDonkey2000 – гібридна, змішана, побудована на розподілених ресурсах і не залежить від будь-якого одного центрального сервера. Серверну частину може завантажити і встановити собі кожен бажаючий. Через деякий час новий сервер з'явиться в загальному списку. Завдяки відсутності центрального сервера система значно менш вразлива, ніж пірингові мережі, які його мають. Крім того, eDonkey містить вбудований Overnet-клієнт, який використовується для обміну файлами без застосування серверів і працює на власній мережі. Overnet використовує той же протокол передачі, що і eDonkey, але його головна відмінність полягає в тому, що при пошуку файлів він запитує їх у інших користувачів, за рахунок чого досягається добре покриття.

eMule

eMule – клієнт для мережі eDonkey2000, створений і розвивається як проект Open Source групою незалежних розробників. Розробка виявилася настільки популярною, що практично повністю витіснила всіх інших клієнтів для ED2K-мережі. Програма використовує кілька мереж (ED2K, Source Exchange, Kad), щоб створити одну більш надійну.

Shareaza

За відгуками користувачів, Shareaza (<http://www.shareaza.com/>) є одним із кращих універсальних P2P-клієнтів, що швидко завоював популярність у користувачів. Програма працює з чотирма піринговими мережами (Gnutella2, Gnutella1, eDonkey2000/eMule, BitTorrent) і розуміє п'ять видів посилань (Magnet, Piolet, Gnutella, eDonkey2000 / eMule, BitTorrent). Як і eMule, проект розробляється як Open Source, тому програма Shareaza безкоштовна і не містить жодної реклами.

BitTorrent

Технологію BitTorrent – «потік бітів», створив американський програміст Брем Коен (Bram Cohen). Вже в 2001 році у нього була готова перша версія BitTorrent, а через три роки, в червні 2004-го, за даними дослідження компанії CacheLogic, більш третини всього трафіку в Інтернеті передавалося з використанням цього протоколу. Подібний феноменальний успіх обумовлений застосованими автором оригінальних архітектурних рішень.

Анонімізовані пірингові мережі.

Анонімізовані мережі – комп'ютерні мережі, створені для досягнення анонімності в Інтернеті і працюють поверх глобальної мережі. Специфіка таких мереж полягає в тому, що розробники змушені йти на компроміс між ступенем захисту і легкістю використання системи, її «прозорістю» для кінцевого користувача. Також важливий аспект збереження анонімності й конфіденційності за умови впливу методів соціальної інженерії або будь-якого тиску на оператора сервера. Багаторівневе шифрування і розподілений характер анонімних мереж, усуваючи єдину точку відмови і єдиний вектор атак, дозволяють зробити перехоплення трафіку або навіть злом частини вузлів мережі не фатальною подією. За анонімність користувач розплачується збільшенням часу відгуку, зниженням швидкості, а також великими обсягами мережевого трафіку.

У децентралізованій мережі будь-яка машина може встановити з'єднання з іншою, а також надіслати їй запит на надання ресурсів. Кожна машина обробляє запити від інших в якості сервера, відсилаючи і приймаючи запити, а також виконуючи інші допоміжні та адміністративні функції. Будь-який учасник такої мережі не зобов'язаний гарантувати постійне з'єднання і може розірвати його в будь-який момент часу. При досягненні певного розміру мережі в ній одночасно починають існувати безліч серверів з однаковими функціями.

ANtsP2P – файлообмінна мережа, що анонімізує весь потік даних, використовуючи систему маршрутизації, в якій, на відміну від BitTorrent, учасники обмінюються трафіком не безпосередньо, а через кілька вузлів. Кожному учаснику відома тільки IP-адреса його безпосереднього сусіда. Таким

чином, відправник не знає, куди йде його файл, а одержувач не знає, звідки він прийшов. Для більшої безпеки дані між окремими відправниками та одержувачами шифруються згідно з симетричним алгоритмом AES.

Bitmessage – децентралізована P2P система обміну електронними повідомленнями з відкритим вихідним кодом, яка дозволяє користувачеві Bitmessage відправляти зашифровані повідомлення іншим користувачам системи, як альтернатива email. Bitmessage побудована на архітектурі, що схожа з мережею Bitcoin, однак адаптована для завдання пересилання повідомлень, а не грошових транзакцій. Мережа отримала різкий сплеск популярності після розголошення Едварда Сноудена про систему таємного стеження за інтернет-користувачами PRISM.

Filetopia – багатофункціональна файлообмінна програма, головною особливістю якої є високий рівень приватності та безпеки. Підтримується зашифрований чат, повноцінний інтернет-пейджер, робота з форумом. Завдяки технології MS Agent при встановленні відповідного голосового движка можливо голосове відтворення одержуваних повідомлень. З метою підвищення ступеня захищеності Filetopia приховує IP-адреса користувача, захищаючи його тим самим від можливих атак хакерів. В якості алгоритму створення відкритого ключа використовуються еліптичні криві, а повідомлення і файли шифруються одним з десяти алгоритмів, який користувач самостійно обирає.

Freenet – це децентралізована і повністю анонімна однорангова мережа, що працює поверх інтернету, що включає велику кількість рівноправних комп'ютерів і дозволяє публікувати будь-які матеріали без можливості вийти на автора. Конфіденційність даних гарантується суворою криптографією: щоб отримати файл, в запиті потрібно повідомити асоційований із ним ключ. Роль такого ключа виконує хеш-код файлу або DSA-ключ, що утворює також механізм перевірки цілісності. В даний час Freenet починає використовувати принцип мереж Onion Routing.

Пірінгові технології також використовуються багатьма популярними службами миттєвого обміну повідомленнями, наприклад ICQ, і системами

передачі голосових повідомлень із застосуванням пакетних технологій передачі даних, такої як Skype, що використовує власний протокол, або іншими програмами, що працюють на базі протоколу SIP (Session Initiation Protocol).

Технологія P2P знаходить застосування в розподілених обчислювальних мережах. Як приклад, можна привести SETI@home (Search for Extraterrestrial Intelligence) – науковий експеримент, учасники якого займаються пошуком активності позаземного розуму в радіочастотному діапазоні.

На сьогоднішній день в пірингових мережах можна знайти практично будь-яку інформацію абсолютно безкоштовно. Звичайно, все це сприяє активному розвитку файлообміну через P2P-мережі, а кількість даних, що кочують у таких системах, з кожним роком зростає в геометричній прогресії.

Для визначеності у подальшій роботі зосередимося на завданнях передачі в децентралізованих пірингових мережах відеопродукції. Як показують численні дослідження, відеоконтент у цих мережах превалює з перевагою більше 90%.

1.5. Процес пошуку ресурсів в пірингових P2P-мережах

1.5.1. Централізована файлообмінна система Napster

Система Napster з'явилася в 1999 році в якості першої файлообмінної системи P2P. У складі Napster один центральний сервер каталогів і набір зареєстрованих користувачів. Сервер підтримує інформацію про всі файли в системі, включаючи індекс з метаданими (список імен і розмірів файлів, список зареєстрованих пірів).

Коли новий пір приєднується до системи, він зв'язується з сервером і повідомляє список файлів, які він має і готовий ними ділитися. Коли пір хоче знайти потрібний файл, він відправляє запит на сервер. У відповідь сервер надає список вузлів, які містять відповідний файл.

Потім пір, який запустив пошук, зв'язується з пірами зі списку, які містять файл, для скачування файлу.

Рис. 1.12 (а) показує процес пошуку в мережі Napster. Коли пір А хоче знайти який-небудь файл, він зв'язується з центральним сервером. Сервер повертає деякі адреси пірів, які зберігають файл. Скажімо, це пір В. Потім пір А починає скачування файлу від піру В.

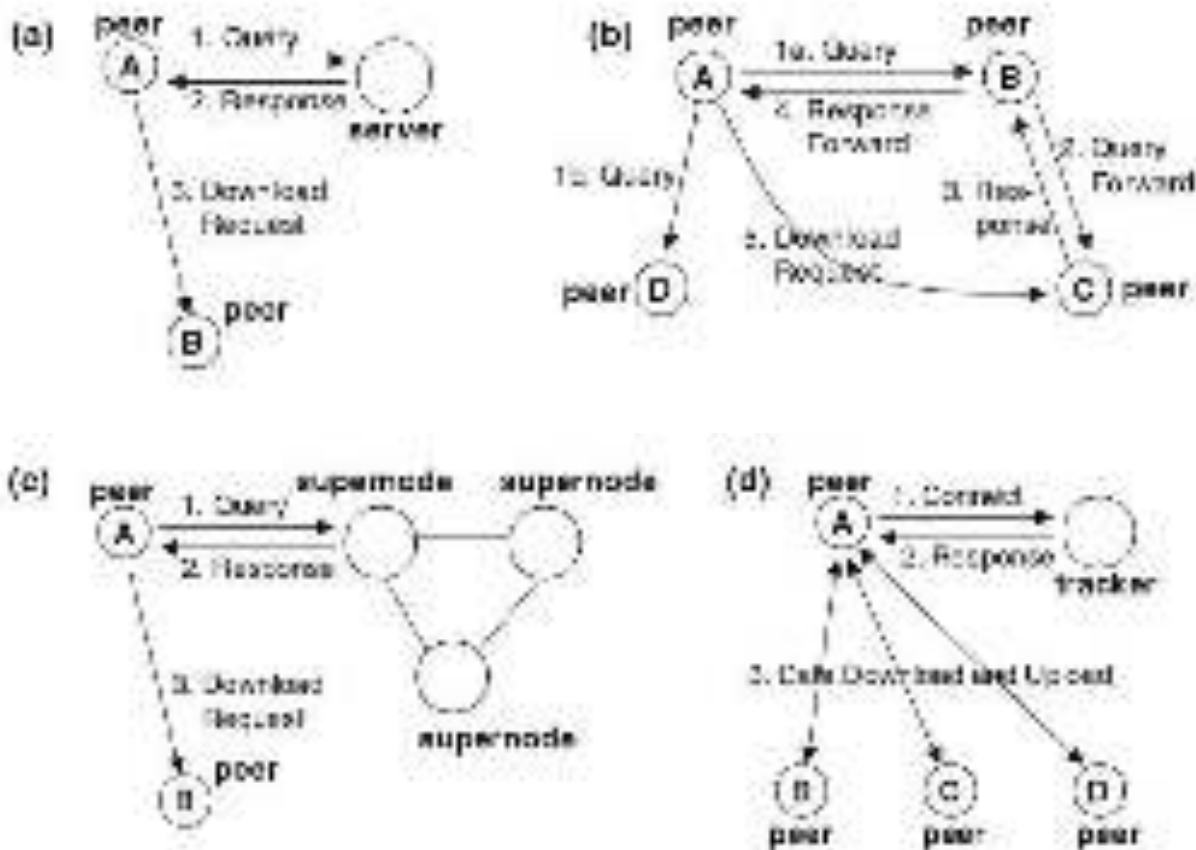


Рис. 1.12. Процес пошуку в неструктурованих P2P мережах. (а) Napster. (b) Gnutella. (c) Kazaa. (d) BitTorrent.

Перевага Napster полягає в легкості його впровадження і простоті розгортання. Системному адміністратору потрібно тільки розгорнути і обслуговувати центральний сервер. Крім того, система має високу адаптивність до підключення і відключення пірів. Основним недоліком є те, що така централізована система не масштабується. Сервер повинен мати багато ресурсів (наприклад, обчислювальні можливості і пропускну здатність) для підтримки великої кількості пірів. Крім того, сервер формує єдину точку відмови. Якщо

сервер не працює, вся система також не працює. Тому він вразливий для цілеспрямованих атак на сервер (наприклад, DDos).

1.5.2. Розподілена децентралізована система: Gnutella

Gnutella – це повністю розподілена P2P-система для обміну файлами [33]. Вперше вона з'явилася в 2000 році й отримала швидкий розвиток через кілька років. Згідно зі звітом Slyck news, станом на червень 2005 року, популяція Gnutella склала 1,81 млн комп'ютерів. Вихідний код Gnutella публічно доступний в Інтернеті. Це дозволяє різними групами розробляти клієнтські програмні продукти (наприклад, LimeWire, розроблений групою LimeWire, або Gnucleus, розроблений групою Morpheus).

У базовому протоколі Gnutella, коли новий пір підключається до системи, він спочатку підключається до деяких загальнодоступних пірів. Наприклад, список загальнодоступних пірів доступний за адресою <http://gnutellahosts.com>. Потім новий пір відправляє повідомлення PING будь-якому піру, до якого він підключений. Повідомлення інформує про існування нового піру. Отримавши повідомлення PING, пір Gnutella повертає повідомлення PONG і передає повідомлення PING своїм сусідам. Повідомлення PONG містить IP-адресу і порт відповідає піру, а також інформацію про файли, які пір, що відповідає, спільно використовує. У динамічній мережі з частим підключенням і відключенням пірів пір періодично відправляє повідомлення PING своїм сусідам.

Пошук в Gnutella заснований на лавинній маршрутизації, яка транслюється в оверлеї. Пошуковий запит передається всім сусідам від вихідного запитувача піру. Запит репліцюється і пересилається кожним проміжним вузлом всім своїм сусідам. Кожен проміжний вузол також досліджує свій локальний вміст і відповідає на джерело запиту на основі подібності. Відповіді на запит перенаправляються назад протилежним шляхом до вихідного піру, що відправив запит. Щоб зменшити кількість запитів у мережі, кожне повідомлення запиту

містить поле часу життя (TTL). Значення TTL зменшується на одне в кожному вузлі. Коли воно досягає нуля, повідомлення буде видалено.

Рис. 1.12 (b) показує процес пошуку в Gnutella. Припустимо, що пір А хоче знайти якийсь файл. Він розсилає запит своїм сусідам, тобто пірам В і D на рисунку. Коли пір В отримує запит, він перевіряє, чи міститься у нього відповідний файл. Якщо немає, він перенаправляє запит своїм сусідам. Як і в прикладі, пір В відправляє запит свого сусіда С. Припустимо, що С містить файл, який потрібен піру А. Пір С повертає відповідь вузла, що відправив запит, тобто піру В на рисунку. Пір В потім продовжує перенаправляти відповідь відправнику А. Нарешті, пір А зв'язується з піром С, щоб завантажити файл.

На відміну від Napster, Gnutella – це динамічна, самоорганізована мережа. Кожен пір незалежно підключається і взаємодіє з декількома іншими пірами в системі. Таким чином, система може містити необмежену кількість пірів, якщо немає обмежень на ефективність пошуку. Тим часом, система має високу стійкість до динаміки пірів. Якщо пір залишає систему, його сусіди можуть підключитися до інших пірів за допомогою обміну повідомленнями PING і PONG.

Обмеженням Gnutella є відносно низька ефективність пошуку. У потоковому пошуку число повідомлень запиту експоненціально збільшується з кількістю хопів в оверлеї. Тоді запит може генерувати багато повідомлень, особливо для непопулярних файлів, де запит повинен пройти через багато хопів в оверлеї і безліч пірів до досягнення відповідного піру. З огляду на величезну кількість пірів в системі, завантаження трафіку для запитів може виявитися надзвичайно високим. Використання TTL може скоротити кількість запитів. Вибір відповідного TTL має компроміси. Якщо значення TTL занадто велике, то піри надмірно завантажують мережу. Якщо значення TTL занадто низька, пір може не встигнути знайти файл, навіть якщо там є копія.

Крім того, в потоковому пошуку є багато повторюваних повідомлень, особливо в сильно пов'язаних мережах. Це відбувається тому, що пір може відправити кілька копій запиту сусіднім пірам. Дані запити, що дублюються,

споживають додатковий обчислювальний ресурс пірів та пропускну здатність мережі. Отже, необхідно розробити деякі механізми виявлення дублювання. Проте, навіть при дублюванні, кількість повторюваних повідомлень при лавинній маршрутизації може бути надмірною, і проблема посилюється зі збільшенням TTL.

Кільце, що розширюється, не усуває проблему дублювання повідомлень. Інший метод використовує множинні паралельні випадкові блукання для вирішення цієї проблеми. У стандартному випадковому блуканні при отриманні пошукового запиту пір випадково вибирає одного сусіда і передає йому запит. Щоб зменшити час пошуку, метод використовує кілька випадкових блукань. Цей метод, поряд з активною реплікацією файлів, може значно поліпшити ефективність пошуку з точки зору часу пошуку, завантаження пірів запитами між пірами і трафіком повідомлень.

У новій мережі Gnutella-2 з'явиться можливість розмноження інформації про фото без його реплікації, що полегшить відстеження вірусів. Для переданих пакетів у мережі розроблений власний формат, що чітко реалізує можливість нарощування функціональності мережі шляхом додавання додаткової службової інформації. Запити та списки ID ключових слів пересилаються на концентратори UDP.

1.5.3. Гібридна система: FastTrack/Kazaa

З огляду на обмеження чисто централізованих мереж і чисто розподілених мереж, існує третій підхід, який об'єднує ці два типи мереж. FastTrack – типовий приклад частково централізованого протоколу P2P. У FastTrack піри з швидким підключенням до Інтернету і найпотужнішими комп'ютерами автоматично позначаються як супервузли. Супервузол зберігає інформацію про деякі ресурси, а також про зв'язки з іншим супервузлом. Коли пір виконує пошук, він спочатку шукає найближчий супервузол, який негайно повертає результати, якщо такі є, і, якщо необхідно, посилається на пошук інших суперпірів. Два практичних

програмних забезпечення на основі FastTrack – це Kazaa і Grokster. Але останнє закрилося у 2005 році через проблеми з авторським правом.

Рис. 1.12 (с) показує процес пошуку в Kazaa. Коли пір А хоче знайти якийсь файл, він відправляє запит найближчому супервузлу. Супервузол або повертає деякі відповідні піри, або пересилає запит іншим супервузлом. Нарешті, пір А отримує деякі відповідні піри від супервузла (скажімо, пір В на рисунку) і завантажує файл з цих вузлів.

Тому звичайний пір (наприклад, пір А на рисунку) спілкується з супервузлом, як ніби він взаємодівав би з сервером в Napster. Потім Gnutella, як пошук, виконується в оверлейній мережі супервузлів з відрубаними гілками дерева.

У порівнянні з чисто розподіленими мережами, такими як Gnutella, Kazaa досягає набагато меншого часу пошуку. Пошук серед супервузлів відбувається набагато швидше, ніж пошук серед усіх пірів, оскільки число супервузлів набагато менше, ніж загальне число пірів. Оскільки супервузли володіють високою пропускною здатністю і великим об'ємом пам'яті, вони можуть ефективно обробляти велику кількість запитів від звичайних пірів. Таким чином, система ефективно використовує гетерогенність пірів. Крім того, на відміну від Napster, це ПЗ не утворює єдиної точки відмови. Якщо деякі суперпіри зависають, підключені до них піри можуть підключатися до інших супервузлів.

З огляду на успіх Kazaa, Gnutella також розглядає можливість використання гібридної структури. Chawathe et al. запропонували в системі Gia поліпшити Gnutella. Тут використовується протокол адаптації динамічної топології для розміщення більшості пірів на невеликій відстані від декількох супервузлів. Супервузли отримуватимуть більшу частину запитів. Разом із деякими іншими поліпшеннями (наприклад, активний контроль потоку і зміщене випадкове блуканням (biased random walk)), Gia збільшує продуктивність системи на три-п'ять порядків. Пізніше Gnutella версії 0.6 формально включає ідею «ультрапірів», тобто піри, що входять в мережу, залишаються на краю мережі у вигляді листів, які не відповідають за будь-яку маршрутизацію. Браузери, які здатні

маршрутизовати повідомлення, просуваються на ультрапіри, які будуть приймати з'єднання листів, і пошук маршруту, а також повідомлення про обслуговування мережі. Як правило, листовий вузол підключається до 3 ультрапірів, а кожен ультрапір підключається до більш ніж 32 інших ультрапірів. В межах цієї мережі максимальна кількість переходів, через які може переміщатися запит, зменшується приблизно до 4. Ефективність пошуку і масштабованість системи в цілому значно поліпшуються.

1.5.4. BitTorrent

BitTorrent – це система P2P, яка не належить ні до однієї з перерахованих вище категорій [34]. У 2004 році CacheLogic оцінила BitTorrent як важливий додаток для обміну файлами P2P, на який припадає 35% всього інтернет-трафіку.

BitTorrent використовує центральне розташування для координації завантаження і скачування даних між пірами. Для спільного використання файла f спочатку створюється невеликий торрент-файл, який містить метадані про f , наприклад його довжину, ім'я та інформацію про хешування. Зазвичай BitTorrent розбиває файл на частини фіксованого розміру, зазвичай від 64 Кб до 4 Мб. Кожна частина має контрольну суму з алгоритму хешування SHA1, яка також записується в файл торрента. Найголовніше, торрент-файл містить URL-адресу трекера, який відстежує всіх пірів, у яких є файл f (частково або повністю), і запитують піри.

Пір, який хоче завантажити файл, спочатку отримує відповідний файл торрента, а потім підключається до зазначеного трекера. Трекер відповідає списком випадкових пірів, які завантажують один і той же файл. Потім запитуючий пір з'єднується з цими пірами для завантаження.

Рис. 1.12 (d) показує процес пошуку в централізованій структурі BitTorrent. Коли пір А хоче знайти якийсь файл, йому спочатку потрібно отримати відповідний торрент для файлу. За допомогою торрента, А знає адресу трекера і

підключається до нього. Потім трекер повертає список пірів, які завантажують або обмінюються файлом. Потім А обмінюється даними з цими пірами.

У системах BitTorrent торрент-файли часто публікуються на великих сайтах, які також служать в якості трекерів. Очевидно, що централізація трекерів вносить деякі бар'єри в систему. Якщо трекер не працює, піри не зможуть почати спільне використання (завантажуючи торренти в трекер), а нові вхідні піри не зможуть почати скачування. Щоб позбутися від необхідності використання центральних трекерів, останні клієнти BitTorrent впроваджують децентралізований механізм відстеження.

1.6. Порівняльна характеристика процесів пошуку в неструктурованих і структурованих P2P-мереж

У системах P2P взаємодіючі піри самоорганізуються в оверлейні мережі і зберігають або ретранслюють дані один для одного. Основна проблема полягає в тому, як домогтися ефективного пошуку ресурсів у великомасштабній розподіленій мережі зберігання даних.

1.6.1. Порівняння і обговорення методів пошуку файлів

Неструктуровані мережі P2P мають свої власні обмеження. Їх головна проблема – низька ефективність пошуку, особливо для непопулярних файлів. Непопулярні файли мають мало копій у системі. Пошук такого файлу може привести до великомасштабної лавинної маршрутизації. На відміну від цього, структуровані DHT-мережі гарантують певну кількість хопів оверлеїв для будь-якого пошуку, що сприяє пошуку непопулярних файлів. Беручи до уваги переваги структурованих і неструктурованих мереж, деякі дослідники докладають зусиль для їх інтеграції [11]. Очевидно, що в інтеграції є багато проблем через їх принципову різницю в мережевих структурах.

Головною характеристикою неструктурованих мереж P2P є те, що зберігання файлів повністю не пов'язане з топологією оверлею. В результаті механізм пошуку файлів в таких мережах, по суті, зводиться до випадкового пошуку. У порівнянні зі структурованими мережами P2P, неструктуровані мережі мають такі переваги:

- Динамічність, відмовостійкість: оскільки в структурі оверлею немає певної вимоги, неструктуровані мережі P2P часто формуються випадковим чином. У разі приєднання і від'єднання пірів, неструктурована мережа P2P може легко відновити оверлей. Для порівняння, реконструкція оверлею в мережах DHT набагато складніша і дорожча, особливо в високо динамічній мережі з частим підключенням і відключенням.

- Підтримка складного пошуку: в неструктурованих мережах P2P піри в кінцевому рахунку перевіряють свій локальний ресурс, щоб відповісти на запит. Отже, неструктуровані мережі P2P за своєю суттю підтримують складний пошук на основі метаданих файлів. Навпаки, в мережах DHT кожен файл має одне ключове слово. Сховище файлів і пошук засновані на цьому ключовому слові. DHT, отже, підтримує пошук тільки по одному ключовому слову.

Централізований підхід вимагає наявності центрального сервера для управління файлами і пошуком, а підхід на базі супервузлів заснований на обраних або попередньо розгорнутих супервузлах. На відміну від них, повністю розподілений підхід покладається на піри для зберігання і пошуку файлів, що не вимагає додаткових коштів. BitTorrent вимагає, щоб центральні трекери ініціювали процес спільного доступу до файлів.

У відповідності з архітектурою системи мають місце різні методи і підходи до пошуку. У централізованому підході запит безпосередньо відправляється на сервер. У підході, заснованому на супервузлі, запит спочатку направляється супервузлом, який пересилає запит іншим супервузлам, якщо це необхідно. У повністю розподіленому підході сліпа або інформована маршрутизація використовується для відповіді на запит, який може споживати велику пропускну здатність мережі. У BitTorrent пір може безпосередньо отримати список пірів, що

спільно використовують файл з трекера. Отже, це виключає процес пошуку потрібних пірів, як в інших системах. Це фундаментальна відмінність BitTorrent від інших неструктурованих P2P-систем.

На практиці підхід на основі супервузлів і BitTorrent є найбільш успішним додатком. Як централізовані, так і повністю розподілені підходи мають серйозні обмеження. Централізований підхід має погану масштабованість. Він не може вмістити велику кількість пірів. Але величезна кількість пірів, і, отже, величезний ресурс в цілому, як раз і є найбільш привабливим аспектом систем P2P. Повністю розподілений підхід не потребує будь-якого центрального компонента, однак він мало масштабується через його неефективність і сильне споживання смуги пропускання під час пошуку.

Разом з тим, підхід, заснований на супервузлі, показав високу масштабованість і високу ефективність пошуку при достатній кількості супервузлів. Він дозволяє використовувати різноманітність пірів. Хоча для BitTorrent потрібні центральні трекери, відповідальність трекера набагато простіша, ніж сервера в централізованому підході. Трекеру потрібно тільки відслідковувати спільне використання та завантаження конкретного файлу. Торрент-файли можуть бути поміщені в різні трекери. Отже, масштабованість BitTorrent не є проблемою на практиці. Крім того, новий децентралізований механізм відстеження повністю виключає центральні трекери з системи.

1.7. Висновки до першого розділу

1. Інформаційний обмін в сучасному суспільстві, який здійснюється за допомогою однорангових пірингових мереж, є вельми популярним видом спілкування, бізнесу і розваг, тому задачі щодо підвищення якості послуг у файлообмінних мережах представляються актуальними, мають наукову і практичну значимість.

2. Пірингові P2P мережі – це накладені, що функціонують як оверлейні по відношенню до мереж Інтернет, однорангові мережі передачі, за допомогою яких

можна обмінюватися файлами будь-якого розміру, що досягає десятків гігабайт. Одночасно з цим, розподілений їх характер дозволяє задіяти віддалені комп'ютери користувачів для виконання складної обробки даних. Важливою властивістю є те, що при створенні пірингових мереж відсутні витрати на створення мережевої інфраструктури (кабелі, передавачі, студії та ін.). Їх застосування базується на вже існуючих Інтернет-ресурсах, утворюючи оверлейні структури. Провайдери при цьому пропонують продукти вторинного ринку (аудіо, відео, розваги).

3. На відміну від мереж Інтернету, що підтримують клієнт-серверну архітектуру, в основі P2P-мереж лежить принцип рівноправності, тобто файли можна не тільки завантажувати, але обов'язково ділитися ними з іншими.

4. Існує два основні підходи до розсилки споживачам відеоконтенту: з використанням мережевої інфраструктури по мережах доставки CDN і з використанням технології Інтернет-піринга, де ця інфраструктура не використовується. Мережі доставки контенту CDN, OTT (IPTV) забезпечують доставку статичного відео-контенту, в той час як P2P-мережі - забезпечують динамічне потокове відео, надання якого не залежить від початку самої події.

5. У світовій практиці використовується велика кількість різних файлообмінних пірингових мереж, серед яких KaZaA, Gnutella, eDonkey, BitTorrent, PPLive, SopCast та ін. В пірингових мережах можна знайти практично будь-яку необхідну інформацію: музика, відеокліпи, фільми будь-якої якості, ігри та ін. При цьому відеопродукція становить до 90% обсягу трафіку цих мереж. З огляду на цей факт, технології пошуку, відбору, формування файлів та їх доставки споживачам будемо аналізувати на тлі передачі відеоданих.

6. Відмітною властивістю пірингових мереж від мереж клієнт-серверної архітектури є те, що зі збільшенням чисельності активних ділянок P2P-мережі якість і надійність надання послуг – зростає, в той час як у клієнт-серверних – падає.

7. Пірингові мережі можуть бути організовані по централізованому, децентралізованому або гібридному принципу. Незважаючи на те, що централізовані мережі простіші, більш організовані і продуктивні, більш

перспективними представляються децентралізовані і гібридні мережі, які функціонують без індексаційних серверів, що надає їм більшу надійність і відмовостійкість. Децентралізація забезпечує більшу масштабованість, динамічність, здатність до самоорганізації. Одночасно з цим пірингові мережі є незахищеними від кібератак, від поширення неліцензійних продуктів.

8. При побудові пірингових мереж зазвичай використовуються схеми двох видів: деревовидні (tree-push) з кодуванням субпотоків і мережеві (mesh-pull), у яких потік даних ділиться на модулі (фрагменти). Mesh-технології виявляються більш динамічними, узгодженими з динамікою P2P-мережі.

9. Незважаючи на широку популярність файлообмінних пірингових мереж в даний час залишаються недослідженими цілий ряд алгоритмів, структур і технологій, що використовуються в цих мережах. До числа найбільш значущих, від яких залежить успішний обмін інформаційними файлами є ряд завдань управління розподілом пірів та відповідного контенту управління скачуваними файлами і фрагментами. Таким чином, в якості наукової задачі в дисертації слід прийняти таке формулювання: розробка математичної моделі інформаційної послідовності в файлообмінних мережах і оптимальна її фрагментація.

РОЗДІЛ 2

АНАЛІЗ АРХІТЕКТУРИ ПІРИНГОВИХ P2P МЕРЕЖ

Архітектура P2P (Peer-to-Peer), на відміну від традиційних клієнт-серверних архітектур, передбачає можливість одного і того ж піру виступати як в якості клієнта (ліч – той, що скачує), так і в якості сервера (сід – той, що роздає). Іншими словами: піри не тільки скачують файли, які їх цікавлять, але і роздають їх іншим користувачам в мережі.

У порівнянні з системами multicast IPTV-мовлення, Over-the-Top (OTT-TV) і відео за запитом, що поширюється в мережах доставки контенту (Content Delivery Network - CDN), технологія пірингового зв'язку має перевагу в тому, що користувач може мати послугу TV-перегляду повністю через будь-який час після початку, при цьому буфер великого розміру не потрібен. Таким чином, P2P-технологія представляється як TV-технологія, інваріантна до початку. При цьому слід виконати п'ять основних дій: 1. Приєднання/від'єднання піру, з яким здійснюється обмін файлами 2. Процес обміну файлами 3. Вибір піру, який володіє потрібними файлами 4. Організація буфера 5. Планування сегментів.

Розглянемо більш детально зміст зазначених стадій.

Цей набір пірів та зв'язків між ними встановлює P2P оверлей – відповідну програму, що керує підключеннями користувачів (пірів).

2.1. Склад і основні функції елементів P2P-мережі

2.1.1. Загальні відомості про архітектуру P2P-мереж

Мережі P2P є децентралізованими розподіленими системами, що дозволяють учасникам мережі спільно використовувати й інтегрувати свої обчислювальні ресурси, дані та послуги. Хоча концепції P2P і Grids значно перетинаються, спочатку вони були запропоновані для різних областей. P2P

завичай застосовується до широкого спектру технологій, які можуть значно збільшити використання колективних ресурсів на краю Інтернету, таких як інформація, пропускна здатність і обчислювальні ресурси, в той час як грид-обчислення призначені для забезпечення сумісності й розширюваності серед різних додатків, платформ і структур [44].

На відміну від існуючих Grid-парадигм, архітектура P2P не покладається на централізований сервер для надання сервісів (рис. 2.1), що пропонує альтернативу існуючим Grid-моделям, особливо для великомасштабних розподілених додатків. У моделі P2P кожен спеціальний робочий вузол (також відомий як клієнт – servant) виступає в ролі клієнта і сервера, запитуючи ресурси, а також направляючи запити й обслуговуючи ресурси інших тимчасових вузлів. Мережа P2P являє собою логічну оверлейную мережу поверх фізичної інфраструктури, як показано на рис. 2.1, яка забезпечує віртуальне середовище для розробників P2P.

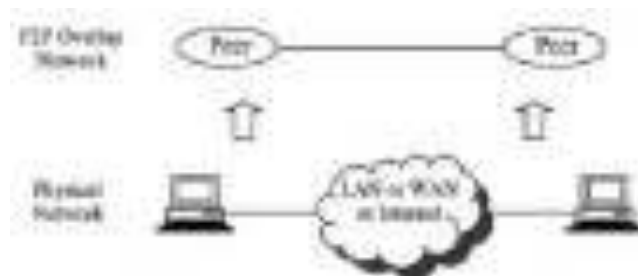


Рис. 2.1. Архітектура Peer-to-peer

Популярність і використання мереж P2P в останні роки зростають в експоненційному ступені. Спільне використання файлів P2P стало одним із найпопулярніших видів діяльності в Інтернеті. Сьогоднішні популярні додатки для обміну файлами P2P надають такі мережі як KaZaA і Gnutella. Вони обслуговують у кожен момент часу більше мільйона споживачів. За результатами досліджень Cachelogic, близько (50-65)% «низхідного трафіку» (тобто, від провайдерів до кінцевих пристроїв) і (75-90)% «висхідного трафіку» (тобто, від

кінцевих пристроїв до інтернет-провайдерів) в Інтернеті – це результат роботи додатків обміну файлами P2P [45].

Популярність P2P мотивується вигодами, які вона пропонує кінцевим користувачам. Переваги P2P в порівнянні з архітектурою «клієнт-сервер» перераховані нижче:

- P2P звільняє користувачів від традиційної залежності від центральних серверів, що дозволяє кінцевим користувачам легко обмінюватися ресурсами (наприклад, музикою, фільмами, іграми та іншим програмним забезпеченням). Кінцеві користувачі можуть обмінюватися або отримувати ресурси безпосередньо з підключених комп'ютерів без необхідності завантажувати їх на централізований сервер.

- При видаленні однієї точки відмови, додатки P2P більш стійкі, ніж ті, які побудовані на архітектурі клієнт-сервер.

- У P2P відповідальність за надання послуг перерозподіляється з централізованих серверів кожному окремому піру в мережі.

- P2P використовує доступну пропускну здатність, процесор, сховище та інші ресурси по всій мережі. Взаємодія P2P відбувається тільки між окремими пірами. При цьому вузьке місце – централізований сервер – участі не бере.

- Більшість додатків P2P використовують віртуальні канали для зв'язку, які долають перешкоди корпоративних приватних мереж, такі як міжмережеві екрани і трансляція мережевих адрес (Network Address Translation - NAT).

- P2P має кращу доступність, оскільки кожен спеціальний робочий вузол може отримувати контент із декількох тимчасових вузлів. Якщо один спеціальний робочий вузол перевантажений, або має апаратний збій, обробляти запити в мережі можуть інші однорангові вузли.

2.1.2. Порядок взаємодії елементів в P2P-мережі

У складі P2P мережі крім ліч- і сід-вузлів можуть бути присутніми сервер і трекер (рис. 2.2).

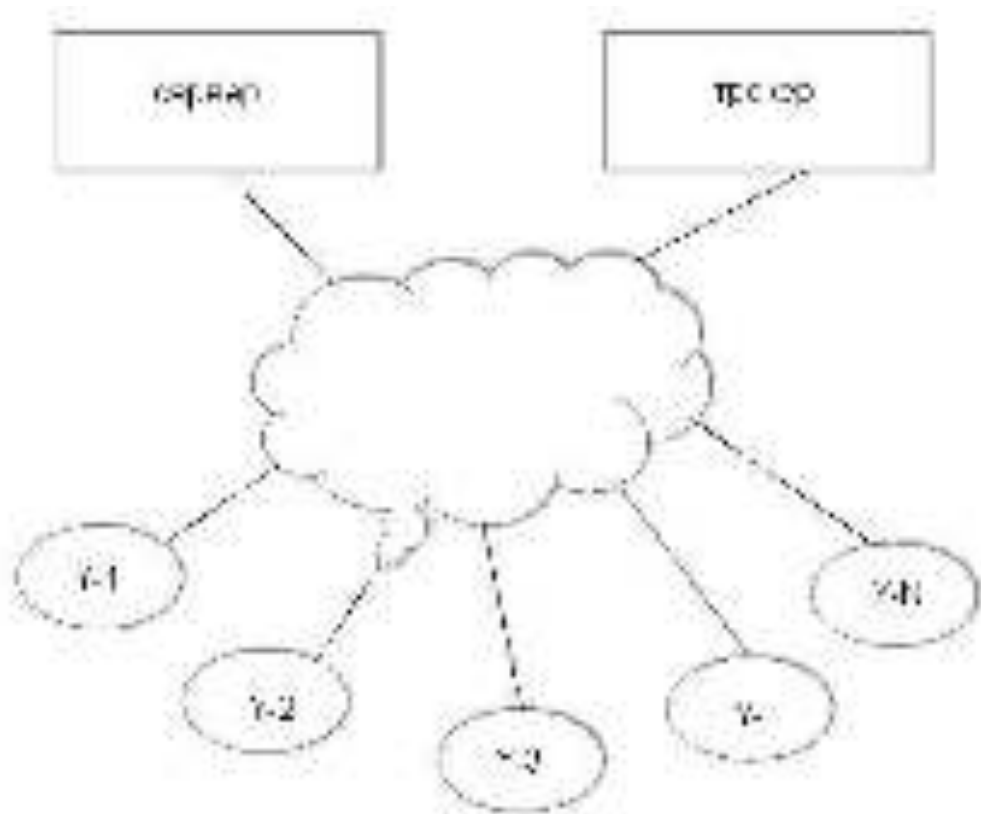


Рис. 2.2. Модель телевізійної P2P-мережі

Процес приєднання нового вузла Y_i до мережі починається зі зв'язку його з трекером. Трекер відомий всім вузлам і підтримує список всіх вузлів в мережі.

Трекер надсилає поточний список пірів новому вузлу. Новий вузол обмінюється буферною картою зі списками пірів та вибирає піра – сусіда, щоб завантажити від нього потокове відео.

Сервер – це вузол, який надає всі фрагменти запитуваного відеопотоку зацікавленому споживачеві в реальному часі.

Розглянемо зміст 5-ї дії мережі при наданні послуги споживачеві.

1. Приєднання/від'єднання піру. Будь-який із пірів може входити і виходити з мережі в будь-який час. Новий пір, який долучився до мережі, посилає повідомлення із запитом до трекера. Трекер додає пір до списку і готує список вузлів, які переглядають потрібний потік. Після отримання списку пірів новий пір обмінюється буферними картами з усіма пірами. Новий пір може вибрати пір з декількох буферних карт інших пірів та запросити фрагменти відео.

2. Обмін інформацією між пірами: пір повинен знати про наявність фрагментів у всіх доступних пірів. Ця інформація передається співвідношеннями BitTorrent – подібних буферних карт (BM), а також даними HAVE (маю) або DO NOT HAVE (не маю). BM використовуються для інформування сусідніх вузлів про видалення фрагментів. Спочатку, новий вузол створює з'єднання з усіма вузлами, про які повідомляє трекер і запитує буферні карти цих вузлів. Відповідні BM містять інформацію про всі фрагменти, які доступні на передавальному вузлі та які зберігаються на вузлі-приймачі. Коли вузол завантажив новий фрагмент або йому довелося фрагмент видалити (через обмежений розміру буфера), він інформує всі підключені вузли, відправивши повідомлення HAVE або DONTHAVE. Використання повідомлень HAVE і DONTHAVE дозволяє уникнути втрати пакетів. Кожне з цих повідомлень містить номер нового або видаленого фрагмента. Повідомлення використовуються усіма приймачами для поновлення BM на передавальному вузлі.

3. Вибір піру: Коли новий пір отримує інформацію про буферні карти інших пірів, він може вибрати деякі з них, щоб завантажити частини потоку. Вузол шукає всі відомі піри, які мають необхідний фрагмент і вибирає джерело випадковим чином.

4. Організація буфера: Для мовлення кожен вузол повинен мати буфер для зберігання відеофрагментів. Довжина буфера може бути меншою, ніж файл, що передається. Буфер кожного вузла складається з трьох частин: буфер даних, буферна карта і змінне вікно. Буфер даних використовується для зберігання відео кадрів. Буферна карта – це бітовий вектор, що представляє інформацію про доступні сегменти на вузлі. Кожен вузол періодично обмінюється своєю буферною картою зі своїми сусідами. Виходячи з цієї інформації, буфер може вирішити, які партнерські вузли використовуються для виклику з пам'яті необхідних сегментів. Якщо один і той же сегмент має кілька партнерів, вузол-пір може вибирати партнерів з мінімальною затримкою, максимальною пропускну здатністю або випадковим чином. Крім буферної карти, кожному вузлу необхідно мати змінне вікно для зберігання кількох показаних фрагментів. Відповідно до

даної організації буфера відеосегменти будуть показані послідовно, а початкова затримка кожного вузла буде обмежена.

У ряді мереж використовується циклічний буфер. Буфер даних ділиться на три частини: оглядовий буфер, буфер відображення і майбутній буфер.

Оглядовий буфер використовується для старих фрагментів, для буферизації потоку даних у певний період часу перед програванням потоку. Кількість кадрів в оглядовому буфері розраховується з затримки, званої затримкою перегляду між піром-відправником і піром-одержувачем. Затримка перегляду між кожним піром – випадкова, тому що використовується mesh-архітектура. Для простоти дана затримка визначається як максимальна межа в цій групі користувачів у цій конкретній роботі. Таким чином, всі піри матимуть однакову затримку.

Буфер відображення (свіжі фрагменти) використовується для зберігання даних, які будуть показані користувачам. Цей буфер проектується при використанні змінного вікна. Кадр на початку буфера – це кадр відображення, а наступний кадр – це наступний кадр у вікні, який буде показаний через кілька хвилин.

Майбутній буфер (майбутні фрагменти) використовується для прийому нових кадрів. Нові кадри, отримані від інших пірів або партнерів за допомогою послідовного планування або планування "найрідкісніший-перший".

5. Планування сегментів: Пір-клієнт вибирає деякі піри для обміну сегментами. Отже, спочатку потрібно передати інформацію про доступні сегменти, а потім вибрати одного (або більше) пір-клієнта як джерело цього фрагмента на основі інформації про доступні сегменти в цих пірах. Важливою інформацією для розрахунку планування фрагмента є час відтворення. Якщо частина відеопотоку вже програна, фрагменти, які повинні бути відтворені далі, повинні стати доступними в локальному буфері. Два з найбільш важливих понять тут – це послідовне скачування і скачування "найрідкісніший-перший". Послідовне скачування завжди вибирає фрагмент, який знаходиться найближче за часом відтворення. "Найрідкісніший-перший" завантажує фрагмент, який доступний найменшій кількості пірів-клієнтів і займає більше часу на

завантаження. Як правило, це найновіший сегмент. У цій технології послідовне скачування використовується для завантаження сегмента кожного партнера. Може бути створено кілька потоків, щоб завантажити інші сегменти від кожного піру.

Вже згадана архітектура (рис.2.2) виявляється чутливою до збільшення трафіку, що призводить до сильного завантаження трекера, а при віддалених з'єднаннях зростають затримки, що позначається на якості надання послуг. Для вирішення цих проблем використовують кластеризацію угруповань пірів за локальною ознакою.

Отримана при цьому ієрархічна архітектура (CBT) забезпечується двома додатковими алгоритмами:

- алгоритмом приєднання піру;
- алгоритмом вибору існуючого піру.

При відборі пірів для кластера використовуються тимчасові критерії RTT – час кругового обертання і TTL – термін до закінчення часу життя пакета.

Кластерна ієрархічна архітектура CBT покращує продуктивність і масштабованість оверлейних P2P-мереж. Відомі три протоколи вибору супер-вузла: SOLE, PoPCorn, H2O [47]. Розглянемо більш докладно кластерну архітектуру P2P.

2.2. Архітектура структурованої системи на основі кластерів

Кластеризація – групування декількох близько розташованих пірів, використовується при збільшенні навантаження. Кластерна система складається з сервера, глобального трекера (GT), супер-вузла (SN) або локального трекера (LT), резервного вузла (BN) і звичайних вузлів (NN). Структурована (кластерна) модель представлена на рис. 2.3. Коли новий вузол приєднується і хоче завантажити новий фрагмент від пірів в кластерній моделі, глобальний трекер повинен вирішувати, який кластер або супер-вузол повинен бути підключений.

Сервер – це вузол, який розподіляє всі фрагменти живого відеопотоку. Глобальний трекер відомий усім вузлам, він підтримує список всіх супер-вузлів. Супервузол виступає в ролі локального трекера, зберігаючи список вузлів у кластері. Всі супервузли пов'язані з глобальним трекером для синхронізації списку всіх вузлів в кластері.

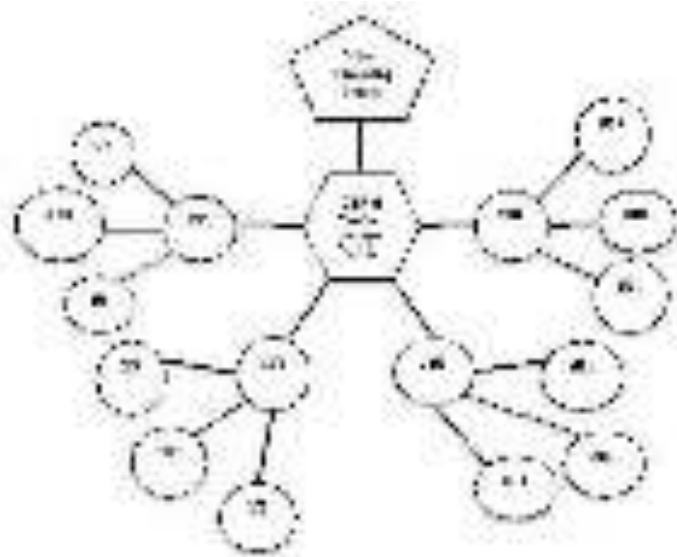


Рис. 2.3. Структурована (кластерна) модель для телевізійного мовлення

Супервузол, звичайний вузол і резервний вузол – скачують (upload) і завантажують (download). Кластер формується виходячи з близькості пірів та ця близькість вимірюється часом приєднання кожного вузла або доступністю першого фрагмента. Кластеризація використовується для контролю потоків трафіку в межах P2P системи, а також додатково допомагає знизити навантаження на сервер і глобальний трекер. Грунтуючись на некластерній моделі для мовлення відеоінваріантного до початку, поведінка і алгоритми (обмін інформацією між пірами, вибір піру, організація буфера, планування сегментів) вузлів у кластерній структурі не змінюються, але продовжуються логічним механізмом кластеризації. Кластеризація реалізується за допомогою поділу вузлів, супервузлів, резервних вузлів і логічних трекерів.

Процедура кластеризації піру складається з 8 наступних стадій: (1) приєднання піра, (2) вибір супер-вузла, (3) вибір резервного вузла, (4) відключення піра, (5) обмін інформацією між пірами, (6) вибір піра, (7) організація буфера, (8) планування сегментів. Розглянемо дані стадії:

1. Приєднання піру: Коли приєднується новий пір, він зв'язується з глобальним трекером для запиту кластера і супер-вузла. Таким чином, алгоритм приєднання піру складається з 2-х фаз: з'єднання з глобальним трекером і з'єднання з локальним трекером. У першій фазі всі вузли знають адресу глобального трекера. Коли новий вузол зв'язується з глобальним трекером і запитує перший фрагмент, глобальний трекер, у свою чергу, зв'яжеться з локальним трекером (LT) кожного кластера для пошуку вузлів, що мають перший доступний фрагмент. Глобальний трекер потім вибирає кластер, який має максимальну кількість вузлів, що містять перший фрагмент. Новий вузол отримує адресу LT і там реєструється. У другій фазі новий вузол зв'язується з LT. LT у відповідь відправляє випадковий список сусідніх пірів у тому ж кластері. Новий вузол отримує цей список сусідніх пірів та потім відправляє повідомлення для обміну буферними картами з сусідніми пірами. Новий вузол відправляє запит сусіднім пірам для скачування фрагментів.

2. Вибір супер-вузла: Після приєднання першого вузла, глобальний трекер призначає цей вузол супервузлом (SN) або локальним трекером для першого кластера. Розмір кластера обмежується до C вузлів. Після того, як глобальний трекер отримав повідомлення про з'єднання від нового вузла, він перевіряє розмір для вибору найбільш відповідного кластера. Глобальний трекер перевіряє кількість членів обраного кластера. Якщо чисельність обраного кластера менше, ніж C , адреса SN даного кластера буде відправлений підключому вузлу. Якщо розмір вибраного кластера дорівнює C (повний), створюється новий кластер. Глобальний трекер відправляє вузол, який містить перший доступний фрагмент, в старий кластер, щоб він був SN в новому кластері. Якщо в системі кластера немає, створюється перший кластер, а перший вузол, що підключається, буде SN в першому кластері.

3. Вибір резервного вузла (BN): Якщо кластер має повний розмір (дорівнює C), BN буде обраний серед усіх звичайних вузлів. Резервний вузол зберігає список усіх пірів в кластері за контактами з SN. Коли SN виходить з кластера, BN стає новим SN. BN отримує список усіх пірів в кластері від SN. BN може бути обраний трьома різними способами:

- вибрати вузол, який приєднується до кластеру після першого вузла (другий вузол, що з'єднується, після другого);
- вибрати вузол, який приєднується до середини групи (С-й 2 вузол);
- вибрати останній вузол, який приєднується до групи (С-й вузол).

У першому способі всі вузли в кластері матимуть однаковий шанс бути SN і BN. Недоліком цього підходу є частий вибір SN і BN. При другому методі нові SN і BN вибираються не часто і працюють добре. Третій спосіб вибирає новий супервузол не часто, але може привести до втрати пакетів.

4. Вузол, що відключається: Якщо вузол виходить з кластера, локальний трекер видаляє його зі списку пірів. Якщо вузол, що відключається, є локальним трекером (LT), резервний вузол буде новим локальним трекером. Якщо останній вузол залишає кластер, кластер буде видалений. Локальний трекер завжди повідомляє глобальному трекеру про вузли, що відключаються, для синхронізації списку SN в глобальному трекері. Процес відключення вузла можна розділити на три випадки: вихід SN, BN і NN. У першому випадку, коли SN залишає кластер, він посилає лавинне повідомлення всім вузлам. SN посилає повідомлення глобальному трекеру. У другому випадку BN періодично обмінюється інформацією з SN. Якщо BN виходить, він посилає повідомлення, щоб проінформувати супер-вузол. В останньому випадку NN може вийти з мережі в будь-який час. Обмін інформацією між пірами, вибір пірів, організація буфера і планування сегментів кластерної моделі схожі на некластерну модель.

Цей набір пірів та зв'язків між ними встановлює P2P оверлей – відповідна програма, що керує підключеннями. Якщо розглянутий пір є піром-джерелом, тобто, якщо він хоче передати відеопотік в P2P оверлей, вхідний відеосигнал повинен бути перетворений в послідовність фрагментів, тобто в самостійну

одиницю даних, яка використовується для обміну в P2P оверлеї. Залежно від типу джерела відеосигналу це може бути аналогове/цифрове перетворення, кодування (наприклад, за допомогою MPEG-4 кодека), а також додавання інформації прямої корекції помилок (FEC).

На відміну від цього, для програвання відео відеопотік повинен бути відновлений з серії отриманих фрагментів. Якщо який-небудь фрагмент відсутній, програма може відновити свою інформацію з інформації FEC, що міститься в сусідніх фрагментах. Після декодування відео можна програти.

2.3. Узагальнена архітектура програмного забезпечення P2P-мережі

Необхідність включення даного розділу виникла для формування цілісності викладеного матеріалу для забезпечення системності сприйняття.

На рис. 2.4. представлена узагальнена архітектура програмного забезпечення P2P-мережі.

2.3.1. Модуль планувальника

"Модуль планувальника" є ядром додатка P2P. Він відповідає за прийом і відправку фрагментів з/в інші піри через мережу, і з/в місцеві "модулі користувачів". Крім того, дані фрагменти повинні бути поміщені в буфер. При цьому кілька примірників цього модуля, що працюють одночасно, використовуються для роботи декількох оверлеїв, кожен із яких обслуговує один або кілька каналів.

Модуль є основою, на якій розміщується алгоритм планування фрагмента. Він визначає, коли починати передачу фрагмента в/з іншого піру. Підставою для цього рішення базується на інформації про наявність фрагментів у інших учасників, яка зберігається в активній базі даних пірів. Ця інформація оновлюється за допомогою повідомлень карти буфера, яка надсилається на розсуд відповідного планувальника. Різні варіанти розробки цього планувальника і вплив

на загальну продуктивність системи представляють особливий інтерес і будуть розглянуті в наступних розділах.

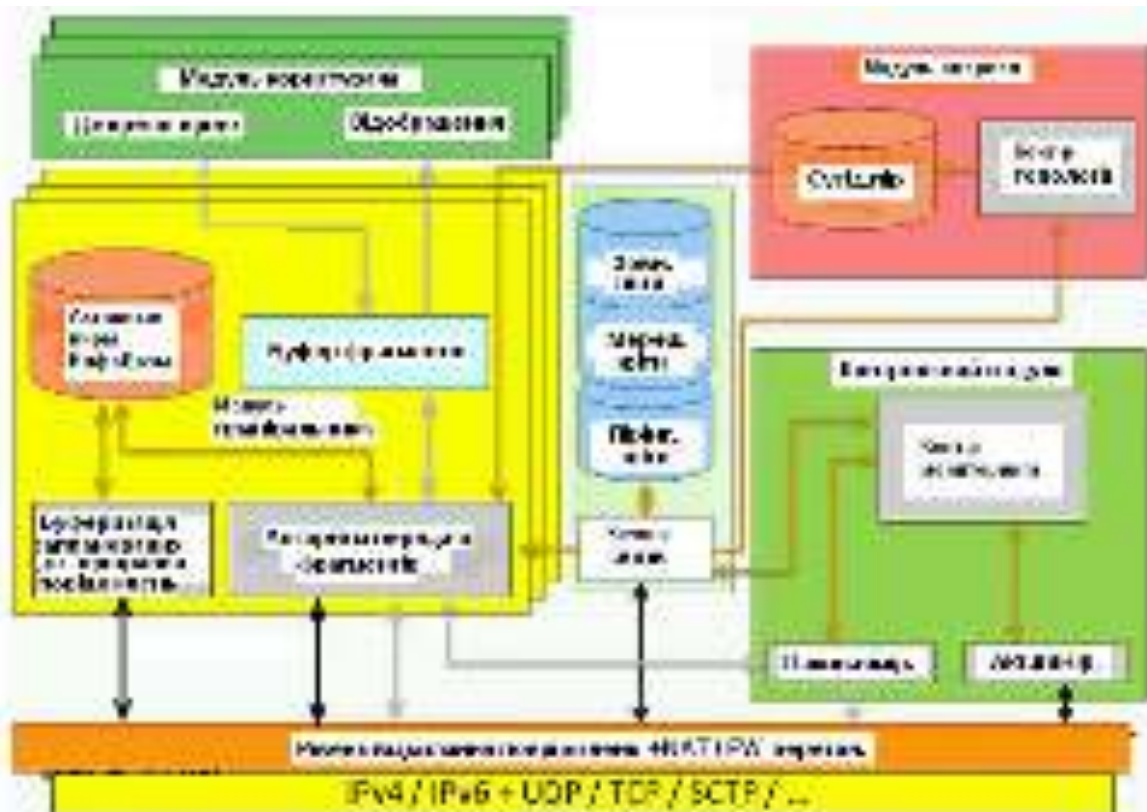


Рис. 2.4. Діаграма узагальненої архітектури програмного забезпечення піру

Проблема рівнодоступності базується на відповідних стимулах, носить суб'єктивний характер і виходить за межі даної роботи.

2.3.2. Модуль оверлею

Завданням модуля оверлею є виявлення нових пірів та встановлення зв'язку з ними, а також передача інформації про них іншим пірам. Кожен пір повинен підтримувати достатню кількість зв'язків з іншими пірами в оверлеї. Маючи досить велику кількість з'єднань важливо підтримувати добрі зв'язки в структурі оверлею, тому що піри можуть покинути його в будь-який час, наприклад, коли користувач перемикає канали або через технічні проблеми.

Однак із причин масштабованості пір не може підтримувати зв'язок з усіма іншими пірами в оверлеї. Отже, черговим завданням модуля оверлею є скорочення з'єднань, які не несуть користі для обміну фрагментів, наприклад, якщо пропускна здатність виявилася занадто низькою.

2.3.3. Контрольний модуль

Контрольний модуль призначений для збору інформації. Він має два основні режими роботи: контроль пасивних вимірювань спостережуваних повідомлень, якими обмінюються два піри, наприклад, для обміну відеофрагментами або сигнальною інформацією, і режим активних вимірювань: контроль спеціально сформованих повідомлень для зондування, які відправляються іншим пірам на розсуд модуля.

У результаті функціонування контрольного модуля з'являється інформація про зв'язок мережі. Така інформація, крім основного призначення, може бути корисна для збору даних про можливості підключення об'єктів, які не є частиною P2P оверлею, так званих «мережевих оракулів». Інформація, отримана з різних джерел, зберігається локально в декількох сховищах. Зміст цих сховищ, а також імпорту та експорту даних управляються контролером сховища.

2.3.4. Рівень передачі повідомлень

Даний рівень пропонує примітиви інших модулів для передачі і прийому даних в/з інших пірів. Він використовує протоколи сімейства TCP/IP і відповідних точок доступу послуги, пропоновані операційною системою, а також розширює їх функціональність. Наприклад, він забезпечує можливість відправки фрагмента, який більше не може вміститися в один UDP-пакет даних, іншим пірів. Рівень повідомлень також надає інтерфейс з контрольним модулем, який може бути використаний для пасивних вимірювань: всякий раз, коли повідомлення надсилається або приймається, відповідна вказівка буде передано в

контрольний модуль, який потім може оновити свою статистику. Ще одна важлива особливість рівня повідомлень – це механізми для відстеження від перетворювачів мережевих адрес [67]. Передача мережевих адрес дозволяє прикріпити кілька комп'ютерів до мережі Інтернет, використовуючи тільки одну унікальну глобальну IP-адресу. Тому вона дуже популярна серед приватних клієнтів, які також є основною цільовою аудиторією для P2P TV. Проте, наявність NAT пристрою може перешкоджати можливості встановлення зв'язку з іншими пірами [68], отже, на рівні обміну повідомленнями необхідні спеціальні функції обходу NAT.

2.4. Методи формування і послідовності відеофрагментів

Кожен медіапотік може бути розділений на фрагменти фіксованого або довільного розміру з використанням традиційного протоколу P2P. Іншими словами, ці системи, як правило, обробляють медіа у вигляді безперервного потоку бітів, які повинні прийти повністю, без втрат, всім пірам. Як наслідок цієї конструкції, важко контролювати затримку відтворення, і кінцеві користувачі ризикують мати субоптимальну якість.

З іншого боку, можуть бути підключені механізми розподілу P2P, які спеціально розроблені для потокового відео, в яких можуть враховуватися деякі особливості відео і використані адаптивні процедури, що дозволяють забезпечити кращу якість для кінцевих користувачів. Наприклад, відеопотоки характеризуються деякою надмірністю, яка може бути використана в системі розподілу. Ці потоки складаються з різних елементів при втраті одних якість відео може погіршитися, при втраті інших – помірний відсоток втрат може бути прийнятним. Всі ці властивості в значній мірі залежать від алгоритму, який використовується для кодування медіа. В результаті на кінцевого користувача впливають втрати, надмірність в самому медіапотіці й різні методи кодування відео. Розглянемо алгоритми розподілу та інформаційні структури.

Для аналізу втрат, які можуть бути припустимі в медіапотоці, необхідно визначити деякі важливі характеристики закодованих бітових потоків. Не концентруючись на якомусь конкретному алгоритмі кодування, можна стверджувати, що кодований аудіо або відеопотік складається з кадрів, кожен із яких починається з одного або декількох заголовків і потім містить стислі дані. Якщо такі заголовки втрачені, декодувати решту кадру неможливо. Більш того, деякі заголовки містять, наприклад, важливі властивості відео, такі як розмір, глибину кольору і т.д. Очевидно, поки ці заголовки не будуть відомі, кадр не можна розшифрувати.

Крім того, більшість сучасних алгоритмів кодування відео кодують три різні види кадрів: I кадри (що не залежать від інших кадрів), P кадри (екстрапольовані кадри, які кодуються як частини щодо попереднього кадру) і B кадри (двонаправлені екстрапольовані кадри, які кодуються як частини щодо попереднього і наступного кадру). Якщо кадр, від якого залежать P або B кадри, неправильно декодований, то цей кадр також не може бути декодованим правильно, навіть якщо він був правильно прийнятий. З цієї причини втрата частини I кадру є більш проблематичною, ніж втрати частини P або B-кадру, тому в потоковій системі P2P можуть призначатися різні пріоритети для фрагментів відповідно до їх змісту (кадри I, B кадри, і т.д.). Дані пріоритети зазвичай виявляють у вигляді штрафів за ті чи інші фрагменти.

Іншою важливою властивістю є те, що в процесі розбиття на фрагменти слід уникати, наприклад, поділу одного кадру на різні фрагменти, завдяки чому вдається зменшити ефект втрати фрагмента. Така фрагментація може також зменшити затримку програвання. Однак, зміст цілого числа кадрів у фрагменті може привести до того, що фрагменти матимуть різні розміри, які не підтримуються деякими алгоритмами P2P-розподілу. При необхідності цю проблему можна вирішити шляхом додавання більш дрібних частин фрагментів із нулями так, щоб всі вони набували однакового розміру. У той же час це може привести до марної витрати пропускнуої здатності мережі. Щоб скоротити ці витрати, можна використовувати звичайну структуру багатьох сучасних

алгоритмів кодування відео, які періодично повторюють одну і ту ж послідовність типів кадрів (так званий GOP - Group Of Pictures або Група картинок) з періодом, званим GOP size (GOP розмір). Отже, якщо фрагмент містить ціле число кадрів, яке є кратним розміром GOP, відмінність між фрагментами буде невеликою, а фрагментація з урахуванням особливостей структури мультимедійних файлів може виявитися достатньою. Разом із тим практика показує, що виграші від ідентичності розмірів фрагментів досить незначні. Більш того при високих швидкостях передачі значний виграш дають великі фрагменти. Таким чином, в даній роботі на завданні рівності фрагментів не будемо зосереджуватися.

Іноді надмірність, що міститься в закодованому середовищі, не достатня для «покриття» втрачених фрагментів, тому деяка додаткова надмірність повинна бути додана при генерації фрагмента. Це може бути зроблено без інформації про середовище (з використанням FEC), але більш висока продуктивність для додавання надмірності очікується при використанні деякою інформацією алгоритму кодування, наприклад, передаючи заголовки кадрів (які є критичними для декодування кадру) кілька разів або дублюючи ключові кадри або I кадри. Ще один спосіб для імпліцитного представлення деякої надмірності, щоб втрата фрагмента не мала великого впливу, заснована на багаторівневому кодуванні або MDC. В цьому випадку кодований потік ділиться на різні шари (в разі шарового кодування) або опису (в разі MDC - Multiple Description Coding).

Шарове кодування використовує базовий шар, який забезпечує основне уявлення середовища про низьку якість, а також кілька рівнів розширення, які самі по собі не є корисними, але можуть бути використані для уточнення якості базового шару (зауважимо, що шар з рівня n можна використовувати тільки тоді, коли шар рівня $n-1$ був успішно прийнятий і декодований). Таким чином, різні шари мають різні рівні критичності (наприклад, якщо базовий шар втрачається, всі інші шари виявляються марними), а шарове кодування корисне тільки тоді, коли в механізмі розподілу P2P передбачена можливість призначати різний ступінь важливості різних фрагментів. MDC (Multiple Description Coding), з іншого боку, ділить потік на набори описів, що мають однакову ступінь

важливості. Кожен опис може бути використаний для декодування варіантів медіа низької якості, а інші описи можуть бути використані для поліпшення якості. Проте, базовий шар першим не повинен бути декодований.

Шарове кодування і MDC строго пов'язані з алгоритмами кодування, і хоча сучасні алгоритми кодування відео підтримують шарове кодування (H.264, MPEG4, і MPEG2), цієї функції з відкритим вихідним кодом немає.

Існує Internet Draft [70] опис того, як взаємодіяти з багаторівневим кодуванням і/або MDC у стандартному алгоритмі кодування відео (попередньою обробкою відео перед його кодуванням і відновлення вихідного відео після декомпресії), але такий підхід може призвести до низької ефективності кодування.

Якщо використовується MDC або шарове кодування, потокова система може скористатися перевагою розподілу потоку на фрагменти. Наприклад, якщо використовується MDC, фрагмент може містити єдиний опис GOP (або цілого числа GOP); отже сегмент відео передається в ряді фрагментів, що дорівнює числу описів, використовуваних MDC-алгоритмом. Однак, оскільки MDC генерує якусь приховану надмірність, використання такого роду методу, як правило, призводить до збільшення швидкості передачі (отже, як і раніше не ясно, чи є застосування MDC перевагою в системах потокової передачі P2P).

З використанням метрики PSNR і SSIM проведемо аналіз впливу втрат фрагментів на якість P2P-TV.

2.5. Структура і топологія оверлею

Відомо кілька використовуваних на практиці технологій оверлею.

Деревоподібна топологія оверлею. В оптимальній системі P2P, кожному піру поставляється тільки одна копія кожного фрагмента. Тобто, якщо можна було б записати трасування фрагмента і його реплікацію, можна було б уявити дерево графіка з коренем у вигляді джерела, який підключається до кожного приймача.

У деревовидній топології оверлею такий деревовидний графік розраховується і встановлюються зв'язки між пірами. Фрагменти просто повинні бути спрямовані уздовж країв. Проте кожного разу, коли пір приєднується або залишає систему, дерево має бути перебудоване.

Якщо внутрішній вузол виходить з ладу, всі вузли в піддереві, яке знаходиться нижче нього, більше не будуть отримувати відео, поки дерево не буде перебудовано. Будучи пов'язаним з MDC, підходи декількох дерев пропонують створити кілька непересічних дерев таким чином, щоб кожен пір мав важливе (внутрішнє) становище тільки в одному з них, будучи в той же час кінцевим вузлом на інших деревах. Кожне дерево буде нести тільки деякий відсоток всіх фрагментів, і якщо важливий вузол на ньому виходить з ладу, тільки цей відсоток фрагментів не дійде до всіх пірів. Хоча такий підхід може зменшити цю проблему, він не вирішує основне питання про те, що деревоподібна топологія повинна бути перерахована, коли пір виходить або дає збій.

Mesh-топология оверлею. Mesh-топологии не мають чіткої структури. Кожен пір має зв'язок із декількома іншими пірами, які можуть бути обрані відповідно до різних стратегій, в найпростішому варіанті – випадково. Більш того, форма цієї сітки постійно змінюється: сполуки, які не несуть корисний трафік, можуть припинятися, а замість них можуть використовуватися інші. Дана система більш динамічна за своєю природою, і вона може краще впоратися з плінністю абонентів і з порушеннями в роботі вузлів і каналів.

Структуровані системи. Ці структури зазвичай пов'язані з топологіями одного дерева або множини дерев. Однак, структура оверлею, в особливих випадках, може слідувати звичайній або напівдетермінованій топології, як, наприклад, гіперкуби або гіпертори, так що рівномірна структура зв'язку дерева порушується.

Більш узагальнено можна стверджувати, що структурована система намагається підтримати топологію оверлею якомога ближче до детермінованої, а для цього необхідні експліцитні функції управління топологією і протоколи, які

якимось чином перебудовують структуру системи незалежно від того, чи приєднуються, від'єднуються піри або їх робота порушується .

Неструктуровані системи. Будь-яка система, що не потрапляє під визначення перерахованих вище, вважається неструктурованою. Більш детально неструктуровані системи виділяють топологію тільки в стохастичних умовах (наприклад, кожен пір повинен знаходитися між 10-20 сусідами, обраними випадково). Результиуюча топологія вийде узагальненою mesh-топологією, можливо з якоюсь додатковою властивістю, наприклад, посилення симетричних зв'язків сусідів, пріоритетна локальність або пропускна здатність пірів. Неструктуровані системи іноді називаються саме структурованими, що відображають високий ступінь динамічності і децентралізації в порівнянні зі структурами, де піри повинні бути однорідні для спільної мети топології.

Нам представляються неструктуровані системи найбільш перспективними, більш того: багато сучасних P2P-мереж функціонують саме за неструктурованими технологіям. Наші подальші дослідження орієнтовані на неструктуровані системи.

2.6. Структура мультимедійних файлів

Медіадані перед впровадженням у систему поділяються на частини в залежності від того, чи походять вони від P2P систем поділу файла або від традиційних систем передачі відеопотоку:

- системи на основі фрагментів;
- системи на основі смуг.

Зазвичай системи, засновані на фрагментах, пов'язані з неструктурованою топологією, в той час як системи на основі смуг – з деревами (або структурованими топологіями в цілому).

2.6.1. Системи на основі фрагментів

У системах, заснованих на фрагментах, медіаконтент ділиться на серії фрагментів. Розміри фрагментів можуть бути різні чи заданого фіксованого розміру (заданої тривалості). Ці фрагменти утворюють елементарні одиниці передачі між пірами. Передача фрагментів виконується за принципом «зі сховища», тобто пір почне перерозподіляти даний фрагмент, коли вже отримано фрагмент цілком. Ці перерозподіли забезпечують необхідні відносини передування фрагментів, мінімізують середній час затримки та ін.

У системах, заснованих на фрагментах, кожен фрагмент в оверлеї має самостійне життя. Фрагменти можуть прибувати в піри в різному порядку, тому кожен вузол потребує буфер фрагментів, щоб змінити їх порядок, перш ніж програти відеоконтент. Даний буфер також використовується в якості джерела фрагментів для інших пірів, може надати фрагменти іншим учасникам, а стан даного буфера (як правило, представлений у вигляді бітової карти з 1 в точці, де фрагменти в даний час належать конкретному піру) використовується під час повідомлення про передачі фрагментів.

Така незалежність між фрагментами є прикметною ознакою систем на основі фрагментів і означає, що піри приймають рішення про передачу (який фрагмент передати і кому направити) на основі принципу фрагмент-за-фрагментом. Це рішення (так зване рішення планування) є найбільш важливою особливістю систем на основі фрагментів. Процес прийняття зазвичай розбитий на послідовність двох рішень: вибір піру для зв'язку і вибір фрагмента для відправки (або запитуваний фрагмент). На сьогоднішній день є численні пропозиції про планувальнику з повідомленням. Нижче проаналізовано їх характеристики і запропоновано нові оптимізовані рішення планування з повідомленням і без, які дають поліпшені, порівняно з попередніми, результати.

Розмір фрагмента також є важливим параметром, що характеризує якість системи. Багато з потокових прототипів P2P на основі фрагментів використовують відносно великі фрагменти. Тут під словом «великий» маємо на

увазі фрагмент із тривалістю в секунду або, якщо вимір виконується в байтах, фрагменти до кількох сотень кілобайт (це практично означає кілька сотень IP пакетів, необхідних для передачі фрагмента).

З одного боку, наявність таких великих фрагментів має ряд переваг: 1) потрібен мінімальний сигнальний трафік, необхідний для обміну станом буфера фрагментів; 2) час, потрібний для дифузійної передачі змін в інформації про стан буфера фрагментів інших пірів мізерно малий в порівнянні з фактичним часом передачі фрагмента; 3) фрагменти можуть бути передані з використанням TCP, який на цій довжині потоку досягає хороших швидкостей передачі, забезпечує доставку при повторній обробці і забезпечує контроль перевантаження для окремих передач.

З іншого боку, великі фрагменти пов'язані з більшою затримкою, яка з'являється вже в джерелі під час генерації фрагмента, що є критичним параметром у потокових додатках. Крім того, оскільки фрагменти передаються за принципом «з сховища», та ж затримка додається кожного разу, коли фрагмент передається від піра до іншого піра.

Коли розмір фрагмента зменшується, стан буфера змінюється частіше, тому сигнальні повідомлення, що попереджають піри про цю зміну, можуть відправлятися теж частіше (оптимальна частота таких повідомлень залишається відкритим питанням для дослідження). Це не тільки збільшує сигнальний заголовок, але ще і скорочує інформацію, доступну для пірів про інших пірів, при плануванні рішення. Оскільки частота планування збільшується, повідомлення про стан буфера не може прийти вчасно, і тому інформація про стан буфера інших пірів може бути застарілою. Більш того, якщо використовуються методи кодування медіа з ефективним змінним бітрейтом, більш дрібні фрагменти також можуть означати більш високу дисперсію в тривалості фрагментів або їх розмірах.

2.6.2. Системи на основі смуг

Системи на основі смуг поділяють медіа-потік на фіксовану кількість субпотоків (смуг) із використанням шаруватого кодування або MDC. Характерна риса таких систем полягає в тому, що кожна смуга має свою індивідуальну структуру розподілу (в основному дерево з коренем в джерелі медіа), в той час як піри змінюються по суті частиною кожного з цих розподільчих структур з тим, щоб забезпечити гарантію отримання всього медіапотіку.

Смуги часто розглядаються суцільними, тобто вони передаються в потоці безперервно від одного піру в інший без механізму «зі сховища», і це є причиною того, чому такі системи пов'язані зі структурованими системами: стратегія доставки інформації включена в саму топологію. Таким чином, в смузі немає доставки в різному порядку (це може повторитися через механізми ARQ, IP маршрутизації і ін.).

Мають місце суттєві відмінності між системами на основі фрагментів і системами на основі смуг. Системи на основі смуг можна апроксимувати за допомогою спеціальних систем на основі фрагментів з малими розмірами фрагментів. На практиці, нижній шар IP завжди підтримує поведінку «зі сховища-вперед», а кадровий характер оцифрованого відео додає ще один шар квантування. Це особливо важливо, якщо обробка виконується на рівні додатків, де ступінь деталізації, в якій потік може бути оброблений, ще вищий.

Механізми відновлення, що використовуються в системах на основі смуг, відрізняються також від неупорядкованої доставки систем на основі фрагментів, проте з відповідною логікою вибору фрагмента і оновленнями стану буфера їх також можна імітувати до певної міри. З цієї причини в подальшому зосередимося в основному на системах, заснованих на фрагментах, але в межах цих систем розуміємо ширше визначення, в якому розмір фрагмента може бути маленьким, а поновлення стану буфера може виконуватися рідко.

З іншого боку, оскільки як в дійсності смуги складаються з кадрів (або часткових їх описів), які в будь-якому випадку нумеруються в більшості (якщо не

у всіх) методів кодування відео, динамічне планування може бути застосоване також і до системи смуг таким чином, що цим смугам не потрібно слідувати дифузійним шляхом, чітко прописаним в топології структурованих систем, але вони моли б слідувати будь-якому динамічному шляху (або неструктурованому) в загальній сітці на підставі планування.

2.7. Висновки до другого розділу

1. Архітектура P2P мереж відрізняється від традиційної клієнт-серверної архітектури тим, що відповідні учасники мережі піри (ліч – ті, що скачують, і сід – ті, що роздають) в залежності від обстановки можуть мінятися місцями, тобто між учасниками – рівноправні відносини.

2. На відміну систем відео за запитом, що поширюється мережами доставки контенту (CDN), технологія пірингового зв'язку має ту перевагу, що не вимагає великих розмірів буфера, а послуга TV-перегляду може бути надана в будь-який час після початку трансляції.

3. Структура мережі P2P може будуватися на кластерній і бескластерній основі. При бескластерній технології виділяється п'ять характерних дій: приєднання/від'єднання піру, процес обміну файлами, вибір піру з потрібними файлами, організація буфера і планування сегментів. При кластерній технології виділяється вісім дій, з них додатковими є вибір супер-вузла, вибір резервного вузла. При виборі кластерної технології вдається поліпшити масштабованість і показники QoS, зокрема скоротити затримки часу. Це дає підставу для вибору кластерної топології.

4. Модуль планувальника реалізується у вигляді відповідної програми і є ядром в мережі P2P, оскільки відповідає за прийом і відправку фрагментів. Центральним завданням, від якого залежить якість викачуваного відео є планування фрагментів, що включає обмін інформацією про наявність тих чи інших сегментів у доступних пірів, вибір відповідних сегментів для скачування і формування в буфері запитаного файлу і підготовці його для гладкого

програвання. Найважливішими алгоритмами при цьому є алгоритм приєднання піру і алгоритм оптимального вибору кращого фрагмента з наданих.

5. Модуль оверлею реалізує не тільки механізм накладеної мережі, але і технологію виявлення нових пірів, встановлення з ними зв'язку, рішення задач масштабованості мережі при зростанні навантаження. Особливу значущість при цьому набуває взаємодія динаміки оверлею і IP-маршрутизації.

6. Для забезпечення рівного відтворення кожен P2P вузол повинен мати буфер для зберігання прийнятих фрагментів. У складі буфера міститься 3 частини: буфер даних, буферна карта і змінне вікно. Буферна карта дозволяє здійснювати обмін інформацією про наявність тих чи інших сегментів. При цьому сід-піру надається можливість вибору партнера з мінімальною затримкою, максимальною пропускною здатністю. Буфер даних у свою чергу ділиться на оглядовий, відображаючий і майбутній.

7. Структура мультимедійних файлів безпосередньо бере участь у процесі фрагментації. При цьому поділ файлів на частини можливий на основі смуг і на основі фрагментів. Системи на основі фрагментів зазвичай пов'язані з неструктурованою топологією, що надає їм відповідну пріоритетність. У той же час відмінність в розмірах фрагментів і їх незалежність вимагають відповідного управління упорядкуванням і розподілом цих фрагментів.

8. Структура і топологія модуля оверлею забезпечує виявлення нових пірів та встановлення зв'язку з ними, а також передачу інформації про них іншим пірам. Крім того функціями модуля оверлею є мінімізація числа з'єднань, що встановлюються при запиті з боку ліч-піру. Архітектура пірингових мереж може бути структурованою або неструктурованою. У структурованих системах P2P-NQTV структура оверлею зазвичай являє напівдетерміновану топологію, що викликає складності в реструктуризації при приєднанні або від'єднанні чергового піру. Відмінною особливістю неструктурованою архітектури P2P-TV систем є те, що топологія оверлею повністю відокремлена від топології розподілу. В результаті утворюється mesh-топологія дозволяє реалізувати високий ступінь

динамічності і децентралізації. Саме неструктуровані технології привертають дедалі більшу увагу. У наших дослідженнях також дотримуємося цих тенденцій.

9. У результаті аналізу архітектури однорангових пірингових P2P- мереж впливає висновок про те, що серед інших систем надання послуг з доставки відео через Інтернет: OTT-TV, IPTV та ін., P2P-системи дозволяють економити мережеву інфраструктуру, мають високу масштабованість, відзначаються гнучкістю для різних типів додатків, можливістю мовлення з різним початком. Велика кількість використовуваних на практиці евристичних алгоритмів, що супроводжують процес планування і скачування фрагментів, управління буфером, вимагає пошуку оптимальних рішень за даними алгоритмами, що, природно, забезпечить якість надання послуги QoS і якість надання відео QoE.

РОЗДІЛ 3

РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ПІРИНГОВИХ МЕРЕЖ

Стан пірингових мереж визначається тим, як взаємодіють і самоорганізуються піри в оверлейній мережі і як вони зберігають або ретранслюють запитані файли один для одного. Основна проблема полягає в тому, щоб домогтися ефективного пошуку ресурсів у великомасштабній розподіленій мережі зберігання даних.

На сучасному етапі популярні пошукові системи P2P можуть бути класифіковані як неструктуровані та структуровані в залежності від структур оверлею. Неструктуровані системи не нав'язують жодної структури оверлейних мереж. Ці системи зазвичай стійкі до динаміки пірів та підтримують складні запити з метайнформацією. Однак вони не є ефективними для пошуку рідкісних непопулярних файлів. Структуровані системи накладають певні обмеження структури на оверлейні мережі, які зазвичай називаються розподіленими хеш-таблицями (DHT). У структурованій системі будь-який файл може бути розташований у невеликій кількості переходів оверлею, що значно знижує вартість пошуку в порівнянні з неструктурованими системами. Проте, DHT підтримує тільки пошуки по одному ключовому слову з точною відповідністю.

Залежно від рівня децентралізації системи ми класифікуємо неструктуровані мережі P2P на централізовані, розподілені, гібридні і аналізуємо їх в умовах динаміки функціонування.

3.1. Математична модель динаміки пірингової P2PTV-мережі

3.1.1. Структура алгоритму управління пірингових мереж

Процеси, що відбуваються в пірингових мережах, відрізняються високою динамікою.

Коли новий вузол приєднується до мережі для скачування відповідних файлів, сервер надає йому випадковий список пірів.

Глобальний трекер (GT), він же сервер, відомий усім вузлам і підтримує список усіх вузлів мережі. Новий вузол обмінюється буферною картою зі списком пірів та вибирає черговий фрагмент з m -пропонованих пірами-сусідами, від яких планує завантажити потокове відео. Роль даного виду контенту в тому, що використання (перегляд) відрізняється певним часовим масштабом і передуманням. Крім того, при скачуванні контенту виникає необхідність урахування інформаційної структури відеопотоку, завантаження буфера, роботи планувальника й ін. Вибір випадає на ті піри, інформація від яких може надходити з мінімальною затримкою, при максимальній пропускну здатності або випадковим чином, в залежності від технології. Якість вибору визначається показником, де m – число, що бере участь в виборі. Розглянемо більш докладно процес скачування. Заходи, що передують скачуванню обраного файлу i -м піром, складаються з наступних циклів [3]:

- приєднання піру NN;
- вибір супер-вузла SN і резервного вузла BN;
- обмін інформацією між пірами;
- вибір пірів;
- організація буфера (буфера даних, буферної карти, ковзаючого вікна);
- планування фрагментів.

Управління буфером передбачає виконання функцій:

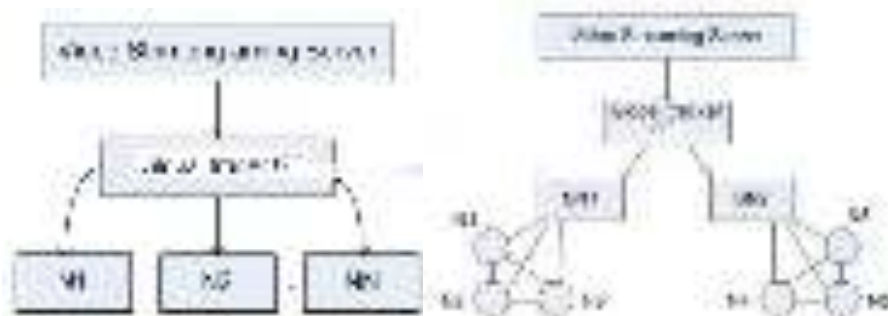
- управління буфером даних,
- управління буферною картою,
- управління ковзаючим вікном.

Буфер даних в свою чергу складається з трьох частин:

- оглядового буфера (зберігання старих фрагментів),
- дисплейного буфера (зберігання свіжих фрагментів),
- наступного буфера (резерв для майбутніх фрагментів).

З метою розвантаження сервера і поліпшення продуктивності поширення контенту піри можуть бути згруповані в кластери. У кластерній моделі потрібно використовувати кілька трекерів, окремо для кожного кластера [50,53].

Структурна схема P2PTV-мережі представлена на рис.3.1



а) Структура некластерної мережі б) Структура кластерної мережі

Рис. 3.1. Схеми не кластерної і кластерної структур мережі P2PTV

3.1.2. Математична модель динаміки P2PTV-мережі

Пірингові мережі P2PTV, так само як й інші інфокомунікаційні мережі, розвиваються в основному за рахунок удосконалення технологічних рішень. Отже, єдиної узагальненої математичної моделі мережі побудувати не вдасться. Спробуємо уявити структурно-функціональну модель, що відображає динамічні властивості скачуваного процесу. Найпростішою приватною моделлю n -узлової мережі є марківська модель [22,28]. У пірингових мережах головним є процес $x_i(t)$, що відбувається при скачуванні фрагментів відеоконтенту. На кожному k -му кроці скачування чергових фрагментів розглядається m -вектор запропонованих реалізацій:

$$x_k = Ax_{ij}^{(k)}, \quad (3.1)$$

де x_{ij}^k – вектор фрагментів, що передається лич-піру від сид-піра $i, j = \overline{1, m}$.

$$A = \begin{bmatrix} A_{ij}^{(1)} & 0 & \dots & 0 & 0 \\ 0 & A_{ij}^{(2)} & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & A_{ij}^{(m)} \end{bmatrix}, \quad (3.2)$$

де A^m – елементи матриці, які демонструють якісні показники кожного з m - запропонованих фрагментів скачуваного файлу.

Розмірність векторно-матричного управління визначається розмірністю скачуваного файлу, де $k \in K = (k_1, k_2, \dots, k_F)$.

Уявлення (3.1) відображає статичний стан. Для переходу до стану динаміки необхідно скористатися диференціальною або різницевою моделлю:

$$d\bar{x}(t)/dt = A\bar{x}(t), \quad (3.3)$$

де $\bar{x}^T(t) = (x_1, x_2, \dots, x_{k_F})$ – вектор динамічного стану i -х пірів, скачують конкретний файл.

У розгорнутому вигляді рівняння (10.3) набирає вигляду:

$$dx_i(t)/dt = \sum_{j=1}^k A_{ij}^{(k)} (x_j(t) - x_i(t)), \quad j = \overline{1, k}, \quad (3.4)$$

де x_j – стан скачуваного файлу,

$A_{ij}^{(k)}$ – матриця,

$\Delta x_{ij} = (x_j(t) - x_i(t))$ – різниця, що визначає темп завантаження контенту.

Слід зазначити, що модель (3.4) вважається базовою для різних багатоагентних мереж.

У більш загальному випадку елементи матриці $A_{ij}^{(k)}$ можуть носити випадковий характер, визначати ту чи іншу ймовірність успішного вирішення при виборі відповідного фрагмента, що належить j -му піру. Саме ж рішення залишається за планувальником.

Кінцево-різницевий аналог диференціальної моделі (3.4) має вигляд:

$$x_i(k+1) = x_i(k) - \varepsilon \sum_{j=1}^n a_{ij} (x_j(k) - x_i(k)) \quad j = \overline{1, 2, \dots, n}, \quad (3.5)$$

де k – крок дискретизації, номер чергового скачуваного кадру або фрагмента;

$\varepsilon > 0$ – крокова постійна різницевої схеми, що обирається, планувальником і визначає швидкість збіжності процедури (3.5).

Для забезпечення керованого процесу (3.5) врахуємо впливу планувальника $u(k)$:

$$x_i(k+1) = x_i(k) - \varepsilon \sum_{j=1}^n a_{ij} (x_j(k) - x_i(k)) + Bu(k), \quad (3.6)$$

де $u(k) = K(x_j(k) - x_i(k))$ – керівний вплив, лінійно залежне від стану j -го піра, з урахуванням затримки, пропускнуої спроможності та ін.

Оскільки стан $x_j(k)$ - відомо, то можна вказати граничне значення затримки τ_{j_0} . Таким чином, управління

$$u(k) \approx x_j(k - \tau_{j_0}), \quad 0 < \tau_k \leq \tau_{j_0} \quad (3.7)$$

де τ_k – реальна затримка.

Управління $u(k)$ в системі (3.6) доцільно вибрати у вигляді релейних впливів

$$u(k) = (-k_1 \text{sign}[S_1(k_1)], -k_2 \text{sign}[S_2(k_2)], \dots, -k_n \text{sign}[S_n(k_n)])^T, \quad (3.8)$$

де k_i – номери кроків, відповідних числу скачуваних фрагментів;

$S_i(k_i)$ – лінійні відображення $S_i: R^n \rightarrow R$, що забезпечують значення керівних впливів:

$$\text{sign}[z] := \begin{cases} 1, & \text{если } z > 0, \\ 0, & \text{если } z \leq 0. \end{cases}$$

Очевидно при $z > 0$ в рядку матриці $A_{ij}^{(k)}$ встановлюється «1» на місці, відповідному скачуваному фрагменту.

Розроблена математична модель динаміки пірингової P2P-мережі відображає важливі системні властивості її функціонування, що зводяться по суті до функціонування планувальника. Слід зазначити, що цілий ряд динамічних процедур, що відбуваються в цих мережах залишилися поза зоною уваги авторів. Водночас видається неможливим охопити однією, як завгодно загальною моделлю всі ці важливі деталі.

Вважаємо, що запропонована диференційна (різницева) математична модель може бути базовою для досить широкого використання і узагальнення. Запропонована математична модель пірингової P2PTV-мережі представлена у вигляді диференціальних або різницевих рівнянь, яка може бути віднесена до класу базових для мультиагентних мереж. Вона узгоджена з процесом фрагментації відеоконтенту, а також дозволяє врахувати можливість отримання фрагментів від різних пірів, включених в створюваний кластер. У розробленій моделі врахована особливість роботи планувальника, процедура якого включає управління ліч або сід - процесами. Вибір релейної стратегії управління дозволяє адекватно відображати зазначені процеси управління.

3.1.3. Аналіз динаміки пірингової мережі

Основним функціональним показником якості динамічної системи є похідна від її стану $dx(t)/dt$, що за своєю сутністю представляє швидкість зміни стану. Однак в складних системах паралельно відбуваються різні процеси в окремих елементах і ці процеси тією чи іншою мірою між собою узгоджені, що забезпечує стійкість функціонування цієї складної системи.

Для аналізу якості функціонування складних систем організовується контроль стану найбільш значущих елементів і процесів у них. Але не менш важливим є сама організація моніторингу, його методика та форма надання результатів, чим забезпечується об'єктивність самого контролю.

У даному розділі представлено розроблену методику аналізу якості і результати її використання стосовно пірингової мережі з обробкою живого потокового відео.

За результатами аналізу динаміки деяким пірам з ряду причин доводиться відмовляти в обслуговуванні. Так, постачальникам послуг важливо стежити за якістю надання послуг і усувати ті піри, які пов'язані з відмовами або з тривалими затримками, недостатньою якістю наданого контенту.

Динамічні характеристики пірингової мережі визначаються швидкістю скачування контенту V_d (download) і швидкістю його завантаження V_u (upload). Для конкретного піра P_i ці характеристики визначаються

$$\begin{aligned} V_d &= d\lambda_d(t)/dt, \\ V_u &= d\lambda_u(t)/dt, \end{aligned} \quad (3.9)$$

де λ_d, λ_u – відповідно інтенсивності потоків скачування і завантаження.

У співвідношеннях балансу між швидкостями завантаження і скачування рівність визначає швидкість завантаження відеосервера

$$\int_0^{N(t)} V_u(t, v) dv + U = \int_0^{N(t)} V_d(t, v) dv, \quad (3.10)$$

де U – швидкість завантаження відеосервера.

Кожен з i -пірів має свою стратегію поведінки. Зокрема можлива повністю егоїстична стратегія, при якій пір тільки викачує інформацію зі швидкістю V_{ui} , але іншим – не завантажує $V_{ui} = 0$. Ступінь егоїзму піра визначається відношенням швидкостей:

$$S(t) = V_u(t)/V_d(t). \quad (3.11)$$

Показник $S(t)$ може змінюватися в межах від $S_i = 0$ до досить великих чисел. При $S_i \geq 1$ відповідний пір демонструє щедрю альтруїстичну стратегію. Очевидно, якість роботи пірингової мережі в цілому визначається значеннями даного показника, наведеного до поточного числа активних пірів:

$$Q(t) = \frac{1}{N} \sum_{i=1}^N S_i(t) \quad (3.12)$$

де N – число активних пірів, змінна величина, від якої залежить якість $Q(t)$. При досить великому числі пірів можна замінити суму на інтеграл, а N на $N(t)$. У цьому випадку (3.12) представляється у вигляді

$$Q(t) = \frac{1}{N(t)} \int_0^{N(t)} S_i(t) t_1 v dv. \quad (3.13)$$

Отримаємо явну залежність зміни якості $Q(t)$ від параметрів.

Для цього знайдемо похідну функції $Q(t)$.

$$N'(t) = \frac{1}{V_d} \int_0^{N(t)} \frac{\partial V_u(t, v)}{\partial t} dv. \quad (3.14)$$

Замінюючи $N'(t)$ з (3.7), отримуємо

$$\frac{dQ(t)}{dt} = -\frac{N'(t)}{N(t)} \cdot Q + \frac{N'(t)}{N(t)}. \quad (3.15)$$

Рівняння виконується при будь-яких t . Замінюючи $Q(t)$ і $N(t)$ відповідно на $Q(t_0)$, $N(t_0)$ отримуємо

$$Q(t) = 1 - \frac{N(t_0)(1 - Q(t_0))}{N(t)}. \quad (3.16)$$

Очевидно зі збільшенням $N(t)$ показник якості $Q(t) \rightarrow 1$.

З урахуванням $V_d(t, v) = V_d$ можна спростити умову балансу (3.16):

$$\int_0^{N(t)} V_u(t, v) dv + U = N(t) \cdot V_d, \quad (3.17)$$

Співвідношення показує, що швидкість завантаження пірів та швидкість потокового сервера разом визначають загальну швидкість завантаження в системі.

З урахуванням (3.17) одержуємо

$$Q(t) = 1 - \frac{U}{V_d \cdot N(t)}. \quad (3.18)$$

Причиною монотонного зростання показника $Q(t)$ є висхідна кількість пропозицій на завантаження, пропорційне зростанню кількості активних пірів. Ця властивість принципово відрізняє динаміку P2P від IPTV, OTT де відбуваються зворотні процеси. Особливо важливо це для знову включених пірів, у яких ще відсутні відеоматеріали, якими вони могли б поділитися з іншими.

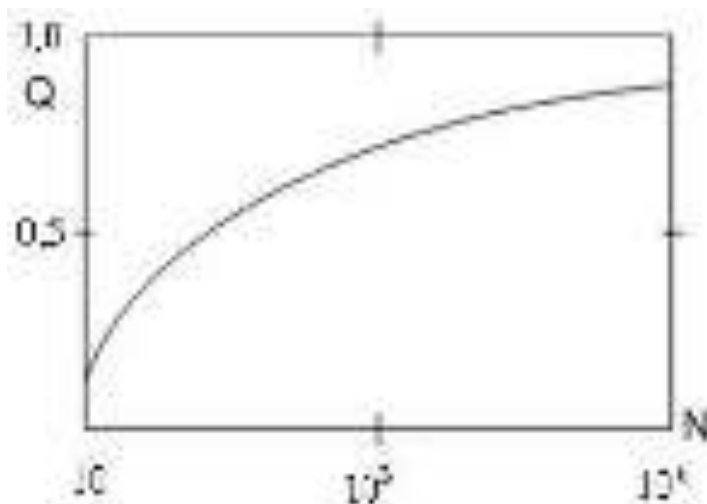


Рис. 3.2. Графік залежності показника якості від навантаження

Характеристика зростання середньої відносної швидкості завантаження в піринговій мережі зі збільшенням числа активних пірів

Ідеалізована ситуація (3.18) на практиці обмежується граничним значенням швидкості завантаження, $V_u \leq V_{u \max}$ і як наслідок – обмеження продуктивності мережі. Такі явища часто наступають, коли починають відбуватися певні особливо популярні масові заходи, інтерес до яких у споживачів різко зростає. При цьому монотонна залежність (рис.3.3) порушується.

Розглянемо окремий випадок, коли відбувається масове підключення нових пірів та система не протидіє їх підключенню. В цьому випадку показник якості – знижується, і на підставі (3.7) отримуємо:

$$dQ(t)/dt = \frac{N'(t)}{N(t)} Q(t).$$

Дане лінійне диференціальне рівняння перетворюється послідовно в наступні форми:

$$\frac{dQ(t)/dt}{Q(t)} = -d(\ln N(t))/dt,$$

$$\ln Q(t) = -\ln N(t) + C,$$

$$\ln Q(t) = \ln \left(\frac{\exp^C}{N(t)} \right),$$

$$Q(t) = \frac{\exp^C}{N(t)}, \text{ де } C = \ln Q(t_0)N(t_0).$$

Остаточно:

$$Q(t) = \frac{Q(t_0)N(t_0)}{N(t)}.$$

Таким чином в умовах великого лавинного навантаження поточна якість $Q(t)$ може знижуватися в $N(t)/N(t_0)$ разів порівняно з якістю, яка мала місце в стаціонарному режимі функціонування мережі.

Графік залежності показника якості при цьому набуває інших форм рис. 3.3.

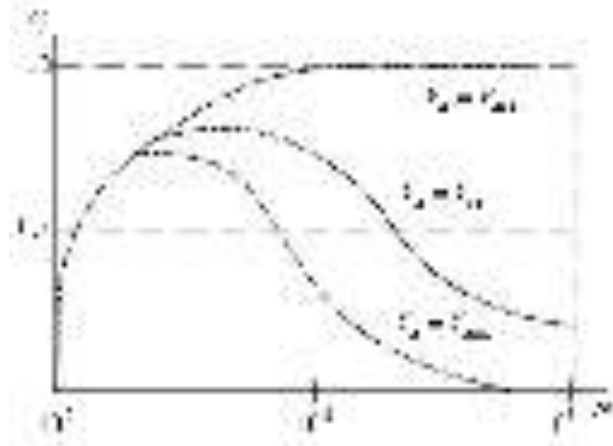


Рис.3.3. Графік залежності показника якості в умовах дефіциту динаміки завантаження і при лавинному попиті на контент

Отже, масова поява нових пірів у мережі, у яких ще немає контенту для завантаження, може призвести до втрати якості функціонування мережі. Поступово з наданням сервером-трекером кожному з нових пірів списку декількох потенційних батьківських пірів, які переглядають один і той же канал, положення вирівнюється і мережа набуває стаціонарного характеру. Нові піри в міру скачування контенту починають роздавати накопичені файли іншим зацікавленим пірам. Крім того, різких падінь мережевої продуктивності, представлених нижніми кривими на рис.3.3 можна уникнути, тимчасово збільшуючи число відмов в періоді ГНН.

3.2. Оптимізація процесу відбору скачуваних файлів у пірингових мережах

Пірингові мережі, будучи приватними, оверлейними файлообмінними мережами, мають ряд специфічних особливостей по відношенню до загальної IP-мережі. Так, у відомих файлообмінних додатках (Bit Torrent) оптимізація продуктивності мережі зводиться до пошуку «кращого» піру, з якого завантажується вміст.

Зупинимося на інших важливих для підвищення продуктивності мережі завданнях, які необхідно вирішувати паралельно в реальному часі:

- алгоритми планування фрагментів, що забезпечує можливість гладкого відтворення,
- вибір піру з низькими затримками,
- забезпечення взаємодії між динамікою оверлею і IP-маршрутизацією.

Користувачі (піри) P2P-мереж можуть завантажувати порції даних як від сервера, так і один від одного. При обміні буферними картами кожен користувач отримує інформацію про те, які порції даних є у того чи іншого сід-користувача і які він готовий завантажити. Таким чином, одночасно є m пропозицій для завантаження потрібної інформації. Кількість цих пропозицій може бути різною і обмежується відповідним протоколом планувальника або числом пірів, що пропонують запитану інформацію і утворюють той чи інший кластер. Разом з тим, кількість пропозицій не може бути надмірно великою, бо при цьому надмірно завантажується мережа і падає пропускна здатність. У зв'язку з цим модуль оверлею зазвичай обмежує число можливих з'єднань. Контрольний модуль при цьому виступає в ролі менеджера по відношенню до планувальника і оверлея, забезпечуючи моніторинг обміну відео- і сигнальною інформацією. Кожен медіапотік в мережі розділяється на фрагменти, розміри яких можуть бути рівними або різними, в залежності від протоколу P2P. При цьому динаміка процедури скачування, крім інших причин (кількості і якості запропонованих варіантів, фрагментів, обмежень пропускної здатності, швидкості скачування

контентів (download) і швидкості його завантаження (upload) і ін.) залежить від того, наскільки чітко здійснюється впорядкування самих фрагментів.

Постановка задачі.

Моніторингу підлягають: вектор параметрів $X_L^N = (x_1, x_2, \dots, x_n)$ спостережуваних на інтервалі $t = 1, 2, \dots, N$ запропонованих $i = 1, 2, \dots, m$ фрагментів, точки зміни їх стану та матриця умовних ймовірностей Q переходів зі стану в стан.

Процес скачування є дискретно-безперервним векторним процесом:

$$\vec{x}(k+1) = f(k, \vec{x}(k), \vec{u}(k)), \quad k \in K = \{k_1, k_2, \dots, k_F\},$$

де $\vec{x}_i^k = (x_{k1}, x_{k2}, \dots, x_{km})$ – послідовність вибраних фрагментів, які утворюють скачуваний файл,

$x_{ki} = (x_{k1}, x_{k2}, \dots, x_{km})$, $i = I \in \{1, 2, \dots, m\}$ – вибірка на кожному k -м кроці з m -класу запропонованих фрагментів,

$u(k)$ – управління процесом відбору фрагментів на кожному з k -кроків.

Якість і час доставки і тривалість кожного з фрагментів випадкові і різні.

Для забезпечення рівного програвання необхідно впорядкувати послідовність фрагментів в буфері відповідно до того, як вони надходять споживачеві. Крім того між фрагментами не повинно бути пауз і вони не можуть набігати на сусіда. Для цього необхідно на кожному з k -кроків відібрати той фрагмент з m - запропонованих, який буде володіти мінімальною затримкою, максимальною швидкістю передачі й ін.

Очевидно, в якості математичної моделі оброблюваного файлу може бути використано рівняння авторегресії, коефіцієнти якої φ_i , $i = (0, m)$ управляються марковським ланцюгом з m - станами:

$$x_t = \varphi_0(h_t) + \sum_{i=1}^m \varphi_i(h_t) x_{t-i} + b(h_t) \xi_t,$$

де ξ_t – вибірка з гауссового білого шуму, $b(h_t)$ – рівень шуму,

$H_{iN} = h_1, h_2, \dots, h_N$ стану послідовності, що управляє параметрами зсуву і масштабу відповідних фрагментів.

$p(h_t / h_{t-1}) = q(h_{t-1}, h_t)$ – умовна ймовірність переходу зі стану h_{t-1} в стан h_t , елементи матриці переходів $Q = \{q(h_{t-1}, h_t)\}$.

Логічно поставити задачу знаходження оптимальних значень станів $\hat{H}_{iN} = \hat{h}_1, \hat{h}_2, \dots, \hat{h}_N$, які упорядковують послідовність фрагментів відповідно до вимог упорядкування. В якості критерію оптимізації виберемо такий набір параметрів h_i , при якому на інтервалі буде забезпечуватися мінімум функціоналу:

$$Y(H_0^N) = d_0(h_0) + \sum_{t=1}^n [\alpha_t(h_t) + \alpha(h_{t-1}, h_t)],$$

де $d_0(h_0)$ – вихідне, ймовірне значення нульового відліку приналежності до даного класу вибірки h_0^i , $i = 1, 2, \dots, m$,

$$a_t(h_t) = \frac{1}{2b^2(h_t)} \left[x_t - \varphi_0(h_t) - \sum_{i=1}^n \varphi_i(h_t) x_{ti} \right]^2 - biq(h_{t-1}, h_t),$$

$$\alpha(h_{t-1}, h_t) = \ln \frac{q(h_t, h_{t-1})}{q(h_{t-1}, h_t)} = \ln \frac{p(h_t / h_{t-1})}{p(h_{t-1} / h_t)},$$

Очевидно при $h_{t-1} = h_t$ величина $\alpha(h_{t-1}, h_t) = 0$, в інших випадках α має сенс штрафної функції, що залежить від різниці і розбіжності сусідніх відліків фрагментів.

Величина $a_t(h_t)$ має сенс неузгодженості значень параметрів фрагмента на ділянці $t - n \leq s \leq t$ з прогнозованими.

Рішення даної оптимізаційної задачі зводиться до методів динамічного програмування або цілочисельного лінійного програмування (ЦЛП).

У припущенні гауссовості закону розподілу коефіцієнтів авторегресії умовна апіорна ймовірність появи комбінації $H_t^N = (h_1, h_2, \dots, h_N)$ визначається перехідними ймовірностями марківського ланцюга [56]:

$$\begin{aligned} f(\chi_1^N / \chi_{-n+1}^0, H_1^N) &= \prod_{t=1}^N f(x_t / x_{t-1}, h_t) = \\ &= \prod_{t=1}^N \frac{1}{b(h_t) \sqrt{2\pi}} \exp \left\{ -\frac{1}{2b^2(h_t)} \left[x_t - \varphi_0(h_t) - \sum_{i=1}^n \varphi_i(h_t) x_{t-i} \right]^2 \right\}. \end{aligned}$$

Логічно припустити гаусівський характер щільності ймовірності початкового значення $d_0(h_0)$, яке можемо представити у вигляді $d_0(h_0) = -\ln p(h_0)$, де $p(h_0)$ – апріорні ймовірності можливих вихідних станів марківського ланцюга. В цьому випадку мінімум функціоналу $Y(H_0^N)$ відповідає максимуму апостеріорної ймовірності комбінацій фрагментів для даної реалізації [57]:

$$\begin{aligned} \hat{H}_0^N &= \arg \min_{H_0^N} Y(H_0^N) = \arg \max_{H_0^N} \ln p(H_0^N / \chi_1^n) = \\ &= \arg \max_{H_0^N} [\ln p(h_0) + \ln p(H_1^N / h_0) = \ln f(\chi_1^n / \chi_{n+1}, H_1^N)] \end{aligned} \quad (3.19)$$

Тут $p(H_1^N / h_0) = \prod_{t=1}^N q(h_{t-1}, h_t)$ – умовна апріорна ймовірність появи комбінації фрагментів $X_i^N = (x_1, x_2, \dots, x_n)$, складених із m – запропонованих різними пірами варіантів.

Побудова оптимальної фрагментації. Визначимо послідовність функцій з урахуванням тимчасового дискретного аргументу:

$$\begin{aligned} d_t(h_t) &= \min_{H_0^{t-1}} \left\{ d_0(h_0) + \sum_{s=1}^{t-1} a_s(h_s) + \alpha(h_{s-1}, h_s) + a_t(h_t) + \alpha(h_{t-1}, h_t) \right\}, \\ t &= 1, 2, \dots, N \end{aligned} \quad (3.20)$$

Дана величина $d_t(h_t)$ показує, яке мінімальне значення критерію $Y(H_0^N)$ може бути досягнуто на послідовності фрагментів до $(t-1)$ -го включно, якщо зафіксувати стан h_t останнього фрагмента. Очевидно, що мінімальна величина останнього значення $d_N(h_N)$, $h_N = 1, 2$ збігається з мінімальним значенням критерію $Y(H_0^N)$:

$$\min d_N(h_N) = \min Y(H_0^N). \quad (3.21)$$

Якщо останній фрагмент оптимальної послідовності H_0^N , то

$$\hat{h}_N = \arg \min d_N(h_N) \quad (3.22)$$

Обчислимо послідовно вектори значень $d_{t+1}(h_{t+1})$ для відповідних моментів часу $t = 0, 1, \dots, N-1$ починаючи з $d_0(h_0)$ за правилом:

$$d_{t+1}(h_{t+1}) = \min_{h_t} [d_t(h_t) + a_t(u_t) + \alpha(h_t, h_{t+1})] \quad (3.23)$$

При цьому цілочисельні величини

$$K_{t+1}(h_{t+1}) = \arg \min \{d_t(h_t) + a_t(u_t) + \alpha(h_t, h_{t+1})\} \quad (3.24)$$

утворюють прямокутну матрицю K_i^N , в якій на N шпальтах розміщуються впорядковані за часом значення $\hat{H}_0^N = (\hat{h}_0, \hat{h}_1, \dots, \hat{h}_N)$. Кількість рядків відповідає значенням m , представленим для вибору на кожному кроці фрагментації.

Відповідні значення \hat{h}_t знаходяться з рекурсивної формули

$$\hat{h}_t = K_{t+1}(\hat{h}_{t+1}) \text{ з початковою умовою (3.22).}$$

Таким чином, оцінка оптимальних значень \hat{h}_t приймається у зворотному порядку: $t = N, N-1, \dots, 1, 0$ після обчислення в прямому порядку елементів – стовпців матриці K_i^N , які містять mN цілих чисел – медіа фрагментів.

Про завдання вибору варіантів з теорії рішень відомо [59], що завдання оптимальної фрагментації надзвичайно громіздке, відноситься до NP-повних і зі збільшенням числа пропонувананих варіантів вибору m виявляється проблематичним отримання рішення в реальному часі. Цілком доступними є рішення при $m = 1; 2; 3$. З теорії автовибору також відомо [40], що перехід від одноразового вибору до двократного дозволяє зменшити ймовірність помилки P_{oui} з $P_{oui} = 10^{-2}$ до $P_{oui} = 10^{-4}$. Подальше збільшення кратності привносить все менш відносне зростання ефективності в прийнятті рішень (рис.3.4). Отже, з метою недопущення перевантаження в рішенні завдання, зупинимо вибір значенням $m = 2$, що для задач фрагментації цілком прийнятне.

З урахуванням $m = 2$ залишається уточнити значення функцій під знаком суми в вираженні функціоналу (3.40), куди будуть входити лише два аргументи.



Рис. 3.4. Залежність ймовірності помилки P_{err} від співвідношення рівня сигналу до спектральної потужності шуму h^2

3.3. Аналіз стану контенту на вузлах P2P-мереж

Успішне отримання заданої послуги в P2P-мережах цілком залежить від того, наскільки швидко і якісно вдається знайти необхідний контент, що зберігається в пам'яті піра даної мережі. Реплікація контенту, отже, відіграє визначальну роль для успішного пошуку. Очевидно, наявність великої кількості копій у мережі дозволить прискорити процес пошуку потрібного файлу. Однак, з іншого боку це зажадає більше пам'яті для зберігання контенту. Розглянемо більш детально механізми реплікації з точки зору компромісу між ефективністю пошуку і вартістю зберігання контенту.

У розглянутих неструктурованих мережах P2P використовується по суті випадковий пошук. При цьому в окремих мережах (Gnutella) піри, що запитують файл, здійснюють випадковий пошук і копіюють ці файли. В інших мережах (FastTrack) забезпечується проактивна реплікація.

Припустимо, що в деякій мережі P2P є m -файлів і n -вузлів. Кожен i -й файл $i = (1 \leq i \leq m)$ реплікується на випадкових пірах $r_i = (1 \leq r_i \leq n)$. Припустимо, що обсяг пам'яті для всіх пірів фіксований і становить величину

$$R = \sum_{i=1}^m r_i \quad (3.25)$$

Ймовірності запитів кожного з i файлів однакові і утворюють повну групу подій:

$$P = \sum_{i=1}^m P_i = 1$$

Ймовірність того, що файл успішно знаходиться на k -й спробі:

$$P_i(k) = \frac{r_i}{n} \left(1 - \frac{r_i}{n}\right)^{k-1}$$

Середня кількість спроб при пошуку потрібних файлів визначається стандартною формулою оцінки середнього

$$N = \sum_{i=1}^m P_i N_i = n \sum_{i=1}^m \frac{P_i}{r_i},$$

де враховано, що середнє число запитів для одного файлу N_i складе величину n/r_i .

Проаналізуємо продуктивність трьох відомих алгоритмів реплікації.

1. Уніфікований алгоритм. У цьому алгоритмі реплікуються всі файли, незалежно від їхньої популярності, тобто

$$r_i = \frac{R}{m},$$

так що середня розмірність пошуку

$$N_y = n \sum_{i=1}^m P_i \frac{m}{R} = \frac{nm}{R}.$$

2. Пропорційний алгоритм відрізняється тим, що при реплікації створюється більше копій для більш популярних файлів, для чого враховується їх імовірнісна міра

$$r_i = RP_i,$$

Отже, середній розмір пошуку

$$N_n = n \sum_{i=1}^m \frac{P_i}{RP_i} = \frac{nm}{R}, \quad (3.26)$$

тобто обидва розглянуті алгоритми дають однаковий середній розмір пошуку.

3. Алгоритм реплікації квадратного кореня.

З використанням (3.26) одержуємо:

$$r_i = \frac{R\sqrt{P_i}}{\sum_{j=1}^m \sqrt{P_j}}$$

Звідси середнє значення:

$$N_k = \frac{n}{R} \left(\sum_{i=1}^m \sqrt{P_i} \right)^2$$

На рис. 3.5. представлені графік середнього розміру пошуку в залежності від числа заявлених файлів m . При цьому враховано, що відхилення загального числа пірів до загального обсягу $(N/R) = const$. Очевидно також, що $n > m$, оскільки число заявлених файлів не може перевищувати числа пірів в мережі P2P.

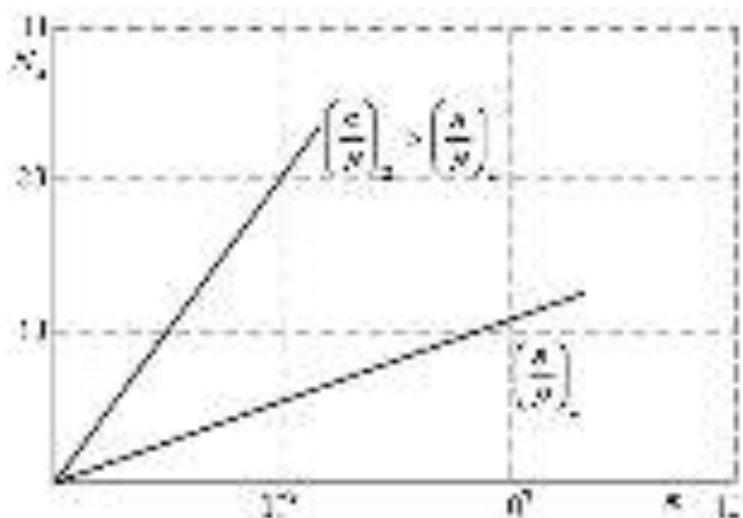


Рис. 3.5. Графік середнього розміру пошуку в залежності від числа заявлених файлів

На рис. 3.6. представлені графіки середнього розміру пошуку в залежності від ймовірності запитів P_i , де $P_i = const$ для всіх i .

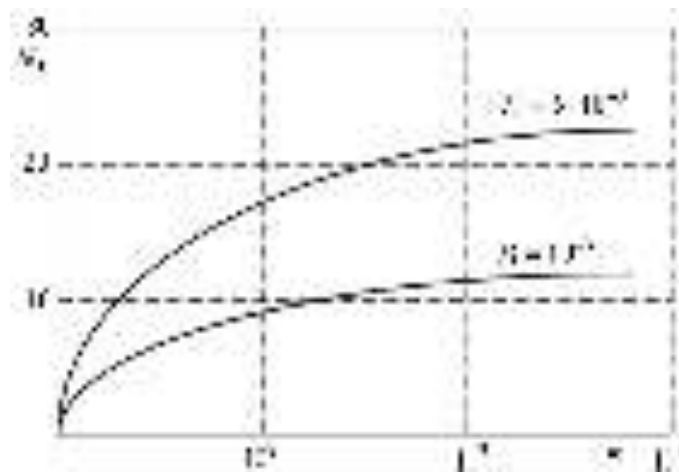


Рис. 3.6. Графік середнього розміру пошуку в залежності від числа заявок для алгоритму реплікації «квадратного кореня»

Як випливає з графіків, зі збільшенням заявлених файлів зростає число запитів у мережі. Однак для двох алгоритмів: уніфікованого і пропорційного цього зростання немає, тоді як для «кореня квадратного», де враховується попит на популярні файли, це зростання нелінійне і більш економне зі збільшенням m .

У неструктурованих P2P-системах, на відміну від структурованих адрес і контент, що зберігаються до конкретного вузла, не пов'язані між собою. Пошук потрібної інформації в цих мережах здійснюється на основі двох рішень:

- сліпого лавинного пошуку (Flooding), при якому запити надсилаються в усі підключені вузли на предмет наявності в них запитаних файлів (Gnutella),
- інформаційний пошук, при якому одноранговим вузлам надається додаткова інформація про топологію мережі й про розташування ресурсів.

Перевага сліпих лавинних методів у тому, що вони не потребують будь-яких службових заголовків для підтримки додаткових індексів про розташування ресурсів і мають високу стійкість в умовах динамічних навантажень. На відміну від цього інформаційні методи пошуку дозволяють вузлам підтримувати додаткову інформацію про інші вузли мережі, їх топології та розташування ресурсів. Суть інформаційних методів у тому, що в вузлах передбачається збір і зберігання топологічної і статистичної інформації про своїх сусідів, що дозволяє

оптимізувати топологію, використовувати адаптивні алгоритми для пошуку (Adaptive Probabilistic Search – APS). Кожен вузол при цьому зберігає індекс маршрутизації, який має детальну інформацію про зміст файлів. На основі індексів маршрутизації (Routing Indexes) на вузлах створюються таблиці маршрутизації запитів через кешування ключових слів файлу. Це дозволяє самоорганізуватися мережі при пошуку потрібних ресурсів. Таким чином при інформаційному пошуку вдається різко збільшити продуктивність виявлення потрібних ресурсів.

3.4. Висновки до третього розділу

1. З використанням марківської властивості процесу скачування побудована модель n -вузлової мережі, яка враховує на кожному k -м кроці вибір кращого з m запропонованих фрагментів для збирання запитаного файлу. Модель може бути віднесена до класу базових для мультиагентних мереж. Доведено, що продуктивність пірингової мережі багато в чому залежить від того, наскільки оперативно виявляється потрібна інформація і проводиться вибір кращого фрагмента з можливої множини запропонованих варіантів. Обґрунтовується використання вибору за принципом «одного з двох» запропонованих фрагментів, при цьому ймовірність помилкового рішення в порівнянні з одиночним рішенням знижується на величину до 17-20 дБ.

2. Рівень динаміки пірингової мережі характеризується співвідношенням швидкостей скачування (download) контенту і його завантаження (upload). Отримане рівняння балансу з урахуванням швидкості завантаження відеосервера. Одночасно з цим відношення швидкостей може характеризувати ступінь альтруїстичної стратегії пірів в даній P2P-системі.

3. Розроблено методику аналізу якості роботи пірингової мережі при різному навантаженні. Показано, що з ростом числа активних пірів якість P2P-мережі збільшується, що пояснюється зростанням обсягу пам'яті й кількості пропонованого контенту. Доведено також, що в умовах лавинного масового

підключення нових пірів, що буває в години популярних подій, може спостерігатися зниження якості роботи мережі, що пов'язано з тимчасовим браком контенту для завантаження. Оскільки швидкість пірів, які підключилися, перевищує швидкість росту пам'яті мережі.

4. У неструктурованих мережах P2P пошук необхідного контенту безпосередньо залежить від обсягу пам'яті й реплікації контенту даної мережі, що забезпечує її максимальну продуктивність і стійкість. Оскільки в цих мережах, на відміну від структурованих, адреса і контент, що зберігаються в конкретному вузлі, не пов'язані між собою, пошук потрібної інформації здійснюється на основі двох рішень: сліпого лавинного пошуку (Flooding) й інформаційного пошуку, що дозволяє на основі збереження індексу маршрутизації отримати дані про топологію і стан сусідів. Ця технологія має також більшу масштабованість і надійність, що робить її придатною для гетерогенного і динамічного середовища Інтернет.

5. Розроблено алгоритм і методику аналізу оцінки середньої кількості спроб запиту для успішного отримання необхідного контенту в залежності від обсягу ресурсів мережі. Показано, що серед відомих методів реплікації найбільш ефективним є метод «квадратного кореня», де враховується попит на популярні файли.

6. Стосовно до моделі дискретно-різницевої динаміки процедури скачування фрагментів синтезований алгоритм управління у вигляді послідовності релейних впливів, що враховує стан j -го піра на кожному з k - кроків фрагментації з урахуванням затримки доставки фрагментів.

7. Розроблено математичну модель процесу відбору скачуваних файлів на базі дискретно-безперервного випадкового векторного процесу авторегресії. З використанням даної моделі вирішена задача оптимальної побудови процедури знаходження вектора рішень щодо послідовного відбору фрагментів запитаного файлу з m - пропозиції. Рішення задачі у загальному вигляді зводиться до процедури динамічного програмування.

РОЗДІЛ 4

МЕТОДИ УПРАВЛІННЯ ДИСКРЕТНО - НЕПЕРЕРВНОЮ МОДЕЛЛЮ СКАЧУВАННЯ ФРАГМЕНТІВ

Особливістю P2P-мереж є процедура залучення машин інших клієнтських користувачів, з яких відбувається скачування відповідних відеофрагментів. У живому потоковому відео TV-контент поширюється усім користувачам у режимі реального часу. В процесі надання послуги приймається ряд рішень, здійснюється значна кількість управлінь: вибір піру, приєднання і від'єднання піру, обмін інформацією між пірами, оцінка затримок, організація буфера, планування фрагментів. Відповідні процедури виконують цілий ряд функцій управляючих машин: сервер, глобальний і локальні трекери та ін. Планувальник забезпечує необхідну координацію всіх протоколів і зазначених органів управління відповідно до своїх алгоритмів і вибраних моделей.

Продемонструємо можливість вибору об'єднаної математичної моделі $x(t)$ -процесу функціонування пірингової мережі й відповідної універсальної процедури оптимального управління в технології P2P-TV. Процес скачування будь-якого файлу $x(t)$ супроводжується розбиттям його на окремі фрагменти, одержувані в загальному випадку від різних пірів. Планування процесу фрагментації здійснюється таким чином, щоб за мінімальний час $t \in T = [t_1, t_F]$ забезпечити надання послуги з необхідною якістю. Сам процес $x(t)$ на інтервалі T є керованою послідовністю окремих фрагментів з незалежними значеннями на кордонах стику. Дане завдання відноситься до класу магістральних, коли управління транспортом здійснюється за обраними ділянками (етапами) і на кожному етапі вибирається найкращий режим швидкості, мінімізації затримок, рівня витрат [18,19].

Математична модель процесу управління скачування файлу в піринговій мережі.

Опис системи. Елементи пірингової мережі, які беруть участь в скачуванні файла, є системою, що складається з одного скачуваного об'єкта і одного або декількох об'єктів, що пропонують відповідні фрагменти для запису в буфер. Така система обслуговування є зазвичай керованою і носить назву одностадійної [48]. При цьому очевидно, в якому конкретному випадку змінюється динаміка оверлею, положення фрагментів, умови їх доставки. З огляду на це кожен фрагмент має самостійну історію.

Фрагменти можуть прибувати в піри в різному порядку, тому кожен вузол потребує буфера фрагментів і відповідного алгоритму з метою зміни порядку їх проходження і впорядкування на осі часу, що забезпечує подальше гладке відтворення. Дані завдання вирішуються в модулі планувальника, а сам процес прийняття рішень розбивається на дві фази: вибір піру для зв'язку, який володіє відповідною інформацією і вибір фрагмента для відправки.

Розмір фрагмента зазвичай становить кілька секцій (сотні-тисячі IP-пакетів) [18]. Використання таких великих фрагментів дозволяє значно мінімізувати службовий трафік порівняно з фактичним часом передачі фрагментів, зменшити час, необхідний для дифузійної передачі змін в інформації про стан буфера фрагментів інших пірів. Крім того, такі фрагменти можуть бути передані в межах технології TSP, що забезпечує доставку і контроль перевантаження для окремих передач. Разом з тим, великі фрагменти сприяють і більшим затримкам.

Таким чином, завданням планувальника є оперативне обслуговування вимог конкретного піру. Переривання при обслуговуванні та обслуговування більше одного піру – заборонені. Задається тривалість обслуговування і директивний термін його закінчення. Простої при обслуговуванні вимог також заборонені. Всі обмеження є предметом вивчення теорії розкладів.

Основні відомості з теорії розкладів.

Основною задачею теорії розкладів є оптимізація впорядкування обслуговування заданого числа вимог і тривалості. Додатковими є обмеження:

- моменти готовності до обслуговування,
- директивні терміни, до яких необхідно завершити обслуговування,

– функції штрафів (ваговий коефіцієнт), що визначає важливість тієї чи іншої вимоги або ступінь терміновості його обслуговування.

Більшість відомих результатів з теорії розкладів зводиться до вирішення задачі оптимізації упорядкування вимог із незалежними умовами. Однак задачі, що виникають на практиці часто характеризуються тим, що обслуговування вимог здійснюється при заданих обмеженнях передування, що виникають у системі різних технологічних умов.

Для задач теорії розкладів загальноприйнятим є трипозиційне позначення, яке визначає стан системи, що оптимізується [53]:

$$\alpha | \beta | \gamma,$$

де α – описує систему обслуговування (цифра позначає кількість приладів),
 β – характеризує умови обслуговування (наприклад, наявність моментів готовності вимог до обслуговування, умови вказування (prec), дозволу переривання, директивні терміни й ін.),

γ – вказує вид цільової функції, що оптимізується.

Є ряд методів і рішень задачі розкладів, кожен із яких передбачає вибір моделі, критерію оптимізації (функції штрафів) та відповідних обмежень. Найбільш загальним рішенням даного завдання є рішення на основі дискретно-неперервної моделі процесу скачування фрагментів. Більш приватними є чисто дискретні моделі, які на нашу думку в ряді випадків є більш конструктивними і адекватними ситуації.

Задача мінімізації максимального штрафу полягає в побудові розкладу, якому відповідає мінімум функціоналу

$$F_{\max} = \max_{1 \leq j \leq n} f_j(c_j),$$

де $f_j(c_j)$ – неубутна функція, що задає штраф за завершення обслуговування вимоги j в момент часу C_j . В [53] показано, що при неодночасному надходженні вимог задачі $1/prec, r_j / L_{\max}$ або $1/prec, r_j / F_{\max}$ є NP-важкою (параметр r_j в позиції β вказує на те, що для кожної вимоги j заданий момент готовності до

обслуговування – момент надходження вимоги в чергу на обслуговування. Очевидно така постановка повністю збігається з вимогою впорядкування фрагментів у буфері й мінімізації затримки кожного з них, що забезпечить безшовне відтворення.

Разом з тим є ряд алгоритмів рішень [19] задачі оптимізації за $O(h^2)$ операцій: спочатку знаходять модифіковані моменти готовності вимог, що не мають нащадків (послідовників) з урахуванням заданого часткового порядку, а потім, на кожному кроці, будуючи розклад із його кінця до початку, знаходять вимогу з найменшим значенням функції штрафів у поточний момент завершення обслуговування всіх вимог і призначають його на обслуговуванні, встановлюючи інтервал з урахуванням можливих переривань процесу обслуговування.

Сам процес управління фрагментами $x(t)$ в загальному випадку являє собою дискретно-безперервну процедуру, де дискретність визначається межами фрагментів, а протягом кожного з фрагментів визначено стан $x(k)$, який надається безперервною функцією. Дискретно-безперервні процеси (ДНП) не піддаються безпосередньому опису в термінах звичайних диференціальних рівнянь. Для них існують деякі узагальнення безперервного опису ДНП [18].

Для побудови, моделі ДНП в якості вихідної приймається різницеве рівняння динаміки якоїсь абстрактної керованої системи [19]:

$$x(k+1) = f(k, x(k), u(k)), \quad k \in K = \{K_j, K_{i+1}, K_F\},$$

$$u(k) \in U(k, x(k)),$$

де k – номер кроку дискретної послідовності, що визначається положенням чергового фрагмента на осі часу,

x, u – змінні інтерпретовані як стан і управління ДНС

$U(k, x(k))$ – безліч, задане при кожному k і x .

Управління протягом викачуваного файлу будемо розглядати у вигляді сукупності $u = (u^d, m^c)$, де $u^d(k)$ – дискретне управління довільної природи, а $m^c(k)$ – деякий безперервний керований процес $(T(k), m^c(k, t), u^d(k, t))$. При цьому

безліч управлінь $U(k, x)$ при фіксованих параметрах k, x, u^d представляє безперервний процес з додатковими обмеженнями:

$$\frac{dx^c(t)}{dt} = f^c(z, t, x^c, u^c), \quad z = (k, x, u^c), \quad t \in T(z) = \{t_i, t_F\}, \quad u^c \in U^c(z, t, x^c) \subset R^{p(k)},$$

$$\gamma^c = (t_i, x^c(t_i), t_F, x^c(t_F)) | \Gamma^c(z),$$

де $x^c(k, t)$ – передбачається кусочно-гладким,

$u^c(k, t)$ – кусочно-безперервний процес при кожному k .

Рішенням цієї гібридної схеми вважається набір $m = (x(k), u(k)) \in D$, $u(k) = (u^d(k), m^c(k))$, $m^c(k) \in D^c(k, x(k), u^d(k))$.

Дану модель можна розглядати як дворівневу керовану систему. Нижній рівень являє собою опис безперервних керованих процесів на окремих фрагментах. Верхній рівень пов'язує ці описи в єдиний процес, таким чином, у задачі оптимізації обидва рівні розглядаються у взаємодії.

Для отриманої моделі завдання оптимального управління вирішується в стандартній формі, як завдання про мінімум на D функціоналу $I = F(x(t_F))$ при фіксованих $k_i, k_F, x(k_i)$ і додаткових обмеженнях $x(t_F) \in \Gamma$, $x(t_i) \in \Gamma$, де $X(k), \Gamma$ – задані множини.

Оператор правої частині для $x(k+1)$ при $k \in K' \subset K$ має вигляд $f(k, x, u) = \theta(z, \gamma^c)$. Це означає, що перехід з нижнього рівня на верхній відбувається через кордон γ^c між сусідніми фрагментами для безперервного процесу.

Достатні умови оптимальності для отриманої моделі є аналогом умов Кротова для дискретних і безперервних систем, при цьому з обмежень множин D і D^c виключаються дискретна послідовність і сама диференціальна система. Цим самим виходить розширена множина E дискретно-безперервних процесів. Вводяться функціонали $\varphi(k+1)$ і параметричне сімейство (з параметром z) гладких функцій $\varphi^c : R^{n(k+1)} \rightarrow R$. Далі рішення зводиться до побудови узагальненого лагранжиана [18,19].

Дещо інша математична модель і завдання оптимізації, що зводиться до мінімізації сумарного зваженого часу обслуговування дається в роботі [20], де мінімізується критерій $J = \sum_{j \in J} \omega_j C_j(\pi_t, p)$, де π_t – безліч станів фрагментів.

4.1. Опис функціональної математичної моделі дискретно-безперервного процесу скачування фрагментів

У результаті скачування файлу спостерігаємо загальний дискретно-неперервний (ДБ) процес x_n^δ , який представляє собою клас вироджених випадкових процесів, що володіють послідовністю точок розриву. Для знаходження траєкторії процесу й рішення рівнянь стану $dx(t)/dt$ характерно необмежене число значень похідних [20,21]. Типовим методом вирішення таких ДБ завдань є заміна послідовності $x_n^\delta(t)$ якоюсь апроксимальною розширювальною послідовністю $x_n^\delta(k, t)$ [26], що дозволяє на множині дії безперервних значень формувати неявно оператор дискретної компоненти x_δ .

Процес формування файлу можна представити у вигляді послідовності безперервних ділянок, що заповнюють простір між (k) і $(k+1)$ – сусідніми межами фрагментів. Модель такого ДБ процесу протягом скачування файлу $[t_1, t_F]$ можна представити у вигляді керованого різницевого рівняння стану [27]:

$$x(k+1) = f(k, x(k), u(k)), \quad (4.1)$$

де $k \in k = 1, 2, \dots, k_F$, $u(k) \in U(k, x(k))$.

Такі дискретно-неперервні моделі (ДНМ) у відомій літературі [26] представляються у вигляді дворівневої структури, яка характеризується послідовністю дискретних значень $x(k) = x_\delta$, $k = 1, 2, \dots$, що інтерпретується як процес верхнього рівня і послідовністю ділянок нижнього рівня безперервного процесу $x_n(t)$, що заповнюють інтервали між черговими дискретами $x(k)$ і $x(k+1)$ (рис.4.1.).
Перехід процесу з нижнього на верхній рівень відбувається через межу:

$$\gamma = (t_1, x_n(t_1), t_F, x_n(t_F)).$$

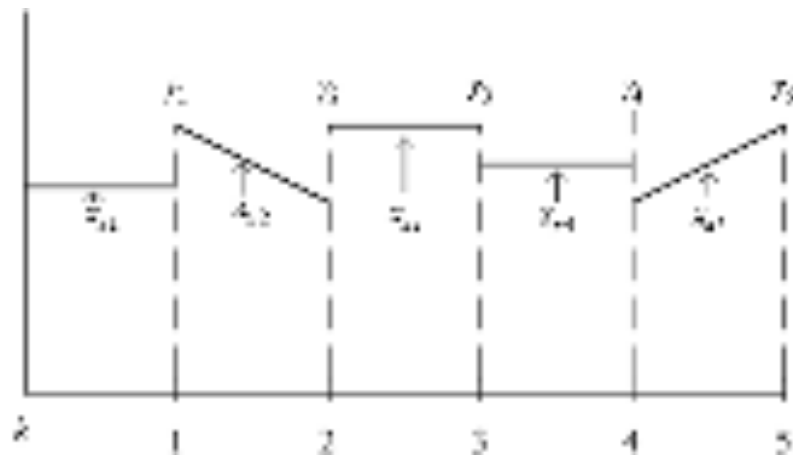


Рис. 4.1. Приклад реалізації дискретно-безперервного процесу складається з 5 фрагментів

Управління процесом $x_n^o(k)$ слід розглядати у вигляді двох окремих функцій:

$u_o(k)$ – управління дискретністю фрагментів,

$u_n(t)$ – управління безперервним потоком даних самого TV-контенту.

Даний гібридний процес $x(k+1)$ є виродженим, оскільки на стику фрагментів мають місце розриви. Для конкретизації даної моделі представимо управління послідовністю фрагментів у вигляді дискретності значень u_o , а передачу даних зображення, як кусочно-гладкий процес

$$dx_n(t)/dt = f_n(z, t, x_n, u_n), \quad t \in T(z) = [t_1(z), t_F(z)] \quad (4.2)$$

Управління $u_n(k, t)$ при цьому є кусочно-неперервною функцією:

$$u_n \subset U_n(z, t, x_n),$$

де $t \in T(z) = [t_1(z) .. t_F(z)]$, $z = (k, x, u_o)$, $k \in K'_n \subset K$ – сукупність змінних, що відіграють роль параметрів при реалізації неперервного управління всередині фрагментів.

При переході x_n через межу сусідніх фрагментів використовується оператор переходу γ_n , за допомогою якого значення похідної (4.2) набуває вигляду

$$f_n(z, t, x_n, u_n) = \theta(z, \gamma_n).$$

Представлена модель (4.1) і відповідні її конкретизації (4.2), дозволяють вирішувати завдання оптимізації процесу управління [28]. Управління може бути побудовано з використанням різних оптимальних ітераційних процедур, до числа яких можна віднести метод динамічного програмування, метод множників Лагранжа, методи цілочисельного лінійного програмування (ЦЛП).

Процес доставки послідовності фрагментів споживачеві розглядається як компонент загального процесу (4.1). Він лежить в основі функціонування мережі P2P-TV і являє собою дискретну керовану модель, стан якої визначено на кожному з k -кроків

$$x_{\partial}(k+1) = A(k, u_{\partial})x_{\partial}(k) + B(k, u_{\partial}), \quad (4.3)$$

де $x_{\partial} \in R^{m(k)}$; $u_{\partial} \in U(k) \subset R^{p(k)}$; $k \in K = \{k_1, \dots, k_F\}$.

Дискретна компонента процесу $x_{\partial}(k)$ являє собою кінцеву (k_F) для даного скачуваного файлу послідовність фрагментів, одержуваних від різних пірів, відповідно до обраного плану.

Управління $U_d(t)$ – забезпечує процес фрагментації, вибір довжини фрагментів, відповідне їх впорядкування та коректне «зшивання», що дозволяє споживачеві спостерігати безперервне гладке зображення.

4.2. Оптимізація процесу управління в пірингової мережі

Завданням управління $u(t)$, здійснюваного в межах кожного з m -фрагментів, є контроль кадрів, управління швидкістю, мінімізація затримки.

Завданням управління процесом завантаження в загальному випадку є багатокритеріальною, багатокроковою процедурою, визначеною на кінцевій множині кроків k_F . Рішення завдання при зазначених обмеженнях і вибраних моделях (4.1), (4.2), (4.3) зводиться до мінімізації функціоналу

$$J = \Phi(x(k_F), u(k_F))$$

або при відповідній лінеаризації:

$$J = \phi(u(k_F)x(k_F)).$$

Оптимальне управління таким дискретно-неперервним процесом знайдемо за допомогою узагальненого лагранжиана [5,6]

$$L = G(x(k_F)) - \sum_{K \setminus K' \setminus t_F} R(k, x(k), u(k)) + \sum_{K_1}^K (G^n(z(k), \gamma^n(k)) + \int_{T(z(k))} (\mu^n(z(k) - R^n(z(k), t, x^n(k, t), u^n(k, t)))) dt) \quad (4.4)$$

де (φ, φ^n) – пара довільних функцій, що забезпечують розширення дискретного і безперервного процесів,

$$G(x) = \Phi(x) + \varphi(k_F, x(k_F)) - \varphi(k_1, x(k_1)),$$

$$R(k, x, u) = \varphi((k+1), f(k, x, u)) - \varphi(k, x),$$

$$G^n(z, \gamma^n) = -\varphi((k+1), f(k, x, u)) + \varphi(k, x(k)) + \varphi^n(z, t, x_F^n) - \varphi^n(z, t_1, x_1^n) - \int_{T(z)} \mu^n(z, t) dt,$$

$$R^n(z, t, x^n, u^n) = \varphi^n(T, x^n) f^n(z, t, x^n, u^n) + \varphi^n(z, t, x^n),$$

$$\mu^n(z, t) = \sup \{ R^n(z, t, x^n, u^n) \},$$

$$l(k) = \inf \{ G^n(z, \gamma^n) \},$$

$$z = (k, x, u^\delta).$$

K' – підмножина апроксимальних значень на безперервній множині процесу $x_n(t)$, $K' \subset K$.

У результаті представленої оптимізаційної процедури знаходиться мінімум функціоналу L . Показано [25], що одночасно з цим забезпечується $\min J(k, x, u)$.

Для вирішення (4.4) необхідно задати пару розширювальних функцій (φ, φ^n) , що забезпечує виконання умови [25]

$$\inf L = 0$$

для всієї сукупності управляючих функцій u_δ, u_n і відповідної сукупності станів верхнього x_δ і нижнього x_n - рівнів. Дані вимоги забезпечують конкретність умов оптимальності процедур типу Беллмана і побудови ефективних ітераційних процедур [29].

Отримана досить універсальна дискретно-безперервна диференціальна модель процесів скачування P2P-TV-контенту з урахуванням процесів фрагментації, управління потоком фрагментів і управління потоком даних.

Показана можливість побудови процедур оптимального управління на основі узагальненого рішення Лагранжа.

Слід зазначити, що рішення задачі управління процесами $x_n^g(k)$ на основі узагальненого рішення Лагранжа є досить громіздким, більш того, у відомих технологіях Р2Р зазвичай обмежуються синтезом управління дискретною послідовністю u_g , при цьому функції управління потоком u_n виконується за рахунок мережевої синхронізації. Це дозволяє уникнути загального громіздкого рішення, обмежившись лише оптимізацією потоків фрагментів.

4.3 Оптимальне управління потоком фрагментів

4.3.1 Математична модель потоку фрагментів

Вихідний від джерела в мережу потік фрагментів визначається спільним розподілом випадкових величин $z_k = t_k - t_{k-1}$, де t_k – моменти надходження k -го фрагмента ($k \geq 1$, $t_0 = 0$, $0 \leq t_1 \leq t_2 \leq \dots$). Очевидно послідовність фрагментів z_k утворює випадковий потік, де для кожного піру відлікові значення z_k є незалежні в сукупності. Такий потік називають потоком із обмеженим післядією і для його визначення достатньо задати набір функцій розподілу $F_k(t) = P\{z_k \leq t\}$, $k \geq 1$.

Окремим випадком такого випадкового потоку є пуассонівський потік, для якого $F_k(t) = 1 - \exp\{-\lambda t\}$, де λ – інтенсивність потоку, що залежить від загального числа фрагментів, що передаються в мережі.

У силу специфіки пірингову мережу можна вважати однорідною, а експонентний характер функції розподілу ймовірностей допускає використання для її досліджень методів теорії масового обслуговування. Для мереж цього типу характерна мультиплікативна форма стаціонарних ймовірностей, яка дозволяє досить просто знаходити умови глобального і локального балансів, що становить основу для дослідження і проектування мережі.

Разом із тим, для побудови такої моделі потоку фрагментів при скачуванні відповідного файлу методи теорії масового обслуговування не підходять, оскільки в даному випадку має місце випадковий керований потік обслуговування, а не чисто випадковий. Знайдемо оптимальний план (управління) надходження фрагментів від різних пірів, який вказував би кожному i -му фрагменту час надходження на обслуговування:

$$U = \varphi_{nl}(t_1, t_2, \dots, t_n).$$

В процесі обслуговування необхідно врахувати наступну специфіку:

а) випадковий час надходження i -го фрагмента на обслуговування $t_i^0(\varphi_i)$, що відрізняється від якогось планованого t_i^{nl} .

б) випадковий час обслуговування i -го фрагмента $\theta_i(\varphi_i)$.

У результаті різних випадкових чинників та керівних впливів можливі дві причини втрат часу, що призводять до відповідних затримок:

– втрати, пов'язані з тим, що i -й фрагмент надійшов раніше, ніж завершилося зчитування j -го фрагмента; $\delta_i^0(\varphi_i, U)$;

– втрати при затримці надходження i -го фрагмента, якщо зчитувач простоює $\delta_i^{\Pi}(\varphi_i, U)$.

Сумарні втрати часу при обслуговуванні n -фрагменти складуть:

$$\Phi(U) = \sum_{i=1}^n (m\{\delta_i^0(\varphi_i, U)\} + cm\{\delta_i^{\Pi}(\varphi_i, U)\}) \quad (4.5)$$

де c – коефіцієнт, що враховує вартість втрат;

$m\{\cdot\}$ – знак математичного очікування.

Завдання оптимального планування процесу зчитування файлу складається в мінімізації функціоналу $\Phi(U)$.

Розглянемо k -й крок процесу. Обслуговування $k-1$ фрагмента завершується у випадковий момент t_{k-1} , функція розподілу ймовірностей якого:

$$F_{k-1}(t) = P\{t_{k-1} < t\}.$$

Фрагмент k надходить у випадковий момент t_k^0 , функція розподілу якого:

$$F_k^0(t) = P\{t_k^0 < t\}.$$

При цьому середнє:

$$\int_{-\infty}^{\infty} t dF_k^0(t) = t_k^{nl}$$

Обслуговування k -го фрагмента починається у випадковий момент часу:

$$t_k^H = \max(t_{k-1}, t_k^0)$$

з функцією розподілу:

$$F_k^H(t) = P\{t_k^H < t\} = P\{t_{k-1} < t, t_k^0 < t\}.$$

Значення випадкових величин t_{k-1} і t_k^0 породжуються різними процесами, пов'язаними відповідно з вибором $k-1$ піра, його віддаленістю і процесом зчитування, що дозволяє прийняти гіпотезу про їх незалежності, тому:

$$F_k^H(t) = F_{k-1}(t)F_k^0(t). \quad (4.6)$$

Час обслуговування k -го фрагмента θ_k є випадковою величиною, заданої функцією розподілу

$$\Phi_k(t) = P\{\theta_k < t\},$$

причому θ_k не залежить від часу каналу обслуговування.

Час кінця обслуговування k -го фрагмента t_k є композицією двох незалежних величин:

$$t_k = t_k^H + \theta_k$$

З функцією розподілу, яка визначається інтегралом згортки:

$$F_k(t) = \int_{-\infty}^{\infty} F_k^H(t-\tau) d\Phi_k(\tau). \quad (4.7)$$

При відомому розподілі $F_{k-1}(t)$ можна визначити втрати часу для фрагмента δ_k^0 і для зчитувача δ_k^{Π} на k -му кроці управління:

$$\delta_k^0 = \begin{cases} 0 & \text{нпу } t_k^0 \geq t_{k-1}, \\ t_{k-1} - t_k^0 & \text{нпу } t_k^0 < t_{k-1}; \end{cases} \quad m\{\delta_k^0\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\tau - t) dF_{k-1}(\tau) dF_k^0(t);$$

$$\delta_k^{\Pi} = \begin{cases} 0, & t_k^0 \leq t_{k-1}, \\ t_{k-1} - t_k, & t_k^0 > t_{k-1}; \end{cases} \quad m\{\delta_k^{\Pi}\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (t - \tau) dF_{k-1}(\tau) dF_k^0(t).$$

Сумарні втрати часу на k -му кроці залежать від вибору управління на k -му кроці від функції розподілу $F_{k-1}(t)$:

$$\begin{aligned}
f_k(t, F_{k-1}(t)) &= m\{\delta_k^0\} + cm\{\delta_k^I\} = \int_{-\infty}^{\infty} \left[\int_t^{\infty} (\tau-t) dF_{k-1}(\tau) + c \int_{-\infty}^t (t-\tau) dF_{k-1}(\tau) \right] dF_k^0(t) = \\
&= \int_{-\infty}^{\infty} \left[m\{t_{k-1}\} - t + c \int_{-\infty}^t (t-\tau) dF_{k-1}(\tau) \right] dF_k^0(t),
\end{aligned}$$

Функція $F_{k-1}(t)$ в силу монотонності інтегрується для безперервного і дискретного розподілів, отже:

$$\int_{-\infty}^t (\tau-t) dF_{k-1}(\tau) = \int_{-\infty}^t F_{k-1}(\tau) d\tau.$$

В результаті отримуємо:

$$f_k(t, F_{k-1}(t)) = m\{t_{k-1}\} - t + c \int_{-\infty}^t \int_{-\infty}^t F_{k-1}(\tau) d\tau dF_k^0(t). \quad (4.8)$$

Ця функція втрат на кожному k -му кроці (4.8) дозволяє використовувати метод динамічного програмування. Стан системи на $k-1$ кроці характеризується функцією розподілу $F_{k-1}(t)$, тому рівняння Беллмана набуває вигляду:

$$S_{k-1}(F_{k-1}(t)) = \min[f_k(t, F_{k-1}(t)) + S_k(\varphi_k(t, F_{k-1}(t)))]$$

де: $F_k(t) = \varphi_k(t, F_{k-1}(t))$ – рівняння стану в рекурентній формі, заданій виразами (4.6) і (4.7).

Функція (4.8) є функцією з монотонним включенням змінних

$$f_k(t, F_{k-1}(t)) = f_k(t, \varphi_{k-1}(t_{k-1}, \varphi_{k-2}(t_{k-2}, \dots, \varphi_1(t_1)) \dots)). \quad (4.9)$$

Завдання мінімізації функціоналу (4.5) є задачею Майера [80], схема якої відома.

Очевидно, стан на $k-1$ кроці, який характеризується однією з можливих реалізацій функції $F_{k-1}(t)$, найбільш істотно залежить від найближчих кроків процесу. Таким чином можна вважати, що $F_{k-1}(t)$ визначається вибором управління t_i на N -х попередніх кроках

$$F_{k-1}(t) = F(t_{k-1}, t_{k-2}, \dots, t_{k-N}).$$

Якщо управління t_i апроксимується m дискретними значеннями управління, при цьому число різних комбінацій управління на N кроках:

$$M = m^N,$$

де стан $F_{k-1}(t)$ може приймати M відповідних дискретних значень.

Конкретні значення N і M вибираються після кореляційного аналізу керованого процесу з урахуванням необхідної точності.

Функції $F_k^0(t)$ і $\Phi_k(t)$ визначаються за статистичними даними, корекція функцій дозволяє адаптувати систему з урахуванням досвіду попереднього планування. Реалізації даного алгоритму має бути подана (4.6), (4.7) і (4.8) в дискретній формі, замінити знаки інтегралів відповідними сумами.

4.3.2. Приклад рішення задачі динамічного програмування

Поняття «стану» є одним із найважливіших у динамічному програмуванні. «Стан» містить у собі передісторію процесу і описується з тим ступенем ймовірності, який дозволяє провести обчислення поточних альтернативних рішень із управління фрагментами на кожному з k -кроків. Даний процес укладається в шаблон мережевої моделі.

Для мережевої моделі вузли є «станом». Дуги виходять з будь-якого вузла, вказують напрямки можливих переходів, що визначають їх відповідне рішення. Процедуру прийняття рішення щодо вибору того чи іншого фрагмента називають стратегією. Оптимальна стратегія – це стратегія, яка оптимальна для кожного стану, на кожному кроці k .

Розглянуту в п. 4.3.1 задачу оптимізації потоків фрагментів можна інтерпретувати як задачу про найкоротший шлях на графі, де довжина дуги – тривалість скачування даного фрагмента. Для отримання практичних результатів спростимо вираз (4.8):

$$f_n(s) = \min[c_{sj} + f_{n-1}(j)], n = \overline{1, k}, \quad (4.10)$$

де k – число етапів на шляху початкової вершини в кінцеву;

c_{sj} – вартість по дузі (S, j) ;

$f_n(s)$ – функція підлягає мінімізації, що відповідає стратегії мінімальних витрат для шляху з пункту S , якщо до кінцевого пункту залишилося n етапів;

$j_n(s)$ – рішення, що дозволяє досягти $f_n(s)$;

f – значення цільової функції;

S – значення залежить від стану системи, індекс n вказує, скільки етапів залишається до кінця шляху.

Рекурентна формула (4.10) справджується тільки для завдання знаходження шляху мінімальної вартості. Для інших завдань рекурентні співвідношення матимуть інший вигляд.

Для визначеності задамося чисельними даними, що відображають типову ситуацію. Порядок розрахунку представлений в Додатку 1. На основі вихідних даних сформуємо граф виконання робіт на основі матриці r_{ijk} та t_{ik} . У матриці r_{ijk} для кожної роботи вказано номер машини. Матриця r_{ijk} представлена у вигляді таблиці 4.1.

Для вирішення завдання методом динамічного програмування сформуємо мережевий граф виконання робіт (програм). В якості вихідних даних використовуємо матрицю розподілу програм за пірами $\{r_{ijk}\}$ і матрицю тривалості зчитування фрагментів програм кожним піром $\{t_{ik}\}$. Матриця $\{r_{ijk}\}$ представлена у вигляді таблиці 4.1. Матриця тривалостей $\{t_{ik}\}$ представлена в таблиці 4.2. Граф задачі представлений на рисунку 4.2. Вершини мережевого графа відповідають виконанню окремих фрагментів програм відповідними пірами і мають подвійну нумерацію $\{i,j\}$, де i -номер фрагмента, j -номер піра. Дуги вказують послідовність зчитування фрагментів, а їх довжина визначає час зчитування.

Відомо, що кожна робота (програма) вимагає в точності одного виконання на кожній з машин (пірів), тобто $q_i = m, i = \overline{1, n}$, тобто всі роботи мають однакове число операцій - m .

Таблиця 4.1

Матриця розподілу програм за пірами

i\j	1			2			3		
k	1	2	3	1	2	3	1	2	3
1	1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	0	1	0
3	0	1	0	0	0	1	1	0	0
4	1	0	0	0	1	0	0	0	1

Таблиця 4.2

Матриця тривалостей

i\k	1	2	3
1	2	3	1
2	3	3	3
3	2	1	2
4	3	2	2

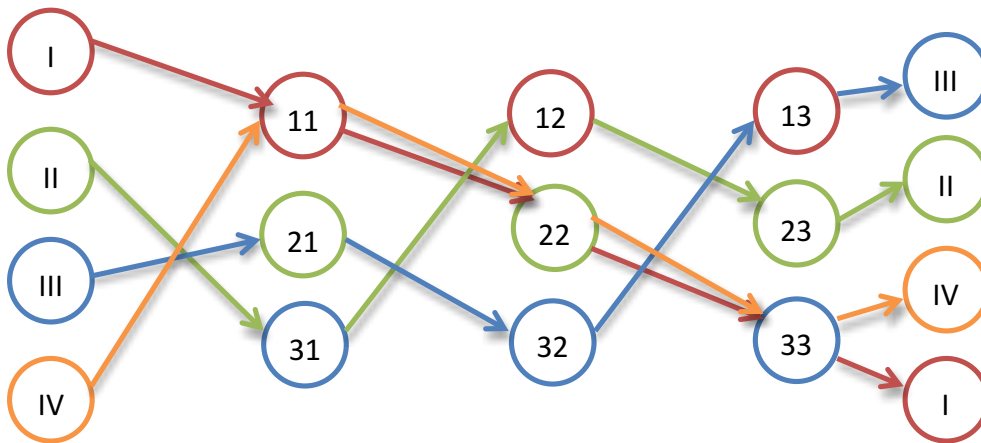


Рис. 4.2. Граф завантаження файлів

Алгоритм знаходження найкоротшого шляху методом динамічного програмування складається з двох етапів. Перший етап містить заключну вершину, (або всі вершини, які не мають вихідних дуг) і пов'язані з нею вершини. Кожен етап складається з двох ярусів. Перший ярус – вершини попереднього етапу, другий ярус, пов'язані з ним вершини. Другий ярус включає тільки ті вершини, всі дуги яких ведуть у вершини першого ярусу. Для зручності виконання завдання перенумеруємо вершини в порядку проходження.

Етапи виконання робіт наведені на рисунку 4.3

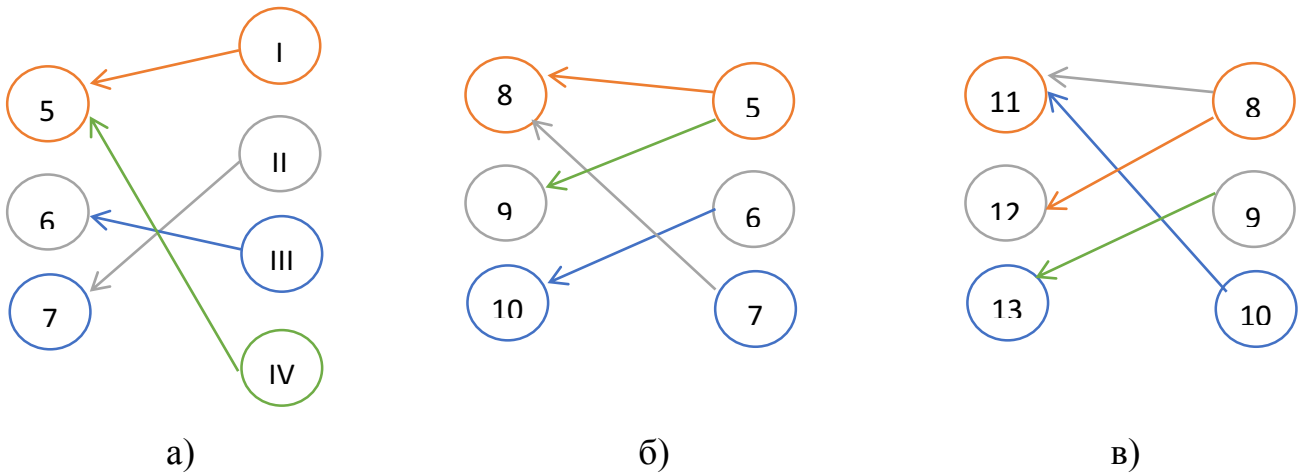


Рис. 4.3. Етапи виконання робіт а) III - етап б) II - етап в) I – етап

Для кожного етапу заповнюється таблиця витрат (таблиця 4.3), яка містить витрати на переміщення з вершини першого ярусу в другий, а так само мінімум витрат для кожної вершини і оптимальну стратегію.

Таблиця 4.3

Витрат і оптимальних стратегій а) для III етап б) II етап в) I етап

$S \setminus j$	12	13	14	15	$f_1(S)$	$j(S)$
8	∞	0+3	∞	∞	3	13
9	∞	∞	1+0	∞	1	14
10	∞	∞	∞	2+0	2	15
11	1+0	∞	∞	∞	1	12

а)

$S \setminus j$	8	9	10	11	$f_2(S)$	$j(S)$
5	∞	3+1	2+2	∞	4	10(9)
6	∞	∞	∞	2+1	3	11
7	3+3	∞	∞	∞	6	8

б)

$S \setminus j$	5	6	7	$f_3(S)$	$j(S)$
1	2+4	∞	∞	6	5
2	∞	∞	3+6	9	7
3	∞	1+3	∞	4	6
4	3+4	∞	∞	7	5

в)

Використовуючи оптимальне рішення на кожному етапі, знаходимо найкоротші шляхи, як послідовність вершин графа. Отримані послідовності вершин являють собою порядок скачування фрагментів програм на кожній із машин. Отриманий розклад представлено за допомогою діаграми Гантта (рис.4.4).

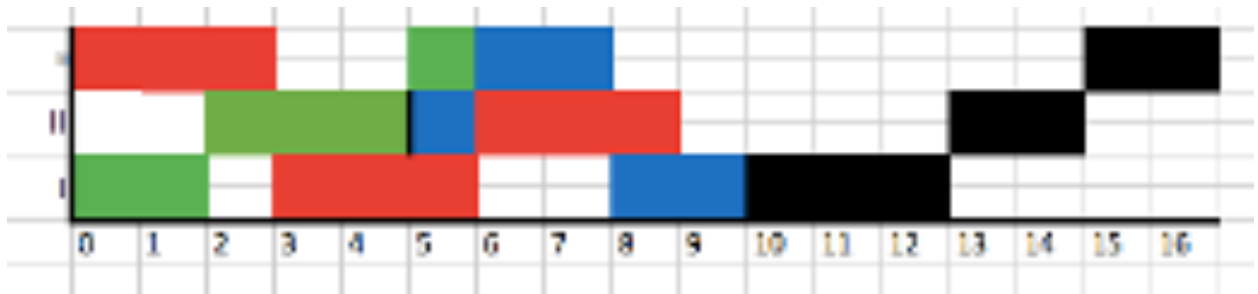


Рис. 4.4. Оптимальна послідовність скачуваних фрагментів

Як показав аналіз результатів рішення задачі двома методами, представленими в 4,5 розділах, загальний час скачування чотирьох програм з трьох машин відрізняється незначно (два тимчасові інтервали). Слід зауважити, що рішення задачі методом лінійного цілочисельного програмування відрізняється складністю підготовчого етапу і при більшій кількості змінних рішення даним методом є трудомістким. При використанні методу динамічного програмування складність представляє безпосередньо програмна реалізація. Тому вибір методу є неоднозначним і визначається для кожного конкретного завдання.

4.4. Висновки до четвертого розділу

1. Процес скачування файлу в загальному вигляді представляється дискретно-неперервною динамічною моделлю, що являє собою клас вироджених випадкових процесів із кінцевою послідовністю точок розриву. Модель такого дискретно-неперервного процесу протягом скачування файлу в спрощеному вигляді представляється керованим різницеvim рівнянням стану, що допускає

представлення у вигляді дворівневої структури, що характеризується послідовністю дискретних значень і послідовністю ділянок неперервного процесу.

2. З використанням теорії розкладів і на основі дискретно-неперервної моделі процесу скачування фрагментів показана можливість упорядкування фрагментів у буфері й мінімізації затримки кожного з них, що забезпечить безшовне відтворення, що представляє собою вирішення задачі за критерієм мінімізації максимального штрафу. Задача в загальному вигляді зводиться до NP – повного важкого рішення. Одержано приватне рішення задачі з використанням кусочно-неперервної моделі, вирішення якої може бути одержано за допомогою узагальненого лагранжиану, динамічного програмування або цілочисельного лінійного програмування.

3. З використанням математичної моделі потоку фрагментів при скачуванні відповідного файла знайдено оптимальний план, який враховує для кожного фрагмента випадковий час знаходження на обслуговуванні, а також випадковий час самого обслуговування кожного фрагмента. Оптимізація плану відбувається згідно з критерієм, що враховує втрати, пов'язані з надходженням i -того фрагменту раніше, ніж скінчилося зчитування j -того фрагменту, і втрати при затримці i -того фрагмента, якщо зчитувач простоює. В результаті оптимізації знайдемо функцію з монотонним включенням змінних, що є стандартною задачею Майєра і зводиться до рівняння Беллмана і відповідної задачі динамічного програмування.

4. При скачуванні файла має місце потік керованих фрагментів, що надходять від різних пірів, що характеризуються випадковим часом надходження фрагмента на обслуговування і випадковим часом обслуговування кожного з фрагментів. У роботі отримано функцію розподілу часу обслуговування фрагмента і функцію розподілу надходження фрагмента на обслуговування.

5. Представлено приклад практичної задачі оптимізації послідовності скачуваних фрагментів із використанням процедури динамічного програмування. Оптимізація полягає в упорядкуванні послідовності фрагментів і мінімізації затрат часу при їх скачуванні.

РОЗДІЛ 5

ОПТИМІЗАЦІЯ ПРОЦЕСУ СКАЧУВАННЯ ФРАГМЕНТІВ НА ОСНОВІ ДИСКРЕТНО-НЕПЕРЕРВНОЇ МОДЕЛІ

5.1. Оптимізація сумарного зваженого часу обслуговування в пінгвовій мережі

Додатки Peer-To-Peer (P2P) стали досить популярні серед інтернет-користувачів. P2P-технології покращують масштабованість системи при низьких витратах реалізації. На відміну від традиційних клієнт-серверних архітектур, піри в мережі виступають в якості клієнта (ліч-викачує) і сервера (сід-роздає). Піри не тільки скачують файл із мережі, а й роздають завантажений файл іншим користувачам у мережі. Частина файлів обмінюються через прямі з'єднання між пірами.

Іншою важливою властивістю пінгвових мереж є можливість впорядкованості фрагментів, переривань, входжень і виходів з мережі в будь-який момент часу.

Практичне застосування пінгвових мереж постійно розширюється, а призначений для користувача трафік за даними Cisco Systems в 2020 році досягне 60 петабайт/місяць. Велику практичну користь дає можливість організації розподілених обчислень, що дозволяє цим мережам конкурувати з продуктивністю сучасних суперкомп'ютерів. Значна частка завдань відводиться на організацію он-лайн ігор (MMOG). Однак, за тими ж даними, близько половини трафіку пінгвових мереж припадає на організацію файлообміну P2P-TV, що привертає до цієї технології особливу увагу.

Архітектура пінгвових мереж може бути структурованою або неструктурованою. У структурованих системах P2P-HQTV структура оверлею зазвичай являє напівдетерміновану топологію, що викликає складності в реструктуризації при приєднанні або від'єднанні чергового піру. Відмінною

особливістю неструктурованою архітектури P2P-TV систем є те, що топологія оверлею повністю відокремлена від топології розподілу. В результаті утворюється mesh-топологія, яка дозволяє реалізувати високий ступінь динамічності й децентралізації.

Пірингові мережі, будучи приватними, оверлейними файлообмінними мережами, мають ряд специфічних особливостей по відношенню до загальної IP-мережі. Так у відомих файлообмінних додатках (Bit Torrent) оптимізація продуктивності мережі зводиться до пошуку «кращого» піру, з якого завантажується вміст. Зазначимо інші важливі для підвищення продуктивності мережі завдання, які необхідно вирішувати паралельно в реальному часі: алгоритми планування елементів, вибір піру з низькими затримками, забезпечення взаємодії між динамікою оверлею і IP-маршрутизацією.

Таким чином, загальносистемні критерії якості: QoS і QoE, хоча і мають безпосередню залежність від якості рішення перерахованих і багатьох інших завдань, не завжди дозволяють отримати рішення в цілому. Отже, багато робіт, присвячених дослідженням пірингових мереж, носять приватний характер. Завдання цього дослідження полягає в спробі отримання загальносистемного рішення, оптимального за критерієм мінімуму затримки при завершенні обслуговування за наявності ряду типових обмежень.

Постановка задачі:

Будемо вважати, що P2P-TV мережа забезпечує одночасний перегляд n_m – програм $m \in M$. Одну з цих програм переглядає l - абонентів (пірів) $l \in L$. У будь-який момент кожен з l -пірів має право припинити обслуговування, а потім – знову почати. У кожного піра є $N = \{1, 2, \dots, n\}$ вимог, які представлені у виді окремих j -фрагментів скачуваного контенту.

Тривалість обслуговування мережею кожного з перерахованих j -фрагментів становить p_j , де $j \in N$. Процес скачування контенту є послідовністю процедур із забезпеченням часткової впорядкованості, що в кінцевому рахунку визначається обмеженнями передування. У разі якщо в обмеженнях є

директивний термін завершення обслуговування по j -фрагменту d_j , то різниця між моментом завершення C_j і директивним моментом d_j при $C_j > d_j$ вважається запізненням в обслуговуванні або обслуговуванням з випередженням при $C_j < d_j$.

Будемо вважати, що кожен користувач має буфер, який здатний вмістити всі безліч вимог $N = \{1, 2, \dots, n\}$, де n – кількість місць для зберігання n -фрагментів, отриманих на попередніх етапах, плюс одне місце, призначене для отримання чергової нової порції даних. Відповідно до правила завантаження, після того, як сервер випадковим чином обрав піра для початкового завантаження, він знаходить в мережі іншого піра і намагається завантажити від нього відсутню порцію даних. Таким чином, в результаті відмінностей в пропускних можливостях каналів і тривалостях завантаження та прийому буфера мають місце випадки порушення часткового порядку.

Для врахування обмежень передування в обслуговуванні задамо бінарне відношення строгого порядку на безлічі вимог N . Вимога i є попередником вимоги j , а вимога j – послідовником, нащадком по відношенню до i . Дане відношення задається орієнтованим бесконтурним графом G , безліч вершин (вузлів) якого ідентифікується з безліччю вимог, а шлях з вершини i в вершину j існує, коли $i \rightarrow j$.

Формулювання завдання мінімізації сумарного зваженого часу обслуговування

Будемо вважати, що при реалізації N -го вимоги на термінал надходить не більше одного фрагмента одночасно. Для кожного j -го фрагмента, що належить безлічі $N = \{1, 2, \dots, n\}$ задані параметри:

– тривалість обслуговування j -фрагмента – p_j , $0 < p_j^L \leq p_j^U$; де індекси L і U відносяться відповідно до нижньої і верхньої меж обслуговування. Зазвичай тривалість обслуговування p_j виявляється невідомою аж до моменту завершення

даного j -обслуговування. Випадковий характер p_j визначається різницею в параметрах ліній скачування і використовуваних технологій;

– кожному j - вимозі зіставляється ваговий множник $w_j > 0$. Значення вагових коефіцієнтів w_j не можуть бути однаковими, в першу чергу через те, що важливість наступних фрагментів до кінця скачування зростає, тому

$$w_n \geq w_{n-1} \geq \dots \geq w_1.$$

Для знаходження рішень задачі скористаємося результатами теорії розкладів (Scheduling Theory), яка зокрема зводиться до побудови функції, визначеної на множині перестановок і до розробки оптимізаційних алгоритмів. На кожному з n - етапів обслуговування можливі різні тривалості реалізації p_j кожного з j -обслуговувань, $j \in N$. На практиці значення p_j утворюють кінченну безліч варіантів з інтервалу $[p_j^L; p_j^U]$. Таким чином можна розглядати k – мірний вектор

$$p_j^T = (p_1, p_2, \dots, p_k), p_j \in [p_j^L; p_j^U]. \quad (5.1)$$

Значення p_j в загальному випадку стають відомі після прийому кожного з фрагментів: $p_j = p_{ps}^{(j)}$.

Нехай P – позначає безліч всіх векторів (5.1) можливих тривалостей обслуговування вимог

$$p_i^T = (p_1^{(l)}, p_2^{(k)}, \dots, p_n^{(k)}), k = 1, 2, \dots, k. \quad (5.2)$$

Вектор (5.2) являє собою сценарій подій, послідовності n -фрагментів по заявці даного піру.

Очевидно P – замкнутий прямокутник невід'ємних чисел.

При плануванні розкладу s слідування фрагментів викачуваного контенту і реалізації сценарію (5.2) в якості критеріальною можна вибрати адитивну зважену лінійну функцію:

1. Сумарно зважений час, що витрачається для прийому вектора (5.2)

$$f_i(C_i) = \sum_{i=1}^n w_i C_i, \quad (5.3)$$

де w_i – ваговий множник, що визначає значимість i -го фрагмента, в тому числі часу його обслуговування.

Для різних w_i значення (5.3) функцій $f_i(C_i)$ представляється у вигляді прямих на графіках рис.5.1.

2. Сумарна зважена кількість фрагментів, прийнятих із запізненням $f_i(C_i) = \sum_{i=1}^n w_i U_i$,

де U_i дорівнює нулю, якщо $C_i \leq d_i$ – директивний термін (у відсутності запізнь) і дорівнює одиниці при $d_i \leq C_i$.

Можливі значення функцій $f_i(C_i)$ представлені на рис.5.1.

3. Сумарне зважене запізнення фрагментів

$$f_i(C_i) = \sum_{i=1}^n w_i T_i, \quad (5.4)$$

де $T_i = \max(0, C_i - d_i)$.

Значення функції представлені на рис.5.1.

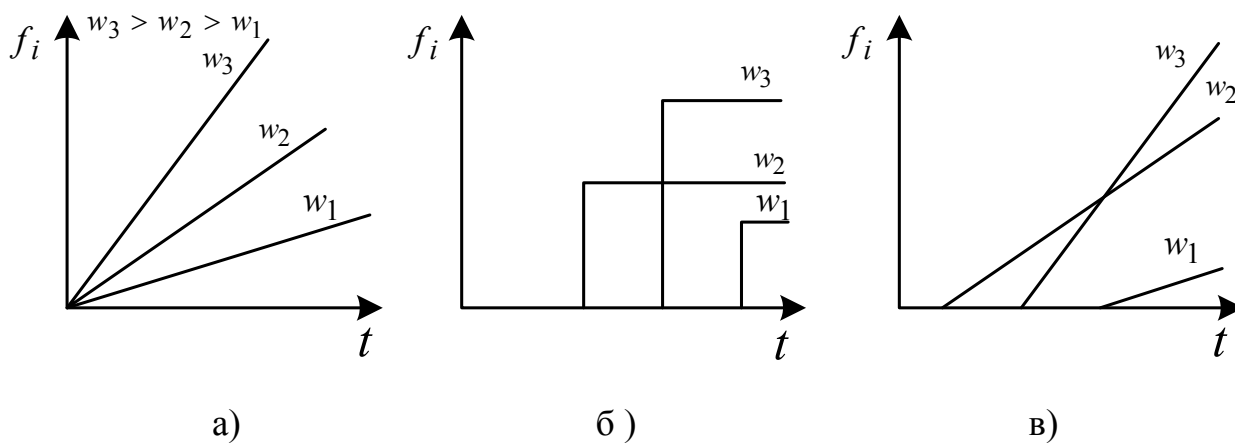


Рис. 5.1. Графік якісної поведінки критеріальних функцій

а-(3); б-(4); в-(5).

Слід мати на увазі, що сценарій (5.3) у загальному випадку залишається невизначеним аж до моменту завершення обслуговування за всіма i і j . При цьому через NP – повноти мінімізація критеріальних функцій неможлива

принаймні на стадії обчислення і складання його оптимального розкладу. Разом з тим, для основних технологій P2P-TV, при скачуванні конкретних файлів, диференціальна фрагментація як правило не використовується. Це дозволяє вибирати для всіх p_i

$$p_i = p, \forall i. \quad (5.5)$$

З умовою (5.3) можна вважати параметри вектора відомим апріорі. При цьому вдається уникнути NP -повноти і вирішити оптимізаційні задачі за час ($n \log n$):

$$f_{opt}(t) = \min_s \sum_{i=1}^n w_i C_i(p_i = p).$$

В теорії розкладів є безліч розкладів подібних завдань за умови додаткових обмежень. Так, відомі рішення, засновані на методі динамічного програмування. Серед інших практично важливих слід вказати на методи дихотомії, робастні методи. Перспективним є метод, заснований на аналізі чутливості, що пропонує двохстадійне прийняття рішень: стадію апріорного планування k -варіантів у режимі off-line і стадію складання розкладів в режимі on-line. В силу відмінностей умов скачування контенту, інших випадкових факторів у послідовності прийнятих піром фрагментів можуть бути порушені умови передування, що потребують упорядкування і відповідних перестановок у порядку проходження фрагментів. Це накладає додаткові обмеження на рішення оптимізаційної задачі. Так, обраний нами раніше критерій (5.3) з урахуванням необхідної процедури перестановки $\pi = (\pi_1, \pi_2, \dots, \pi_n!)$ $\pi = (\pi_1, \pi_2, \dots, \pi_n!)$, отримує вигляд:

$$J = \sum_{j=1} w_j C_j(\pi, P) = \min_{\pi_k} \{ \sum_{\pi} \sum_j w_j C_j(\pi_k p) \} \quad (5.6)$$

5.2. Синтез процедури мінімізації сумарного зваженого часу обслуговування в піринговій мережі

Дане завдання спрямоване на оптимізацію зчитування і планування фрагментів, що становлять конкретний файл. Тому рішення даної задачі є більш

приватним по відношенню до задачі динамічного програмування. Передбачається, що це рішення необхідно знаходити для кожного файлу незалежно.

5.2.1. Математична постановка і рішення задачі цілочисельного лінійного програмування

Черговим важливим обмеженням, яке зустрічається в процесі функціонування пірингових мереж, є наявність переривань, викликаних технологічними причинами при обслуговуванні в умовах великого навантаження або за рахунок впливів більш пріоритетних обслуговувань.

Розглянемо приклад оптимізації середнього зваженого часу початку обслуговування. Розглянемо як приклад ситуацію, коли в мережі P2P знаходиться m - машин (пірів) і один користувач запитує n програм для зчитування з даних пірів. Тривалість обслуговування мережею кожного фрагмента відома. Пір володіє буфером, здатним містити всі безліч вимог. Процес скачування контенту є послідовністю процедур.

Необхідно забезпечити порядок скачування файлів із мінімізацією середньо зваженого часу початку обслуговування. Порядок вирішення задачі представлений в Додатку 1.

В якості одновимірного випадку розглянемо задачу зчитування чотирьох файлів одним піром при наступних вихідних даних, обраних довільним чином: вектор значень тривалостей обслуговування фрагментів $p = \{6, 2, 2, 3\}$; вектор значень штрафів вимог $W = \{4, 2, 2, 3\}$.

Виберемо число M на два порядки більше величин тривалостей для того, щоб завжди виконувалися нерівності (4-5), рівним 100.

Тоді задача цілочисельного лінійного програмування матиме такий вигляд:

Мінімізувати цільову функцію:

$$F(X) = 4x_1 + 2x_2 + 2x_3 + 3x_4 \rightarrow \min$$

при наступних обмеженнях:

$$-x_1 + x_2 - 102x_5 \leq -2$$

$$-x_1 + x_3 - 102x_6 \leq -2$$

$$-x_1+x_4 -102x_7 \leq -3$$

$$-x_2+x_3 -102x_8 \leq -2$$

$$-x_2+x_4 -102x_9 \leq -3$$

$$-x_3+x_4 -103x_{10} \leq -3$$

$$x_1-x_2 +106x_5 \leq 100$$

$$x_1 -x_3 +106 x_6 \leq 100$$

$$x_1- x_4 +106 x_7 \leq 100$$

$$x_2 - x_3 +102 x_8 \leq 100$$

$$x_2 - x_4 +102 x_9 \leq 100$$

$x_3 - x_4 +102 x_{10} \leq 100, x_{1,4} \in \mathbb{N}, x_{5,10} \in \{0,1\}$. Тут задана наступна відповідність: $x_5 = y_{12}, x_6 = y_{13}, x_7 = y_{14}, x_8 = y_{23}, x_9 = y_{24}, x_{10} = y_{34}$.

З використанням пакета MatLAB отримано оптимальне рішення, представлене на рис.5.2 діаграмою Гантта у вигляді послідовності фрагментів, об'єднаних у файл. З діаграми випливає, що сформований файл являє собою щільну множину, де перестановки визначені наявністю явищ передування. Текст програми приведений в додатку 2.

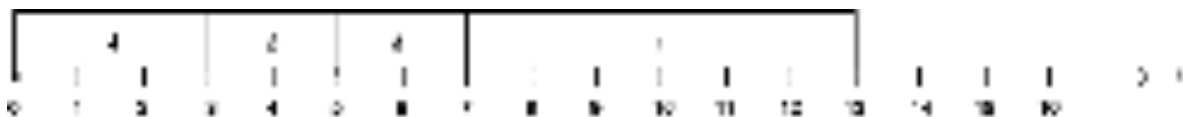


Рис. 5.2. Оптимальна послідовність зчитування фрагментів

В якості багатовимірного випадку розглянемо задачу при $m = 3$ і $n = 4$. Матриця $\{r_{ijk}\}$ розподілу програм за пірами представлена у вигляді таблиці 5.1. В якості тривалості виконання робіт виберемо випадкові числа від 1 до 10. Матриця тривалостей $\{t_{ik}\}$ представлена в таблиці 5.2.

Таблиця 5.1

Матриця розподілу програм за пірами

i\j	1			2			3		
	1	2	3	1	2	3	1	2	3
k	1	2	3	1	2	3	1	2	3
1	1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	0	1	0
3	0	1	0	0	0	1	1	0	0
4	1	0	0	0	1	0	0	0	1

Таблиця 5.2

Матриця тривалостей

i\k	1	2	3
1	2	3	1
2	3	3	3
3	2	1	2
4	3	2	2

Введемо наступні позначення для шуканих змінних $t_{ik} = (x_1 : x_{12})$. Булеві змінні Y_{ijk} та відповідні їм імена змінних вектора змінних X представлені в таблиці 5.3.

Таблиця 5.3

Таблиця відповідності номерів змінних

	1 операція	2 операція	3 операція
$Y_{ijk} :$	$Y_{121} = x_{13};$ $Y_{122} = x_{14};$ $Y_{123} = x_{15};$ $Y_{131} = x_{16};$ $Y_{132} = x_{17};$ $Y_{133} = x_{18};$ $Y_{141} = x_{19};$ $Y_{142} = x_{20};$ $Y_{143} = x_{21};$	$Y_{231} = x_{22};$ $Y_{232} = x_{23};$ $Y_{233} = x_{24};$ $Y_{241} = x_{25};$ $Y_{242} = x_{26};$ $Y_{243} = x_{27};$	$Y_{341} = x_{28};$ $Y_{342} = x_{29};$ $Y_{343} = x_{30};$

Вектор значень штрафів цільової функції (4.11) $W = \{2,1,3,4,6,2,5,3,2,2,4,1\}$, заданий випадковим чином, булеві змінні входять у цільову функцію з нульовими коефіцієнтами.

Тут досить велика константа M дорівнює сумі всіх тривалостей зчитування фрагментів. Тоді обмеження (5.14-5.16) приймуть наступний вигляд:

$$\begin{array}{lll}
 x_1 - x_4 + 30 \cdot x_{13} \geq 3; & x_1 - x_{10} + 30 \cdot x_{19} \geq 3; & x_4 - x_{10} + 30 \cdot x_{25} \geq 3; \\
 x_2 - x_5 + 30 \cdot x_{14} \geq 3; & x_2 - x_{11} + 29 \cdot x_{20} \geq 2; & x_5 - x_{11} + 29 \cdot x_{26} \geq 2; \\
 x_3 - x_6 + 30 \cdot x_{15} \geq 3; & x_3 - x_{12} + 29 \cdot x_{21} \geq 2; & x_6 - x_{12} + 29 \cdot x_{27} \geq 2; \\
 x_1 - x_7 + 29 \cdot x_{16} \geq 2; & x_5 - x_8 + 28 \cdot x_{23} \geq 1; & x_7 - x_{10} + 30 \cdot x_{28} \geq 3; \\
 x_2 - x_8 + 28 \cdot x_{17} \geq 1; & x_4 - x_7 + 29 \cdot x_{22} \geq 2; & x_8 - x_{11} + 29 \cdot x_{29} \geq 2; \\
 x_3 - x_9 + 29 \cdot x_{18} \geq 2; & x_6 - x_9 + 29 \cdot x_{24} \geq 2; & x_9 - x_{12} + 29 \cdot x_{30} \geq 2;
 \end{array}$$

$$\begin{array}{lll}
 -x_1 + x_4 - 30 \cdot x_{13} \geq -3; & -x_1 + x_4 - 29 \cdot x_{13} \geq -27; & x_1 - x_4 + 29 \cdot x_{13} \geq 27; \\
 -x_2 + x_5 - 30 \cdot x_{14} \geq -3; & -x_2 + x_5 - 30 \cdot x_{14} \geq -27; & x_2 - x_5 + 30 \cdot x_{14} \geq 27; \\
 -x_3 + x_6 - 30 \cdot x_{15} \geq -3; & -x_3 + x_6 - 28 \cdot x_{15} \geq -27; & x_3 - x_6 + 28 \cdot x_{15} \geq 27; \\
 -x_1 + x_7 - 29 \cdot x_{16} \geq -2; & -x_1 + x_7 - 29 \cdot x_{16} \geq -27; & x_1 - x_7 + 29 \cdot x_{16} \geq 27; \\
 -x_2 + x_8 - 28 \cdot x_{17} \geq -1; & -x_2 + x_8 - 30 \cdot x_{17} \geq -27; & x_2 - x_8 + 30 \cdot x_{17} \geq 27; \\
 -x_3 + x_9 - 29 \cdot x_{18} \geq -2; & -x_3 + x_9 - 28 \cdot x_{18} \geq -27; & x_3 - x_9 + 28 \cdot x_{18} \geq 27; \\
 -x_1 + x_{10} - 30 \cdot x_{19} \geq -3; & -x_1 + x_{10} - 29 \cdot x_{19} \geq -27; & x_1 - x_{10} + 29 \cdot x_{19} \geq 27; \\
 -x_2 + x_{11} - 29 \cdot x_{20} \geq -2; & -x_2 + x_{11} - 30 \cdot x_{20} \geq -27; & x_2 - x_{11} + 30 \cdot x_{20} \geq 27; \\
 -x_3 + x_{12} - 29 \cdot x_{21} \geq -2; & -x_3 + x_{12} - 28 \cdot x_{21} \geq -27; & x_3 - x_{12} + 28 \cdot x_{21} \geq 27; \\
 -x_4 + x_7 - 29 \cdot x_{22} \geq -2; & -x_4 + x_7 - 30 \cdot x_{22} \geq -27; & x_4 - x_7 + 30 \cdot x_{22} \geq 27; \\
 -x_5 + x_8 - 28 \cdot x_{23} \geq -1; & -x_5 + x_8 - 30 \cdot x_{23} \geq -27; & x_5 - x_8 + 30 \cdot x_{23} \geq 27; \\
 -x_6 + x_9 - 29 \cdot x_{24} \geq -2; & -x_6 + x_9 - 30 \cdot x_{24} \geq -27; & x_6 - x_9 + 30 \cdot x_{24} \geq 27; \\
 -x_4 + x_{10} - 30 \cdot x_{24} \geq -2; & -x_4 + x_{10} - 30 \cdot x_{24} \geq -27; & x_4 - x_{10} + 30 \cdot x_{24} \geq 27; \\
 -x_4 + x_{10} - 30 \cdot x_{25} \geq -3; & -x_4 + x_{10} - 30 \cdot x_{25} \geq -27; & x_4 - x_{10} + 30 \cdot x_{25} \geq 27; \\
 -x_5 + x_{11} - 29 \cdot x_{26} \geq -2; & -x_5 + x_{11} - 30 \cdot x_{26} \geq -27; & x_5 - x_{11} + 30 \cdot x_{26} \geq 27; \\
 -x_6 + x_{12} - 29 \cdot x_{27} \geq -2; & -x_6 + x_{12} - 30 \cdot x_{27} \geq -27; & x_6 - x_{12} + 30 \cdot x_{27} \geq 27;
 \end{array}$$

$$\begin{aligned}
 -x_7 + x_{10} - 30 \cdot x_{28} &\geq -3; & -x_7 + x_{10} - 29 \cdot x_{28} &\geq -27; & x_7 - x_{10} + 29 \cdot x_{28} &\geq 27; \\
 -x_8 + x_{11} - 29 \cdot x_{29} &\geq -2; & -x_7 + x_{10} - 29 \cdot x_{28} &\geq -27; & x_7 - x_{10} + 29 \cdot x_{28} &\geq 27; \\
 -x_9 + x_{12} - 29 \cdot x_{30} &\geq -2; & -x_9 + x_{12} - 29 \cdot x_{30} &\geq -27; & x_9 - x_{12} + 29 \cdot x_{30} &\geq 27;
 \end{aligned}$$

З використанням пакета MatLAB отримано оптимальне рішення, представлене за допомогою діаграми Ганта на рис.5.3. Текст програми наведений у Додатку 2.

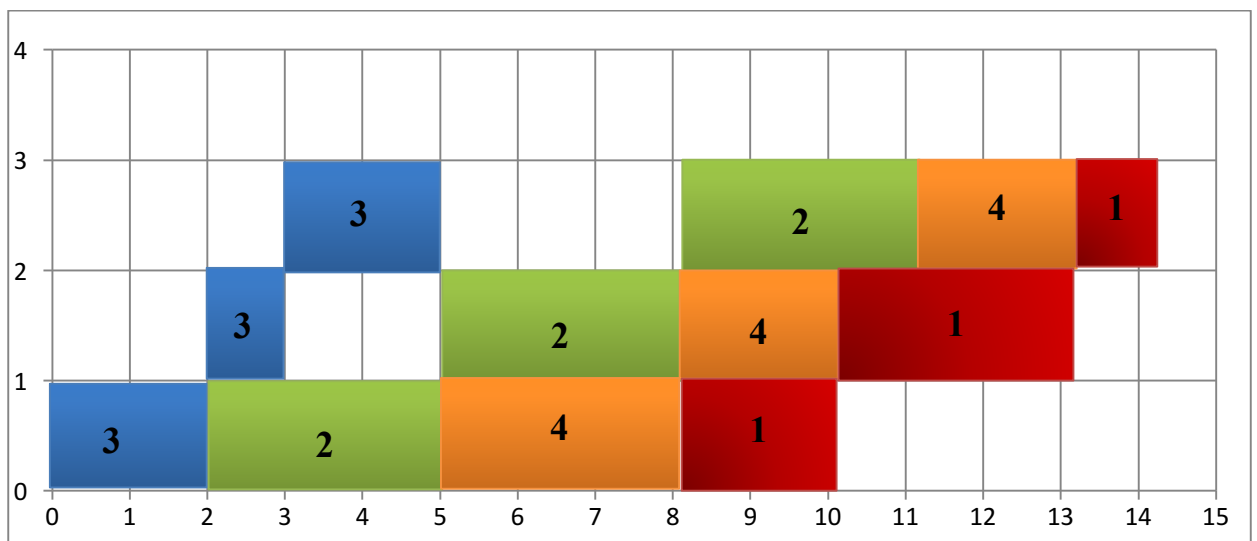


Рис. 5.3. Оптимальна послідовність зчитування фрагментів

5.3. Аналіз процедури мінімізації сумарного зваженого часу обслуговування

Для аналізу ефективності запропонованої моделі цілочисельного лінійного програмування (ЦЛП) була проведена серія експериментів ($N = 1000$) в пакеті Matlab. Оскільки тривалості зчитування фрагментів кожним із бенкетів є випадковими з невідомими законами розподілу, моделювання було проведено в припущенні нормального і рівномірного розподілів. Були отримані реалізації цільової функції і гістограми (рис.5.4-5.5). Цільова функція в завданні ЛЦП є середньозважене часу початку зчитування всіх файлів, ваговими коефіцієнтами якої є значення пріоритетів зчитування файлів. При моделюванні розглядалися

два варіанти, за однакової кількості пріоритетів і при розподілі значень пріоритетів за нормальним законом. На рис.5.4-5.5 наведені результати моделювання при нормальному розподілі тривалості обробки фрагментів ($m.o. = 4$; $C.K.O. = 1$) при рівних вагових коефіцієнтах цільової функції. На підставі результатів моделювання можна зробити висновок про нормальний розподіл значень цільової функції в серії експериментів за критерієм ХІ-квадрат при рівні значущості 0.2, що не суперечить теоретичним положенням про властивості нормально розподіленої випадкової величини. Вибіркове середнє $m = 4.5215$, вибіркова дисперсія $d = 0.1432$.

Метою оптимізації є мінімізація загального часу зчитування файлів (мінімізація затримки). В експерименті було визначено загальний час зчитування файлів на підставі значень змінних ЗЛЦП і відомих тривалостей обробки фрагментів файлів на кожному з пірів: максимальне значення часу початку зчитування (максимальне значення змінної) для кожного з бенкетів плюс тривалість обробки даного фрагмента. Отримані значення були оброблені за допомогою процедури рекурсивного оцінювання Роббінса-Монро, результати наведені на рис.5.6. Вибіркове середнє одно 14.7863 одиниць модельного часу, умовне середнє одно 14.9278 одиниць модельного часу, апостеріорна дисперсія помилки оцінки прагне до нуля (рис.5.7).

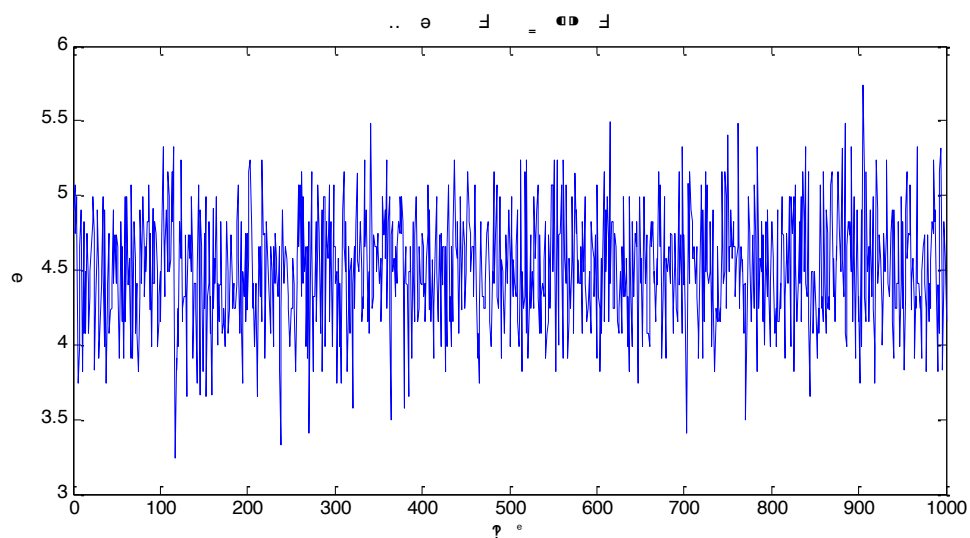


Рис. 5.4. Значення середньозваженого часу початку зчитування файлів

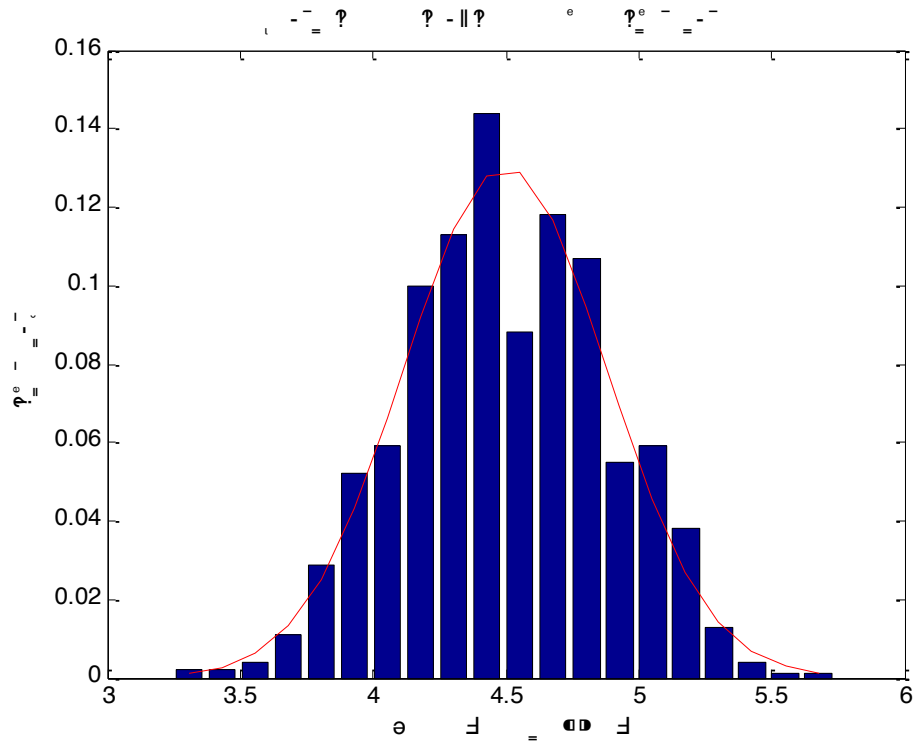


Рис.5.5. Гістограма цільової функції

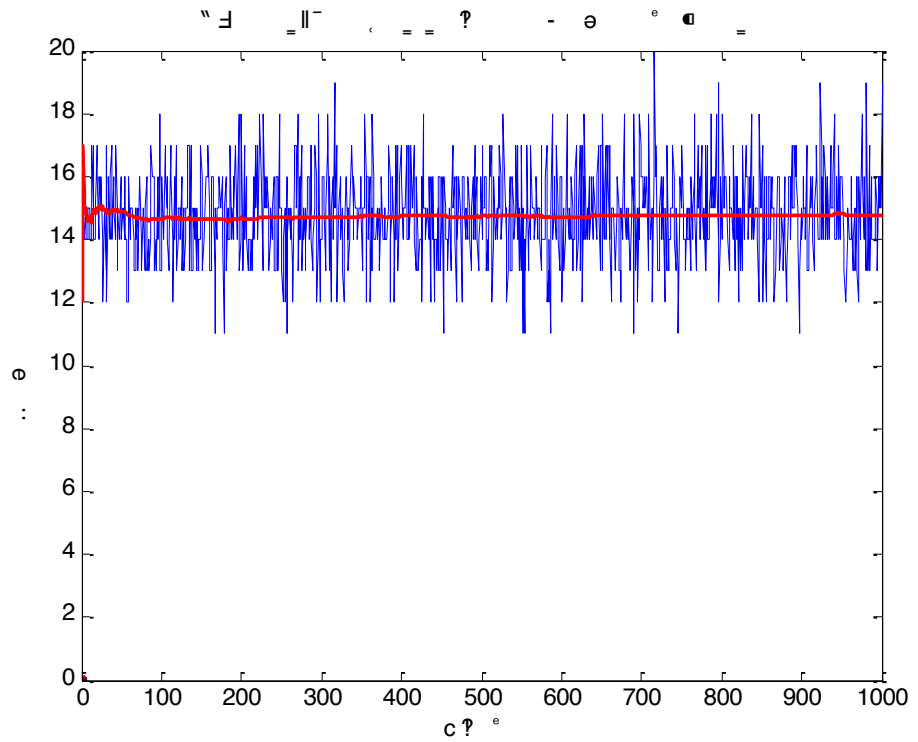


Рис.5.6. Оцінка оптимального часу зчитування всіх файлів

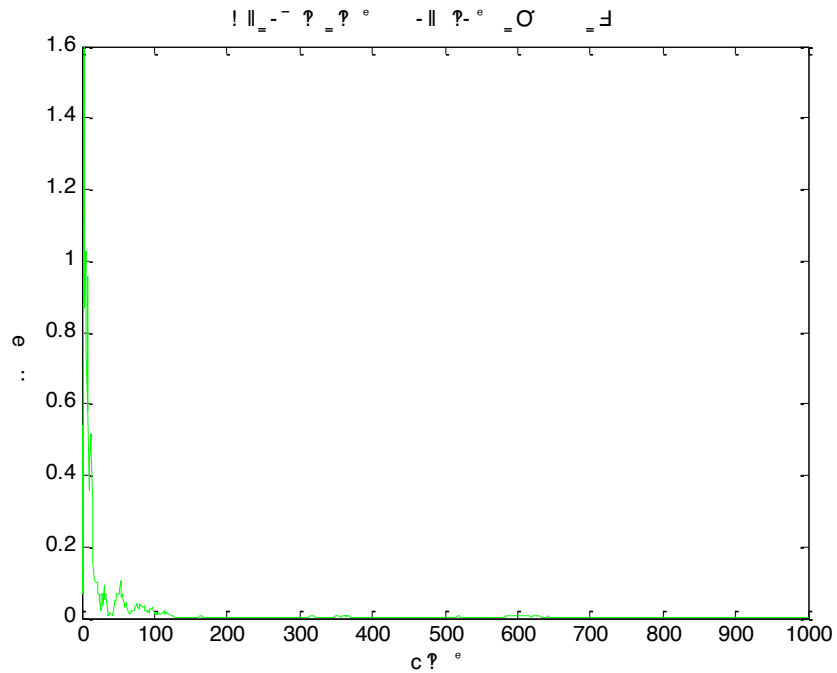


Рис.5.7. Апостеріорна дисперсія помилки оцінки

Далі було проведено моделювання при розподілі вагових коефіцієнтів за нормальним законом (м.о.=0.08; с.к.о.= 0.02). Гістограма наведена на рис. 5.8, вибіркоче середнє $m=4.5740$, вибіркова дисперсія = 0.1625, рівень значимості = 0.2. умовне середнє оптимального часу зчитування файлів наведено на рис. 5.9.

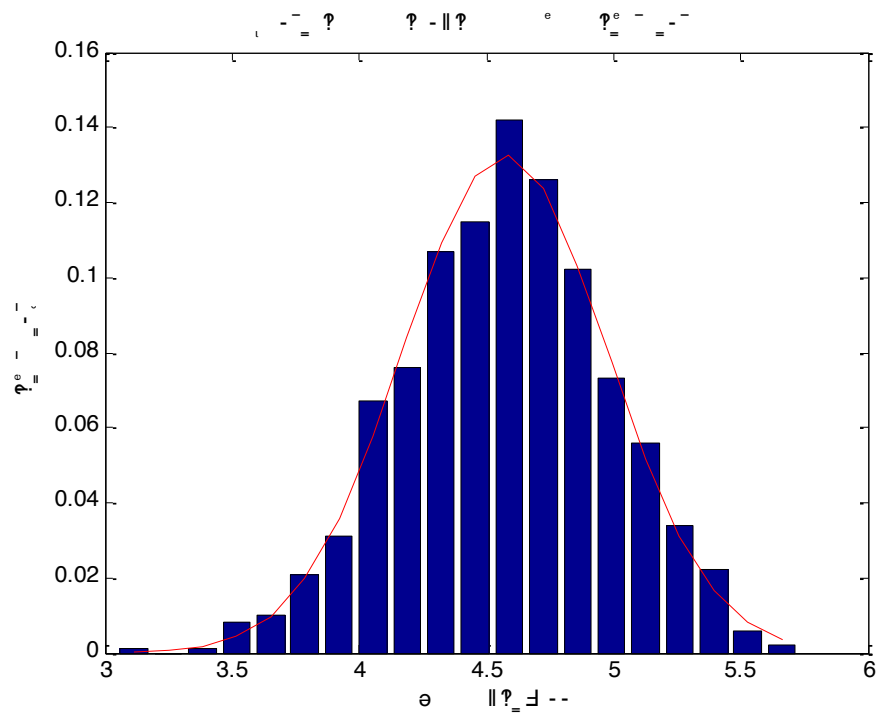


Рис.5.8. Гістограма цільової функції

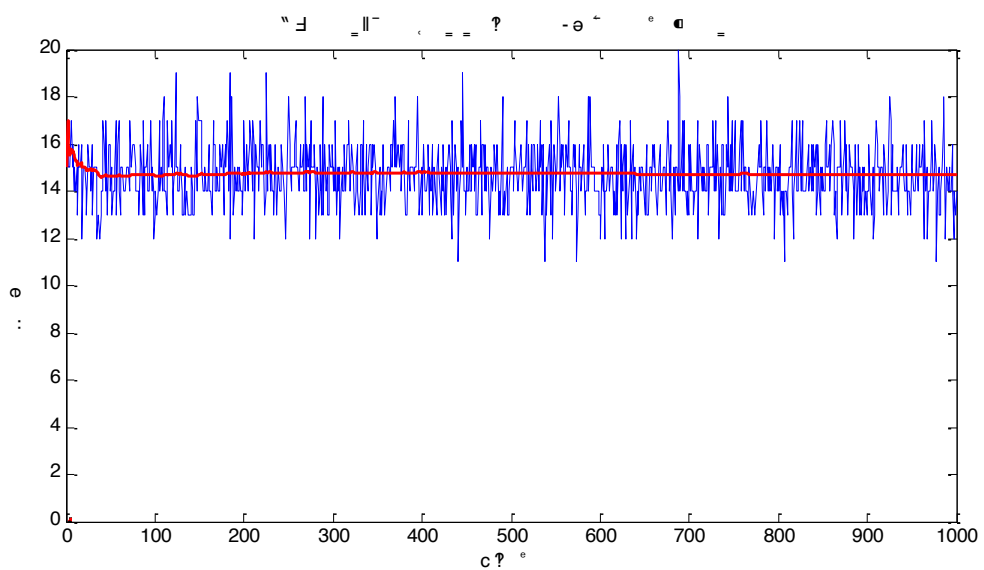


Рис. 5.9. Оцінка оптимального часу зчитування файлів

Результати моделювання при рівномірному розподілі тривалості обробки фрагментів ($a=2$; $b=6$) при рівних вагових коефіцієнтах цільової функції представлені на рис. 5.10-5.11. Значення цільової функції розподілені нормально з

рівнем значимості, рівним 0.2, вибіркоче середнє дорівнює $m = 4.4896$; вибіркова дисперсія $d = 0.1685$. Оцінка оптимального часу зчитування файлів із використанням процедури Роббінса-Монро наведена на рис. 5.12.

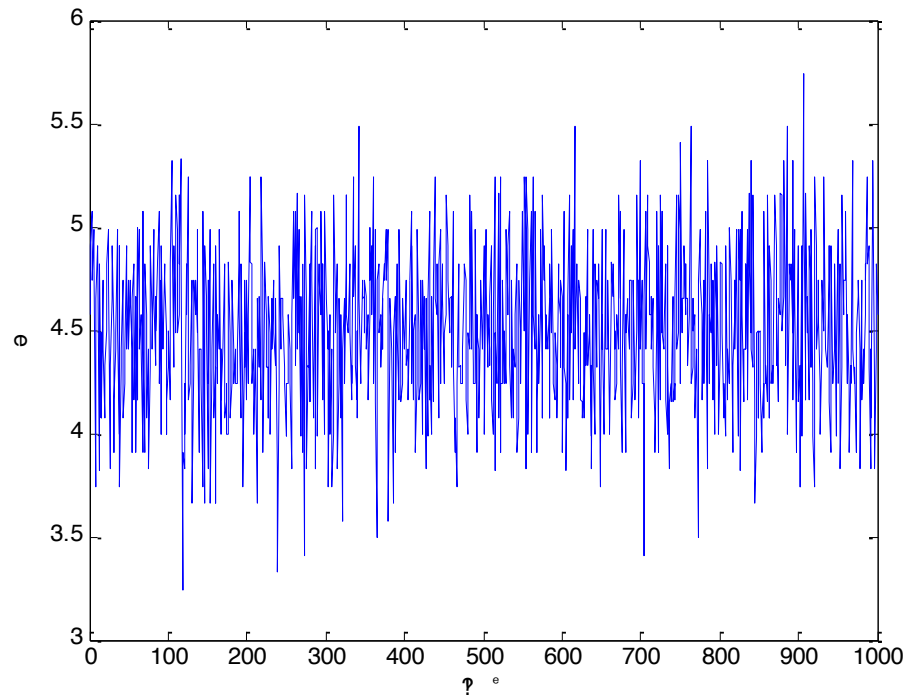


Рис. 5.10. Середньозважений час початку зчитування файлів

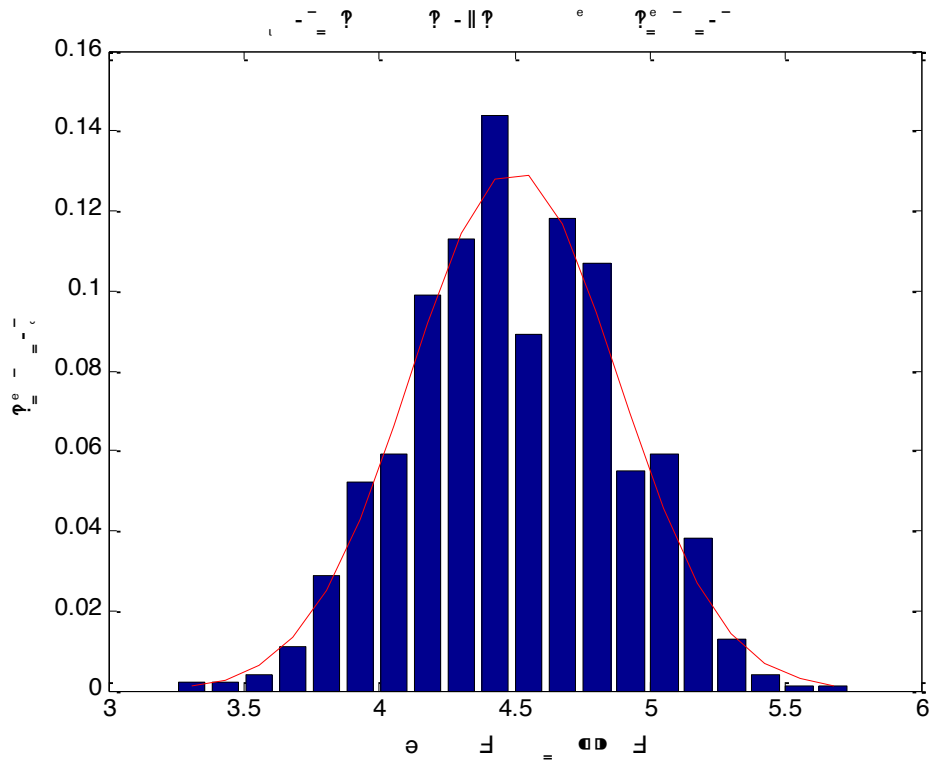


Рис. 5.11. Гістограма середньозваженого часу зчитування файлів

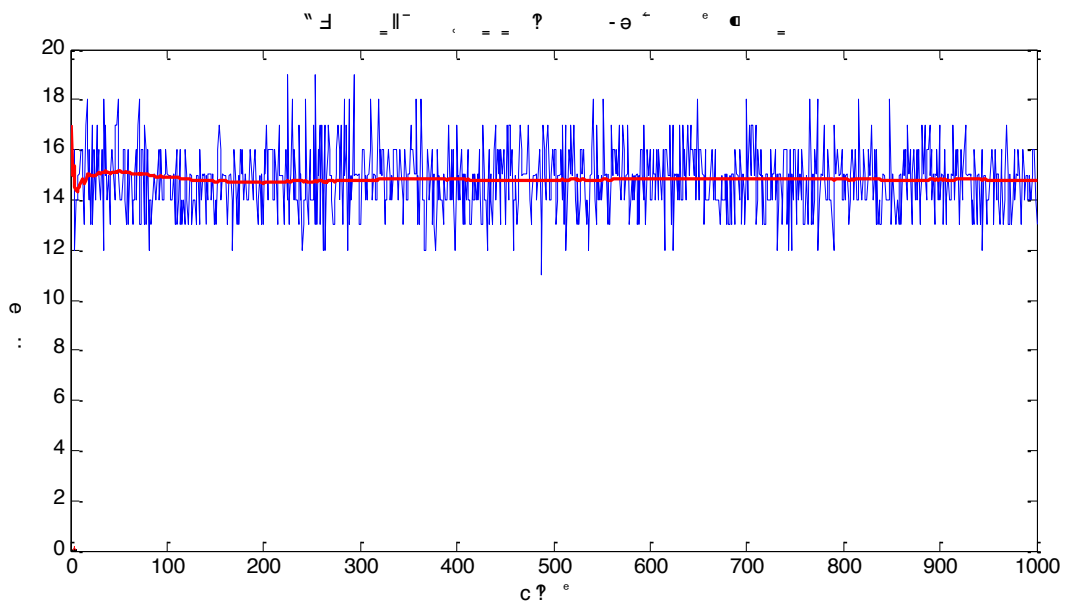


Рис. 5.12. Оцінка оптимального часу зчитування

Для порівняльного аналізу отриманий час зчитування файлів без оптимізації як сума довгот зчитування всіх фрагментів усіх файлів у серії з 1000

експериментів при нормальному розподілі довготи зчитування фрагментів файлів і при рівних вагових коефіцієнтах цільової функції. Умовне середнє і апостеріорна дисперсія помилки наведені рис.5.13-5.14. вибіркоче середнє дорівнює 47.5732 модельних одиниць, що майже втричі перевищує оптимальний час зчитування.

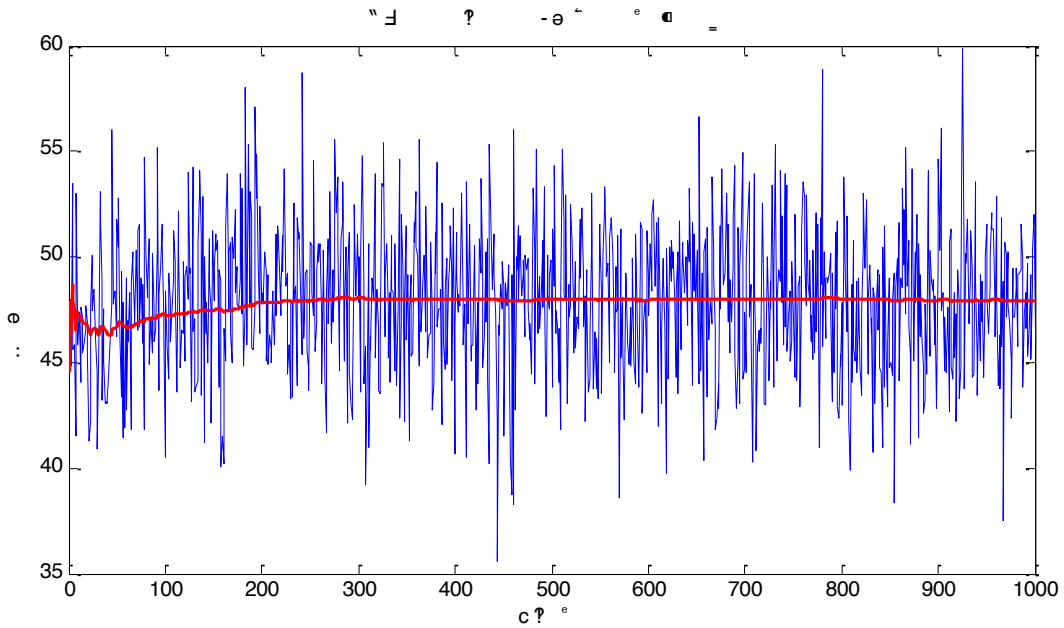


Рис.5.13. Оцінка часу зчитування файлів без оптимізації

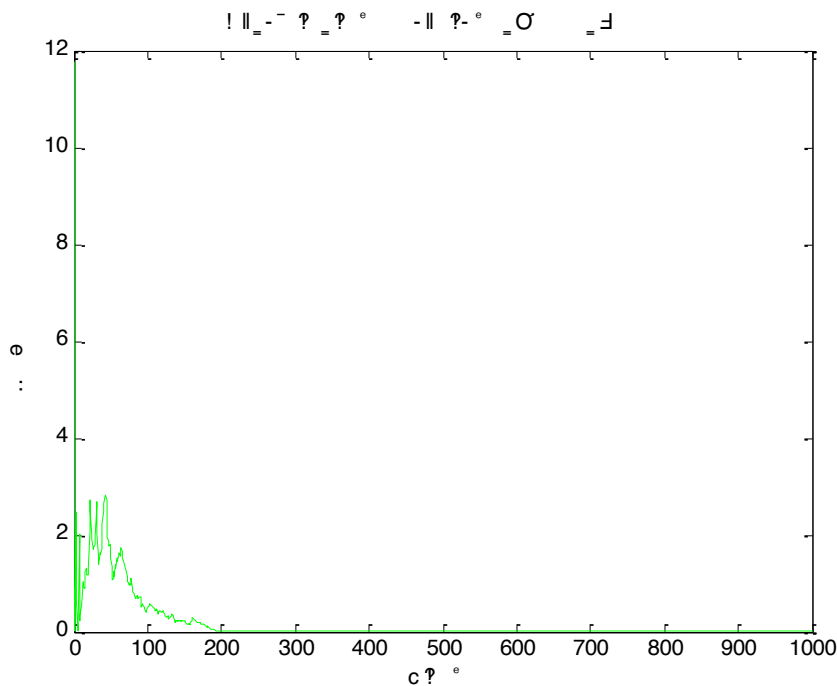


Рис. 5.14. Значення апостеріорної дисперсії помилки оцінки

5.4. Висновки до п'ятого розділу

1. З урахуванням спрощеної моделі процесу завантаження, коли на термінал надходить більше одного фрагмента одночасно, розроблена математична модель процедури впорядкування послідовності векторів, що відображають множину фрагментів.

2. З використанням теорії розкладів розроблена двохстадійна процедура прийняття оптимальних рішень (стадія апріорного планування в режимі offline і стадія складання розкладів в режимі online). Сформульована і вирішена задача оптимізації середньозваженого часу обслуговування, що зводиться до цілочисельного лінійного програмування.

3. Наведені приклади використання процедури мінімізації середньозваженого часу обслуговування. Для конкретного прикладу продемонстровано рішення у вигляді числових послідовностей і діаграма Гантта для одновимірної задачі й для задачі оптимізації трьох файлів.

4. Зіставлення одержаних результатів оптимізації процесів фрагментації методом динамічного програмування і цілочисельного лінійного програмування показало їх високу спроможність і близькість одержаних часових втрат. Дещо скорочені часові втрати для задач цілочисельного лінійного програмування пояснюються тим, що рішення задачі динамічного програмування вирішено для більш загальних умов.

5. Із результатів моделювання аналізу процедури мінімізації сумарного зваженого часу обслуговування слідує, що як середньозважене час зчитування файлів, так і умовне середнє оптимальний час зчитування файлів практично не залежить від закону розподілу значень вагових коефіцієнтів цільової функції і від закону розподілу значень тривалостей зчитування фрагментів файлів бенкетами і залишається постійним. Інваріантність процедур до законів розподілу випадкових тривалостей обробки фрагментів файлів пірами спрощує рішення і забезпечує стійкість процедури у великому діапазоні змін параметрів контенту. Крім того, результати імітаційного моделювання підтверджують адекватність обраних моделей і методів розв'язання задачі лінійного цілочисельного програмування та його достовірність.

РОЗДІЛ 6

АНАЛІЗ МЕТОДІВ ПЛАНУВАННЯ ІНФОРМАЦІЙНИХ СЕГМЕНТІВ ДЛЯ ПОТОКОВОГО ПІРИНГОВОГО ВІДЕО

6.1. Моделі планування потокового відео

Завданням планувальника є оперативне обслуговування вимог конкретного піра. Однак дана задача в різних системах вирішується по-різному.

На відміну від структурованих систем, таких як SplitStream, заснованих на створенні багатоадресних оверлейних мереж за допомогою одного або декількох статичних дерев, неструктурованим системам властива масштабованість і надійність. Це робить їх більш придатними для гетерогенного, динамічного середовища Інтернет, де піри мають різний обсяг завантаження і відправки й можуть приєднуватися чи покинути систему в будь-який випадковий момент часу.

Продуктивність неструктурованих систем P2P критично залежить від алгоритму вибору піра, визначеного для завантаження інформаційного сегмента. Відомо два типи алгоритмів, що класифікуються як push або pull в залежності від того, відправник або одержувач робить вибір про відправку сегмента.

Push-based більше підходить для скачування від відправників із обмеженою пропускною спроможністю лінії, оскільки поширення сегментів регулюється відправником, в залежності від його ресурсів.

Pull-based більше підходить для систем з обмеженням завантаження, оскільки швидкість запитів сегментів адаптується до завантаження ємності кожного піра. Це дозволяє пірам запитувати ті сегменти, які ближче всього до їх кінця відтворення.

Розглянемо більш докладно модель потокового пірингового відео і алгоритмів планування за схемою push-based, що в більшою мірою відповідає усталеній практиці.

Реалізація схеми push-based здійснюється в двох варіантах: з вибором піра-одержувача або вибором сегмента першим. Від цього залежить процедура планування інформаційних сегментів. Мережевий граф $G=(P,E)$, утворений безліччю пірів P , розташованих на вузлах мережі, містить в кожен момент часу пари $(u,v) \in P$ взаємодіючих сусідів. Нехай $C(u)$ буде сукупністю сегментів, які пір u отримує, а c – безліч сегментів, що належать піру. При цьому push-based схема з топологічних позицій реалізує собою відображення

$$Z: P * C^N \rightarrow P * C, \quad (6.1)$$

де N – потужність множини P .

Іншими словами: відображення формує два безлічі, відповідно $C \in c(u)$ - для відправки і безліч $C(u)$ – для піра-одержувача.

Уявімо кілька схем вибору пірів та сегментів:

1. RP-пір – одержувач вибирається випадково з безлічі сусідів u . (Випадковий сліпий вузол).

2. UP-пір одержувач вибирається рівномірно випадковим чином серед тих сусідів v з безлічі u , які задовольняють умову

$$C(u) \setminus C(v) \neq \emptyset,$$

де c – безліч можливих сегментів (груп фрагментів), що належать піру. Коли сегмент c обраний першим, вибір піра-одержувача обмежується тими сусідами v з u , які задовольняють умові $c \notin C(v)$.

3. DP – самі обмежені піри. Пір-одержувач вибирається рівномірно випадковим чином серед тих сусідів v з u які задовольняють умові $|C(u) \setminus C(v)|$ - максимум. Коли сегмент c обраний першим, вибір піра-одержувача обмежується тими сусідами v з u , які задовольняють умову.

4. LB – останній сліпий сегмент. Пір-відправник u вибирає найостанніший сегмент (тобто сегмент із найвищим індексом) у своїй безлічі $C(u)$.

5. LU – останній корисний сегмент. Пір-відправник u вибирає найостанніший сегмент c у своїй безлічі $C(u)$, що має хоча б один з його сусідів

v . Коли пір-одержувач v обраний першим c , c є найостаннішим сегментом у множині $C(u) \setminus C(v)$.

6. RU – випадковий корисний сегмент. Пір-відправник вибирає рівномірно випадковим чином сегмент c з безлічі $C(u)$ так що $c \notin C(v)$ для хоча б одного з його сусідів v . Сегмент пір-одержувач v обраний першим, c вибирається рівномірно випадковим чином із безлічі $C(u) \setminus C(v)$.

Широкий клас push-based схем впливає з комбінацій відбору алгоритмів цих вузлів і сегментів. Уявімо деякі нижче в таблиці 6.1. На рисунку 6.1 дається приклад вибору вузлів / сегментів у цих схемах. Представлена ситуація відповідає фазі, коли остання порція піра-відправника вже була отримана усіма своїми сусідами. Пропускна здатність піра-відправника витрачається даремно в цьому стані в схемах LB / UP і LB / RP, оскільки пір отримує дві або кілька копій одного і того ж сегмента.

Таблиця 6.1.

Деякі push-based схеми розподілу

Позначення	Схема
RP/LB	випадковий пір, останній сліпий сегмент
RP/LU	випадковий пір, останній корисний сегмент
DP/LU	пір з найбільш обмеженими правами, останній корисний сегмент
DP/RU	пір з найбільш обмеженими правами, випадковий корисний сегмент
LB/RP(=RP/LB)	останній сліпий сегмент, випадковий пір
LB/UP	останній сліпий сегмент, корисний пір
LU/UP	останній корисний сегмент, корисний пір
LU/DP	останній корисний сегмент, пір з найбільш обмеженими правами

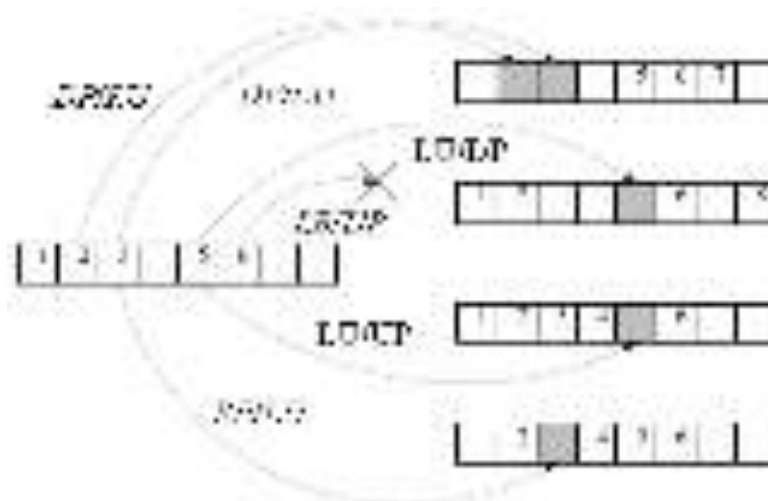


Рис. 6.1. Приклад вибору піра / сегмента піром-джерелом (зліва) при різних push-based схемах планування

6.2. Аналіз якості відео з урахуванням особливостей планування відео-контенту

Якість відео у користувача визначається рядом критеріїв [67]:

- співвідношенням пікових рівнів сигналу і шуму (peak signal-to-noise ratio - PSNR),
- середньоквадратичним відхиленням (MSE) для двох монохромних зображень L і K , одне з яких вважається зашумленим наближенням іншого,
- індексом структурного подібності SSIM, що враховує основні відмінності показників: яскравості, контрасту, структури зображення.

Аналіз втрат фрагментів. Як показують дослідження в P2P-мережах найбільш значущою для якості відтворення, серед інших, є втрата фрагментів. Очевидно, втрата кадру або фрагмента не настільки небезпечна для технологічного процесу, оскільки при цьому відбувається заміщення втрачених кадрів останніми прийнятими дублікатами, то на якості сприйняття така втрата позначається негативно.

Кількість втрачених фрагментів залежить від схеми основних мережевих планувань фрагментів, затримок відтворення. При цьому фрагмент вважається загубленим, якщо він не був отриманий протягом свого часу відтворення. Графік рівня втрат фрагментів як функцію від бітрейта представлений на рис. 10.4. Наведемо ряд кількісних характеристик, пов'язаних із втратою фрагментів. При обліку, що плануванні використані схеми LUs / Pur. Експеримент був організований за таких умов:

- Планувальник керував оверлеями 1000 пірів, пов'язаних відповідно до виду випадкового n-регулярного графа 20 ступеня;
- Кожен пір, також як і джерело, мав межу смуги пропускання завантаження 1 Мбіт / с, швидкість завантаження перевищує цю цифру;
- Моделювання засноване на поширенні 2000 фрагментів, але тільки 100 з них, які знаходяться в середині, використовувалися для оцінки якості відео;
- Кожне моделювання повторювалося 50 раз (при різних випадкових топологіях) і були обчислені середні результати.

З графіків випливає, що значні втрати фрагментів відзначаються зі збільшенням бітрейту (швидкості передачі відеопотоку). Це пов'язано з перевантаженням потокової системи P2P.

Іншою причиною втрат фрагментів є збільшення швидкості кодування відео, коли при збільшенні розміру фрагмента його дифузія займає більше часу, яке в деяких випадках перевищує допустимий.

На рис. 6.3 і 6.4 представлені характеристики PSNR і SSIM без урахування втрат фрагментів при різних затримках відтворення. Затримки даються в одиницях періодів кадрів. З графіків випливає, що фрагментація з урахуванням медіаструктури кадрів, при наявності фрагментів з GOP розмірів виявляється більш якісною порівняно з випадком, коли ця структура не враховується і фрагменти генеруються фіксованого розміру.

З верхніх графіків випливає, що при низькій затримці $T_s = 12$ якість відтворення прийнятого потоку падає при зростанні ступеня втрат фрагментів.

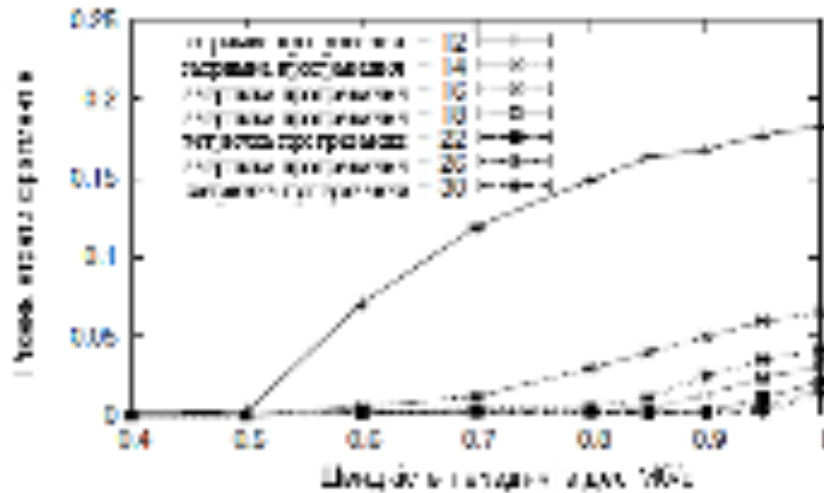


Рис. 6.2. Рівень втрати фрагментів як функція бітрейта відео і затримки відтворення

Рис. 6.3 і 6.4 показують PSNR і SSIM при заданій цільовій затримці відтворення, в залежності від бітрейту медіаданих і застосованої схеми фрагментації. Всі графіки показують якість (PSNR або SSIM) без втрати фрагмента (якість кодованого потоку в джерелі, яке є функцією датчика і служить тільки для довідки); якість, досягнута за наявності фрагментів з GOP розміром (фрагментація з урахуванням структури медіа); і якість, досягнута без урахування структури медіа. Це дозволяє зрозуміти вплив співвідношення втрат на якість відео.

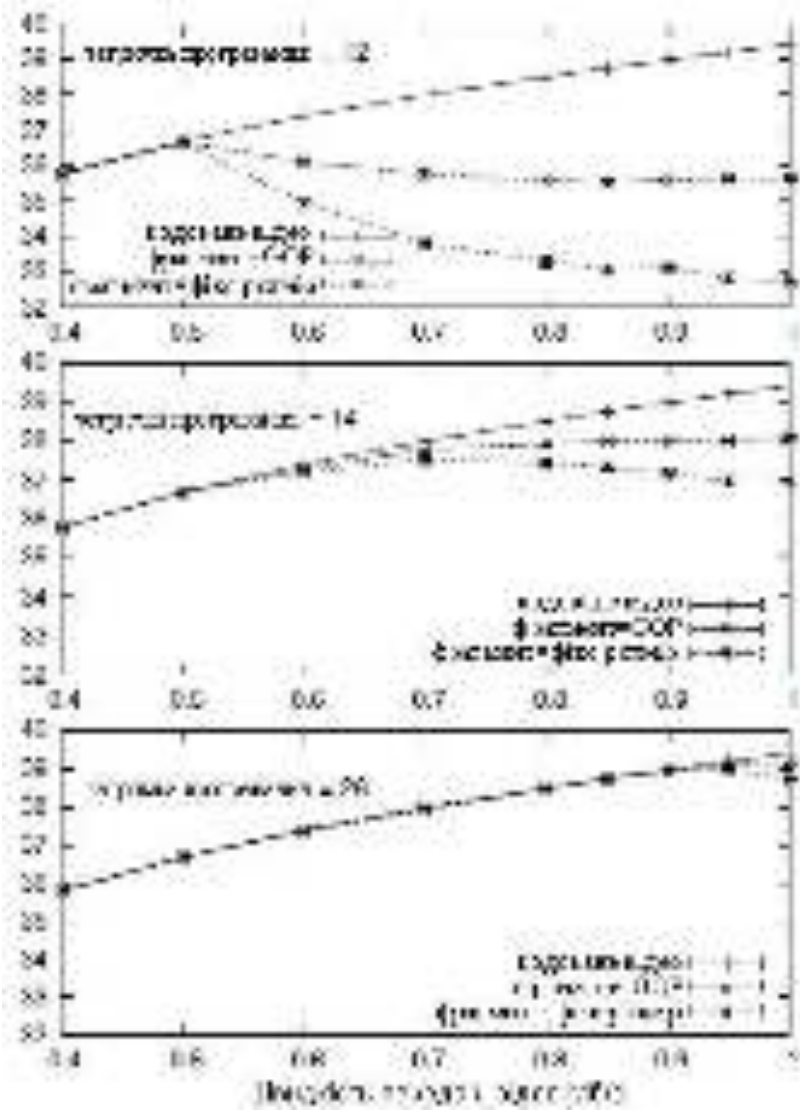


Рис. 6.3. PSNR [дБ] при фрагментації з урахуванням медіаструктури і без як функція швидкості кодування відео при різних затримках відтворення

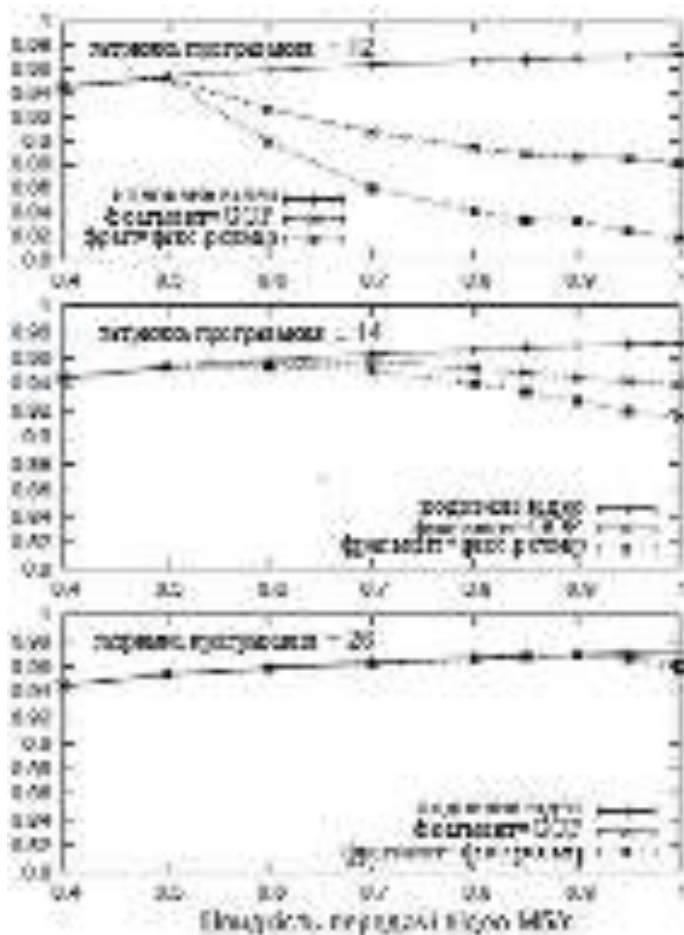


Рис. 6.4. SSIM при фрагментації з урахуванням і без урахування медіа-структур як функція швидкості кодування відео при різних затримках відтворення

Наприклад, верхній графік на рис. 6.3 показує, що відбувається, якщо затримка відтворення встановлюється відносно низькою $T=12$: в цьому випадку як тільки ступінь втрати фрагментів починає зростати (при бітрейті 0,5 Мбіт / с відповідно до рис. 6.3.) Якість прийнятого потоку значно зменшується щодо якості вихідного кодованого потоку. PSNR завжди набагато вище в разі врахування медіаструктури при фрагментації, ніж при фрагментації без медіаструктури.

При збільшенні затримки відтворення до $T=14$ для фрагментації з урахуванням медіаструктури PSNR може бути збільшено за рахунок збільшення бітрейта приблизно до 0.8 Мбіт / с (в разі фрагментації без урахування структури

відео, збільшення бітрейта вище 0,6 Мбіт / с не призводить до зростання PSNR, а збільшення його на 0,8 Мбіт / с зменшує PSNR).

У нижній частині рисунка показано якість відео з великою затримкою відтворення $T = 26$, що знижує втрати фрагментів (рис. 6.4) до мінімуму і дозволяє домогтися високої якості відео лише за рахунок збільшення бітрейта приблизно до 0.95Мбіт / с.

На рис. 6.5 представлені результати планування фрагментів і пірів, що позначаються на якості відео. Показана швидкість втрати фрагментів і значення PSNR, отримані з різних планувальників (а саме LUC / RUp, LUC / ELp і DLc / ELp6).

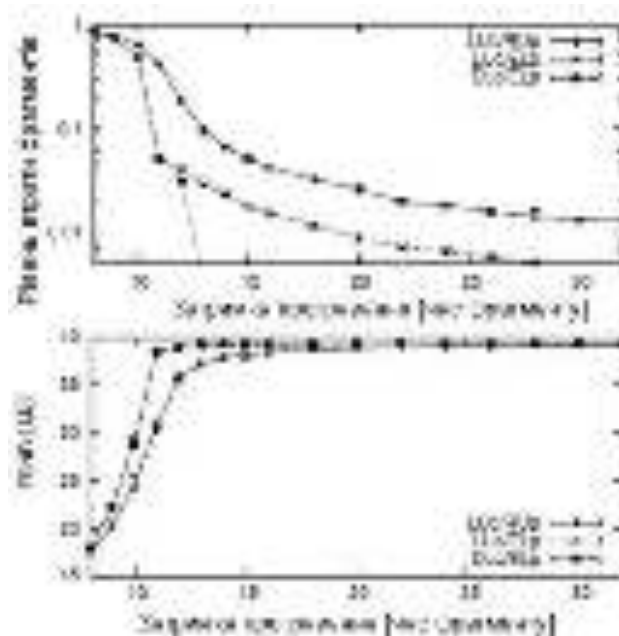


Рис. 6.5. Рівень втрати фрагментів і PSNR з різними планувальниками і фрагментами з GOP-розмірами як функція затримки відтворення

Виходячи з представлених результатів, можна зробити висновок, що фрагментація з урахуванням медіаструктури забезпечує значно вищу якість (PSNR і SSIM), ніж фрагменти без урахування цієї структури. Більш того, в разі відсутності втрат фрагментів, PSNR збільшується з бітрейтом, а одночасно з цим

система розподілу не може впоратися зі збільшенням обсягу трафіку, одночасно підтримуючи обмеження затримки відтворення.

6.3. Висновки до шостого розділу

1. У пірингових мережах найбільш використовуваними є алгоритми планування потокового відео, що класифікуються, як push або pull в залежності від того, відправник або одержувач приймає рішення щодо відправки сегментів. В умовах обмеженості ресурсів мережі кращими показниками володіє push-based. Проведено аналіз рівня втрат фрагментів залежно від величин бітрейта відео [Мб/сек] і затримок відтворення за критеріями відносини пікового рівня сигналу шуму (PSNR-peak signal-to-noise), середньоквадратичного відхилення (MSE) й індексу структурного подібності SSIM. При аналізі використані стандартні методики P2PTVSim 3 і SSSim.

2. Аналіз проведено з використанням схеми планування LUp / RUp (останній корисний фрагмент, випадковий корисний пір) показує, що рівні втрат фрагментів зростають зі зростанням швидкості передачі відео, при цьому зменшення затримки також призводить до збільшення цих втрат. Причиною цього є те, що зі збільшенням розмірів фрагментів відбувається зростання даних затримок (дифузії фрагментів).

3. Результати аналізу якості, визначеного за критерієм PSNR і SSim, проведені без урахування втрат фрагментів, показують, що при зменшенні затримки якість відтворення прийнятого потоку падає. Одночасно з цим облік медіаструктури кадрів і фрагментів дозволяє компенсувати дане падіння. Якість вдається зберегти також при великих допустимих затримках. Найбільші втрати якості в системах, де медіаструктура не враховується.

ВИСНОВКИ ПО РОБОТІ

1. Пірингові файлообмінні мережі є оверлейними мережами з передачею контенту через Інтернет. Область застосування мереж: паралельне програмування, кешування даних, резервне копіювання даних. Завдяки високій живучості, відмовостійкості, масштабованості пірингові мережі є перспективним напрямком розвитку, і знаходять застосування в системах управління виробництвом, надання послуг споживачам, зокрема при передачі відеоконтенту, в системах Bitcoin.

2. Якість надання послуг по доставці заявлених файлів залежить від великої кількості чинників, що супроводжують цей процес. Серед множини задач, які необхідно вирішувати при наданні послуг по доставці заявлених файлів, центральною є задача фрагментації, що в загальному вигляді представляє собою процедуру упорядкування розкладу фрагментів з інтервальною тривалістю обслуговування. В якості математичної моделі процесу фрагментації використано дискретно-неперервну модель з управлінням на кожному кроці і з обмеженням передування.

3. Проведено аналіз якості функціонування пірингової мережі при різній її динаміці. Показано, що якість пірингової мережі зростає зі збільшенням числа активних її учасників. Водночас при різкому збільшенні кількості учасників може спостерігатися зниження якості, що пов'язано з відсутністю контенту для завантаження у щойно підключених пірів.

4. Розроблено методику аналізу стану контенту на вузлах P2P мережі. Доведено, що успішне отримання заданої послуги залежить від того, наскільки швидко і якісно вдається виявити необхідний контент, що знаходиться в пам'яті пірів даної мережі. Реплікація контенту визначається наявністю достатньої кількості копій у мережі й пам'яті для зберігання контенту. Отримано залежність середнього числа заявок від числа заявлених файлів. Середнє число успішно виконаних заявок пропорційне числу заявлених файлів. Водночас при використанні алгоритму реплікації квадратного кореня ця залежність нелінійна,

що робить її більш привабливою в умовах значних навантажень.

5. Розглянуто методи оптимізації процесу завантаження файлів, де оптимізації підлягає відбір кращого фрагмента серед наданих на кожному кроці дискретизації і вирішено задачу оптимізації процесів упорядкування фрагментів самого файлу для запису його в буфер з метою гладкого відтворення без затримки. Сформульовано рекомендації щодо вибору на кожному кроці двох з m - представлених фрагментів, що на 17-20 дБ покращує якість рішення в порівнянні з одиничним вибором.

6. З використанням методу динамічного програмування вирішено задачу оптимальної фрагментації скачуваного файла з використанням дискретно-неперервної моделі при відповідному управлінні дискретністю процесу, що забезпечує впорядковане відтворення фрагментів при узгодженні черговості їх надходження. В ході аналізу встановлено, що при оптимізації в 3 рази скорочується час скачування, а вибіркова дисперсія при цьому зменшується в 2,2 рази. Це забезпечує більшу оперативність надання послуг і мінімізацію втрат фрагментів.

7. З урахуванням медіаструктури відеоконтенту на основі цілочисельного лінійного програмування одержано рішення задачі фрагментації для різних відрізків часу обслуговування. Аналіз результатів показав, що за даних умов вдається дещо скоротити час на формування файлу на 10-15%.

8. Проведено аналіз якості надання послуг відео з урахуванням особливостей планування контенту. Представлені результати дозволяють зробити висновок, що фрагментація з урахуванням медіа-структури забезпечує значно вищу якість (PSNR і SSIM), ніж фрагментація без такого урахування. Аналіз кількості втрат фрагментів збільшується з бітрейтом, а одночасно з цим система розподілу не може завжди впоратися зі збільшенням обсягу трафіку, одночасно підтримуючи обмеження затримки відтворення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Постанова Кабінету міністрів України від 12 грудня 2005 р. № 1208 «Про затвердження Національної таблиці розподілу смуг радіочастот України» із змінами і доповненнями, внесеними постановами, від 12 квітня 2017 р. № 265.
2. Постанова Кабінету міністрів України від 9 червня 2006 р. № 815 «Про затвердження Плану використання радіочастотного ресурсу України» із змінами і доповненнями, внесеними постановами, від 12 квітня 2017 р. № 265.
3. Закон України Верховної Ради України «Про телекомунікації» від 18 листопада 2003 р. № 1280-IV.
4. Закон України «Про радіочастотний ресурс України» зі змінами і доповненнями згідно із Законом № 5459-VI від 16.10.2012 р.
5. Закон України «Про телебачення і радіомовлення». Відомості Верховної Ради України (ВВР). – 1994. - № 10. – с. 43
6. Теплицкая С.Н. Анализ качества оценки случайных величин / С.Н. Теплицкая, Е.О. Поповская, Я.Т. Хусейн // Радиотехника: Всеукр. межвед. науч.- техн. сб. – 2010. – Вып. 163. – С. 204-208.
7. Saburova S.A. Prospects for service platform pre – IMS / S.A. Saburova, E.V. Bondar, E.O. Popovskaya // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2010.– Вып. 163. – С.13 -19.
8. Мельникова Л.И. Управление перегрузкой сетевых ресурсов с использованием процедуры Калмана-Бьюси / Л.И. Мельникова, С.Н. Горяева, Е.О. Поповская // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2009. – Вып. 159. – С. 35-41.
9. Поповская Е.О. Анализ качества двумерного рекурсивного фильтра / Е.О. Поповская, Б.С. Тур, Я.Т. Хуссейн // Восточно-Европейский журнал передовых технологий. – 2012. – Том 1. – № 9(55). – С. 26-28.
10. Saburova S.A. Development of services in the IP multimedia subsystem / S.A. Saburova, E.O. Popovska, Y.T. Hussein // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2009. – Вып. 159. — С. 229 - 233.

11. Москалец Н.В. Модели одноранговых пиринговых сетей для передачи потоков IPTV/ Н.В. Москалец, Е.О. Поповская // "Международный научно-исследовательский журнал". – 2014. – Ч. 2. – № 4(23). – г. Екатеринбург. – С. 46-50.
12. Ощепков М.Ю. Разработка математических моделей одноранговых сетей с передачей IPTV потоков / М.Ю. Ощепков, Е.О.Поповская // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. – Ростов-на-Дону: ПЦ "Университет" СКФ МТУСИ'2014.– Ч.1. – С. 304 - 307.
13. Ощепков М.Ю. Математическое моделирование пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // «Science and world» International scientific journal. – 2014. – № 3(7). –Vol.1. – С. 195 – 197.
14. Москалец Н.В. Оптимизация суммарного взвешенного времени обслуживания в пиринговой сети / Н.В. Москалец, Е.О.Поповская, К.А. Тарасов // Международный научно-исследовательский журнал. – 2016. – № 6-2(20). – С. 110-116.
15. Москалец Н.В. Функциональная модель управления в сетях пиринговой связи / Н.В. Москалец, Е.О.Поповская // Радиоэлектроника и информатика. – ХНУРЕ. – 2017. –№1. – С. 48-52.
16. Москалец Н.В. Фрагментация TV-контента в P2P-сетях с помощью процедуры динамического программирования / Н.В. Москалец, Е.О.Поповская // Зв`язок. – 2017. – №2(126). – С.14-17.
17. Поповская Е.О. Анализ модели трафика реального времени и данных / Сб. науч. тр. 4-го Международного радиоэлектронного форума "Прикладная радиоэлектроника" МРФ-2011– Х.: АНПРЭ, Украина, Харьков. – 2011. – С. 130–131.
18. Поповская Е.О. Математические модели одноранговых сетей // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – 2014. – Том 4. – С. 79 – 80.

19. Oshchepkov M. Ways for organization of IPTV services over peer-to-peer networks / M. Oshchepkov, K. Popovskaya // Modern problems of radio engineering, telecommunications and computer science, Proceedings of the XII-th International Conference TCSET' 2014, February 25- March 1, Slavske. – P. 568.
20. Способы организации услуги IPTV поверх пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // 10-я Международная молодежная научно-техническая конференция РТ-2014 «Современные проблемы радиотехники и телекоммуникаций». – Севастополь. –12-17 мая 2014. – С. 95.
21. Moskalets N. Mathematical scheduling models of IPTV peer to peer networks / N. Moskalets, E. Popovskaya // Матеріали 3-ї Міжнародної наукової конференції «Інформація, комунікації, суспільство». – Славське. – 21-24 травня. – 2014. – С. 20-21.
22. Linnik E.V. Optimization of the total weighted holding time in a p2p network / E.V. Linnik, K.O. Popovskaya // Материалы IV международной научно-практической конференции «Фундаментальные и прикладные науки сегодня». – 20-21 октября 2014 г. – North Charleston, USA, Том 3. – С. 160 -163.
23. Popovsky V.V. Mathematical model of P2PTV-network dynamics / V.V. Popovsky, K.O. Popovska // 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2014), October 14-17, 2014. – Kharkiv, Ukraine. – С. 23 - 25.
24. Москалец Н.В. Методы управления потоком фрагментов в пиринговых сетях / Н.В. Москалец, Е.О. Поповская // Збірник матеріалів Міжнародної науково-технічної конференції «Проблеми телекомунікацій». – Київ: НДІ телекомунікацій ІТС КПІ ім. Ігоря Сікорського, 2017. – С.1-3.
25. Kuzminich I. Minimization process of the total weighted holding time in a P2P network / I. Kuzminich, K. Popovska // 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2016), October 4-6, 2016. – Kharkiv, Ukraine. – С. 154 – 155.
26. Popovsky V. Searching optimization method for requested media data in peer-to-peer network / V. Popovsky, K. Popovska // 2017 Fourth International Scientific-

- Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2017), October 10-13, 2017. – Kharkiv, Ukraine. –P.1-4.
27. Abeni L. On the Optimal Scheduling of Streaming Applications in Unstructured Meshes / L. Abeni, C. Kiraly, R. Lo Cigno // Proc. of IFIP Networking. – May 11–15, 2009. – Aachen. – P. 25.
28. Abeni L. SSSim: a Simple and Scalable Simulator for P2P Streaming Systems / L. Abeni, C. Kiraly, R. Lo Cigno // Proc. of IEEE CAMAD'09. – June 12 2009. – Pisa.
29. Arad C. Building and evaluating p2p systems using the kompics component framework. In Peer-to-Peer Computing / C. Arad, J. Dowling, S. Haridi // IEEE Ninth International Conference on. – 2009. – P. 93–94.
30. Адами Амину, Аппроксимация вероятностных характеристик модели сети P2P / Адами Амину, Ю.В. Гайдамака // Инф.упр.сети, 2013. – С.4 – 10.
31. Лазарев А.А., Свойство оптимальных расписаний задачи теории расписаний минимизации суммарного взвешенного момента окончания для одного прибора / А.А. Лазарев, А.Г. Кварцхелия // М: Автоматика и телемеханика №10, 2010. – С.80-90.
32. Поповский В.В. Математические основы управления и адаптации в телекоммуникационных системах: учебник / В.В. Поповский, В.Ф. Олейник. – Харьков: СМИТ, 2011. – 362 с.
33. Ноттль В.В. Оптимальная сегментация экспериментальных кривых / В.В. Ноттль, И.Б. Лучник., В.Г. Яковлев // АиТ. –1983. – №8. –С. 84-95.
34. Расина И.В. Оптимизация управления в билинейных системах / И.В. Расина, О.В. Батурина // АиТ. –2013. – №5. – С. 102-111.
35. Zhi-Hui Lu, Ye Wang, Yang Richard Yang. An Analysis and Comparison of CDN-P2P-hybrid Content Delivery System and Model // JCM. – 2012. – 7(3). – P. 232-245.
36. Gurman V.I. Discrete-continuous Representations of Impulsive Processes in the Controllable Systems / V.I. Gurman, I.V. Rasina // Autom. Remote Control. – 2012. – V. 73. – No.8. – P. 1290-1300.
37. Gurman V.I., Ni Ming Kang. Degenerate Problems of Optimal Control. I, II, III // Autom. Remote Control. – 2011. – V. 72. –No.4. – P. 727-739; No.5. – P. 929-943.

38. Van der Shaft A.J., Schumacher H. An Introduction to Hybrid Dynamical Systems // Springer-Verlag. – London Ltd. – 2000.
39. Tarkoma S. Overlay Network / Toward Information Networking // CRC Press, – 2010.
40. Richard E. Bellman Applied Dynamic Programming / Richard E. Bellman, Stuart E Dreyfus. – University Press. – 1962. – P. 363.
41. Wehrle K. Modeling and tools for Network simulation / K. Wehrle, M. Gunes, J. Gross // Springer, Heidelberg, Dordrecht, London, New York. – 2010.
42. Дорт-Гольц А.А. Анализ функционирования наложенных сетей в сетях операторов // М.Электросвязь №3. – 2014. – С.22-27.
43. Гордон В.С. Задачи теории расписаний с заданным частичным порядком / В.С. Гордон, А.Б. Долгий // М. Автоматика и телемеханика. – №10. – 2010. – С.15-27.
44. Baptiste Ph. Scheduling Equal-Length Jobs on Identical Parallel Machines // Discrete Appl.Math. – 2000. – №103. – P. 21-32.
45. A. P. Couto da Silva, E. Leonardi, M. Mellia, M. Meo. Exploiting heterogeneity in p2p video streaming// In Submitted for publication. – 2009.
46. Venkata N. Padmanabhan Distributing streaming media content using cooperative networking / Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou Kunwadee Sripanidkulchai // In NOSSDAV. – 2002. – P. 177-186.
47. Таненбаум Э. Компьютерные сети 5-е изд. / Э. Таненбаум, Д. Уэзеролл // СПб.: Питер, 2012. — С. 960.
48. Олифер В. Компьютерные сети. Принципы, технологии, протоколы. Учебник / В. Олифер, Н. Олифер // Питер. – 2016. – С. 996.
49. Куроуз Д. Компьютерные сети. Нисходящий подход / Д. Куроуз, К. Росс // Эксмо. – 2016. – С. 450.
50. Сергеев А. Основы локальных компьютерных сетей // Лань. – 2016. – С. 184.
51. Куроуз Д. Компьютерные сети. Настольная книга системного администратора / Д. Куроуз, Т. Росс // Эксмо. – 2016. – С. 912.
52. Pinedo M. Scheduling Theory, Algorithms and Systems // Springer-Verlag. – 2008.

53. Dan G. On the performance of multiple-tree-based Peer-to-Peer live streaming / G. Dan, V. Fodor, I. Chatzidrossos // In INFOCOM. – 2007.
54. С. Равал Децентрализованные приложения. Технология Blockchain в действии // Питер. – 2017. – С. 192.
55. Popovskii V.V. Video traffic delivery over the Internet / Popovskii V.V., Popovska K.O. // LAMBERT Academic Publishing. – Germany. – 2017. – С.161.
56. Поповский В.В., Лемешко А.В., Мельникова Л.И., Андрушко Д.В. Обзор и сравнительный анализ основных моделей и алгоритмов многопутевой маршрутизации в мультисервисных телекоммуникационных сетях // Прикладная радиоэлектроника. 2005. Том.4. Вып. No 4. – С. 372-382.
57. Amir H. Rasti, Reza Rejaie, Walter Willinger. Understanding the effect of p2p overlay on the as-level underlay // In CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference. – New York, NY, USA. – 2007. – P. 1-2.
58. Sylvia Ratnasamy, Mark H, Richard Karp, Scott Shenker Topologically aware overlay construction and server selection // In Proceedings of IEEE Infocom. – 2002.
59. Reza Rejaie, Antonio Ortega Pals: peer-to-peer adaptive layered streaming // In NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video. – New York, NY, USA. – 2003.– P. 153-161.
60. Dongni Ren, Y. T. H. Li, S. H. G. Chan. On reducing mesh delay for peer- to-peer live streaming // In INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. – 2008. – P. 1058-1066.
61. Tim Roughgarden, Éva Tardos How bad is selfish routing // Journal of the ACM, 49. – 2002. – P. 236–259.
62. E. Salvadori, R. Lo Cigno, Zsoka Z. Dynamic Grooming in IP over Optical Networks Based on the Overlay Architecture // Optical Switching and Networking. – 3(2). – August 2006. – P. 118-133.
63. Sujay Sanghavi, Bruce Hajek, Laurent Massoulié Gossiping with multiple messages // In Proceedings of IEEE INFOCOM 2007. – Alaska, USA. – May 2007. – P. 2135-

2143.

64. Stefan Saroiu, P. Krishna Gummadi, Steven Gribble A measurement study of peer-to-peer file sharing systems // In SPIE Multimedia Computing and Networking (MMCN2002). – 2002.
65. S. Shalunov, R. Penno, R. Woundy ALTO Information Export Service // Internet-Draft draft-shalunov-alto-infoexport-00, Internet Engineering Task Force. – October 2008. – Work in progress.
66. T. Small, B. Liang, B. Li Scaling laws and tradeoffs in Peer-to-Peer live multimedia streaming // In ACM Multimedia. – 2006.
67. P. Srisuresh, K. Egevang Traditional IP Network Address Translator (Traditional NAT) // RFC 3022, IETF. – January 2001.
68. P. Srisuresh, B. Ford, D. Kegel State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs) // RFC 5128, IETF. – March 2008.
69. D. A. Tran, K. A. Hua, T. T. Do. Zigzag An efficient peer-to-peer scheme for media streaming // In Proc. of IEEE INFOCOM. – 2003.
70. A. Vitali, M. Fumagalli Standard-compatible Multiple-Description Coding (MDC) and Layered Coding (LC) of Audio/Video Streams // Internet Draft draft-vitali-ietf-avt-mdc-lc-00, IETF. – July 2005.
71. A. Vlavianos, M. Iliofotou, Faloutsos BiTos: Enhancing BitTorrent for supporting streaming applications // In 9th IEEE Global Internet Symposium 2006. – April 2006.
72. D. J. Watts, S. H. Strogatz Collective dynamics of 'small-world' networks // Nature, 393(6684). – June 1998. – P. 440–442.
73. Bernard Wong, Aleksandrs Slivkins, Emin Sirer Meridian: A lightweight network location service without virtual coordinates // In SIGCOMM. – 2005.
74. Haiyong Xie, Richard Y. Yang, Arvind Krishnamurthy, Yanbin G. Liu, Abraham Silberschatz P4p: provider portal for applications // SIGCOMM Comput. Commun. Rev. – 38(4). – 2008. – P. 351–362.
75. Hui Zhang, Ashish Goel, Ramesh Govindan Improving lookup latency in distributed hash table systems using random sampling // IEEE/ACM Trans. Netw. – 13(5). –

1121–1134, 2005.

76. M. Zhang, L. Zhao, Y. Tang, J. Luo, S. Yang. Large-scale live media streaming over Peer-to-Peer networks through global Internet. In Workshop on advances in Peer-to-Peer multimedia streaming, 2005.

77. Meng Zhang, Yongqiang Xiong, Qian Zhang, Qian Yang. Optimizing the throughput of data-driven peer-to-peer streaming. In Lecture Notes in Computer Science, volume 4351, 2007.

78. Meng Zhang, Qian Zhang, Lifeng Sun, Shiqiang Yang. Understanding the power of pull-based streaming protocol: Can we do better? In IEEE JSAC, special issue on Advances in Peer-to-Peer Streaming Systems, 2007.

79. Xinyan Zhang, Jiangchuan Liu, Tak shing Peter Yum. Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming. In in IEEE Infocom, 2005.

80. Y.P. Zhou, D.M. Chiu, J.C.S Lui. A simple model for analysis and design of P2P streaming protocols. In IEEE ICNP, October 2007.

81. Luca Abeni, Csaba Kiraly, and Renato Lo Cigno. On the optimal scheduling of streaming applications in unstructured meshes. In IFIP Networking, 2009.

82. Vinay Aggarwal, Anja Feldmann, Christian Scheideler. Can isps and p2p systems co-operate for improved performance? ACM SIGCOMM Computer Communications Review (CCR), 37(3):29–40, July 2007.

83. Shahzad Ali, Anket Mathur, and Hui Zhang. Measurement of commercial peer-to-peer live video streaming. In Proc. of ICST Workshop on Recent Advances in Peer-to-Peer Streaming, august 2006.

84. R. Alimi, D. Pasko, L. Popkin, Y. Wang, and Y. Yang. P4P: Provider Portal for P2P Applications. Internet-Draft draft-p4p-framework-00, Internet Engineering Task Force, November 2008. Work in progress.

85. Á. Bakay, A. Bokor, D. Croce, A. Garcia, K. Kada, D. Károly, C. Kiraly, E. Leonardi, F. Mathieu, M. Mellia, L. Muscariello, T. Szemethy, and G. Tropea. Deliverable 1.1: First definition of metrics and algorithms by wp1.1. Technical report,

NAPA-WINE project. – 2008.

86. Suman Banerjee, Bobby Bhattacharjee, Christopher Kommareddy Scalable application layer multicast // In SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications. – vol. 32. – New York, NY, USA. – October 2002. – P. 205-217.

87. R. Bindal, Pei Cao, W. Chan, J. Medved, G. Suwala, T. Bates, A. Zhang Improving traffic locality in bittorrent via biased neighbor selection // In 26th IEEE International Conference on Distributed Computing Systems. – 2006. – July 2006. – P. 66-77.

88. Thomas Bonald, Laurent Massoulié, Fabien Mathieu, Diego Perino, Andrew Twigg Epidemic live streaming: optimal performance trade-offs // In Zhen Liu, Vishal Misra, and Prashant J. Shenoy, editors, SIGMETRICS. – Annapolis, Maryland, USA. – June 2008. – P. 325-336.

89. Lorenzo Bracciale, Francesca Lo Piccolo, Dario Luzzi, Stefano Salsano Opss: an overlay peer-to-peer streaming simulator for large-scale networks // SIGMETRICS Perform. Eval. Rev. – 35(3). – 2007. – P. 25-27.

90. Damiano Carra, Renato Lo Cigno, Ernst W. Biersack Graph Based Analysis of Mesh Overlay Streaming Systems // IEEE Journal on Selected Areas in Communications – JSAC, 25. – December 2007. – P. 1667-1677.

91. Damiano Carra, Renato Lo Cigno, Ernst W. Biersack Stochastic Graph Processes for Performance Evaluation of Content Delivery Applications in Overlay Networks // IEEE Transactions on Parallel and Distributed Systems. – 19(2). – February 2008. – P.247-261.

92. Miguel Castro, Peter Druschel, Y. Charlie, Hu Antony Rowstron Exploiting network proximity in peer-to-peer overlay networks // Technical report, Tech. Rep. MSR-TR-2002-82, Microsoft Research. – 2002.

93. Miguel Castro, Peter Druschel, Anne marie Kermarrec, Animesh Nandi, Antony Rowstron, Atul Singh Splitstream: High-bandwidth multicast in cooperative environments // In proceedings of IPTPS'03. – 2003.

94. David R. Choffnes, Fabián E. Bustamante Taming the torrent: a practical approach

to reducing cross-isp traffic in peer-to-peer systems // SIGCOMM Comput. Commun. Rev. – 38(4). – 2008. – P. 363-374.

95. Гуревич И.М. Динамическая модель сети связи / И.М. Гуревич // Теория телетрафика в системах информатики. – М.: Наука, 1989. – С.54-64.

96. A.P. Couto da Silva, E. Leonardi, M. Mellia, M. Meo A bandwidth aware scheduling strategy for p2p-tv systems // In Proceedings of the 8th International Conference on Peer-to-Peer Computing 2008 (P2P'08). – Aachen. – September 2008.

97.A. P. Couto da Silva, E. Leonardi, M. Mellia, M. Meo Exploiting heterogeneity in p2p video streaming // In Submitted for publication. – 2009.

98. Frank Dabek, Jinyang Li, Emil Sit, James Robertson, M. Frans Kaashoek, Robert Morris Designing a dht for low latency and high throughput // In NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation. – Berkeley, CA, USA. – 2004. – P.7.

99. Wang Y., Wang Z. Explicit routing algorithms for Internet Traffic Engineering // Proc. of 8th International Conference on Computer Communications and Networks. Paris, 1999. P. 582-588.

100. Byung gon Chun, Ben Y. Zhao, John D. Kubiatowicz Impact of neighbor selection on performance and resilience of structured p2p networks // In Proceedings of 4th International Workshop on Peer-to-Peer Systems (IPTPS). – 2005.

101. C. Griffiths, J. Livingood, R. Woundy Comcast's ISP Experiences in a Recent P4P Technical Trial // Internet-Draft draft-livingood-woundy-p4p - experiences-02, Internet Engineering Task Force. – October 2008.

102. Mohamed Hefeeda, Ahsan Habib, Boyan Botev, Dongyan Xu, Bharat Bhargava Promise: peer-to-peer media streaming using collectcast // In MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia. – New York, NY, USA. – 2003. – P.45-54.

103.Mohamed Hefeeda, Ahsan Habib, Dongyan Xu, Bharat Bhargava, Boyan Botev Collectcast: A peer-to-peer service for media streaming // ACM Multimedia 2003, 11. – 2003. – P. 68-81.

104. Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu, Keith W. Ross A measurement study of a large-scale p2p iptv system // *Multimedia, IEEE Transactions on.* – 9(8). – 2007. – P.1672-1687.
105. Weibo Gong Honggang Zhang, Yong Liu, Don Towsley On the interaction between overlay routing and mpls traffic engineering // In *ACM/SIGCOMM, Poster Session.* – 2004.
106. Yang hua Chu, Sanjay G. Rao, Srinivasan Seshan, Hui Zhang Enabling conferencing applications on the internet using an overlay multicast architecture // In *Proceedings of ACM SIGCOMM.* – 2001. – P. 55-67.
107. Wenjie Jiang, Dah-Ming Chiu, John C. S. Lui On the interaction of multiple overlay routing. *Perform. Eval.* – 62(1-4). – 2005. – P. 229-246.
108. T. Karagiannis, P. Rodriguez, K. Papagiannaki Should internet service providers fear peer-assisted content distribution? // In *ACM Internet Measurement Conference.* – October 2005.
109. Richard M. Karp, Abhijit Sahay, Eunice E. Santos, Klaus E. Schauer. Optimal broadcast and summation in the logp model // In *ACM Symposium on Parallel Algorithms and Architectures.* – 1993.
110. R. Keralapura, C. N. Chuah, N. Taft, G. Iannaccone Can ISPs take the heat from overlay networks? // In *ACM Workshop on Hot Topics in Networks (HotNets - III).* – November 2004.
111. S. Kiesel, L. Popkin, S. Previdi, R. Woundy, Y. R. Yang Application-Layer Traffic Optimization (ALTO) Requirements // *Internet-Draft draft-kiesel-alto-reqs-01.txt*, Internet Engineering Task Force. – November 2008.

ДОДАТКИ

ДОДАТОК А

Вихідними даними будуть такі величини, необхідні для складання розкладу:

$I = \{i\}$ – різноманітність робіт (контентів);

$T_{ik} = (t_{ik})$ – матриця тривалості обслуговування;

$W = \{w_i\}$ – вектор-стовпчик штрафів вимог;

$X = \{x_i\}$ – вектор-стовпчик моментів початку робіт;

m – число машин (пірів);

n – число програм (робіт);

g_i – кожна робота складається із g_i операцій, кожна операція має 3 індекси ($i = \overline{1, n}$);

j – номер операції в середині роботи ($j = \overline{1, g_i}$);

k – номер машини, на якій операція повинна виконуватися ($k = \overline{1, m}$);

Для спрощення рішення виконаємо кілька умов.

1. Кожна робота потребує в точності одного виконання на кожній із машин, $g_i = m; i = \overline{1, n}$; тобто всі роботи мають однакову кількість операцій, тобто m .
2. Для кожної операції вказана машина для її виконання, тобто відома матриця $\{r_{ijk}\}$.

Тоді цільова функція буде мати такий вигляд:

$$F(X) = W^T \cdot X \rightarrow \min \quad (1)$$

Сформулюємо обмеження. Із того, що одна машина в один момент часу може виконати не більш однієї роботи, слідує, що для кожної пари робіт $(i, j) \in I$ виконується тільки одна з нерівностей:

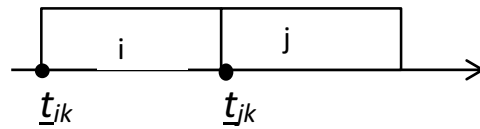
$$t_{ik} - t_{jk} \geq t_{jk} \quad (2)$$

Виконання робіт j передуює виконанню робіт i .

i	.
---	---

$$t_{ik} - t_{jk} \geq t_{ik} \quad (3)$$

Виконання роботи і передуює виконанню роботи j .



Таке обмеження типу «або – або» не можна описати в межах звичайного лінійного програмування і потребує введення булевих змінних.

$$\text{Пусть } Y_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

1 – якщо робота і передуює виконанню роботи j.

0 – у протилежному випадку.

Тепер сформульовані вище обмеження типу «або – або» можна записати у вигляді двох умов, кожна з яких повинна бути виконана:

$$(M + t_{jk}) \cdot Y_{ijk} + (t_{ik} - t_{jk}) \geq t_{jk} \quad (4)$$

$$(M + t_{ik}) \cdot (1 - Y_{ijk}) + (t_{jk} - t_{ik}) \geq t_{ik} \quad (5)$$

Нехай і передуює j, тобто $t_{ik} < t_{jk}$ и $Y_{ij} = 1$, тоді (4) точно збігається з другою умовою в (5). Умова (5) завдяки великому параметру M перетворюється в стійке обмеження, що не суперечить усій системі в цілому.

Сформулюємо нерівність, що забезпечує дотримання обмежень на порядок виконання операцій.

$$\sum_k r_{ijk} (t_{ik} + t_{ik}) \leq \sum_k r_{i(j+1)k} t_{ik} \quad (6)$$

Задача (1), (4-6) є задачею цілочисельного лінійного програмування і може бути вирішена одним із відомих методів.

ДОДАТОК Б

```
clear
```

```
% Таблиця
```

```
tabl=[1 0.95 0.9 0.8 0.7 0.5 0.3 0.2 0.1 0.05;  
1 0.004 0.016 0.064 0.148 0.455 1.074 1.642 2.71 3.84;  
2 0.103 0.211 0.446 0.713 1.386 2.41 3.22 4.6 5.99;  
3 0.352 0.584 1.005 1.424 2.37 3.66 4.64 6.25 7.82;  
4 0.711 1.064 1.649 2.2 3.36 4.88 5.99 7.78 9.49;  
5 1.145 1.610 2.34 3 4.35 6.06 7.29 9.24 11.07;  
6 1.635 2.2 3.07 3.83 5.35 7.23 8.56 10.64 12.59;  
7 2.17 2.83 3.82 4.64 6.35 8.38 9.8 12.02 14.07;  
8 2.73 3.49 4.59 5.53 7.34 9.52 11.03 13.36 15.51;  
9 3.32 4.17 5.38 6.39 8.34 10.66 12.24 14.68 16.92;  
10 3.94 4.86 6.18 7.27 9.34 11.78 13.44 15.99 18.31;  
11 4.58 5.58 6.99 8.15 10.34 12.9 14.63 17.28 19.68;  
12 5.23 6.3 7.81 9.03 11.34 14.01 15.81 18.55 21;  
13 5.89 7.04 8.63 9.93 12.34 15.12 16.98 19.81 22.4;  
14 6.57 7.79 9.47 10.82 13.34 16.22 18.15 21.1 23.7;  
15 7.26 8.55 10.31 11.72 14.34 17.32 19.31 22.3 25;  
16 7.96 9.31 11.15 12.62 15.34 18.42 20.5 23.5 26.3;  
17 8.67 10.08 12.00 13.53 16.34 19.51 21.6 24.8 27.6;  
18 9.39 10.86 12.86 14.44 17.34 20.6 22.8 26 28.9;  
19 10.11 11.65 13.72 15.35 18.34 21.7 23.9 27.2 30.1;  
20 10.85 12.44 14.58 16.27 19.34 22.8 25 28.4 31.4];
```

```
% Задача цілочисельного лінійного програмування
```

```
for i=1:12
```

```
    f(i)=normrnd(0.085,0.02,1);
```

```

end
for i=13:30
    f(i)=0;
end
% Constraints
A = sparse([-1 0 0 1 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 -1 0 0 1 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 -1 0 0 1 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
-1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 -1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 -1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
-1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 -1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 -1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
1 0 0 -1 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 1 0 0 -1 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;

```

```

0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0;
0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0;
1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 -1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;]);
lb = zeros(30,1);
int = 'IIIIIIIIIIIBBBBBBBBBBBBBBBBBBBB';
%b          =          [-3;-3;-3;-2;-1;-2;-3;-2;-2;-2;-1;-2;-3;-2;-2;-3;-2;-
2;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;27;-2;-3;-3;-3;-2;-2;-3;-2;-
];
for i=1:1000
    M=0;
    for j=1:18
        b(j)=normrnd(-4,1,1);
        M=M+b(j);
    end
    for j=19:36

```



```

    b(j)=-M;
end
for j=37:44
    b(j)=normrnd(-4,1,1);
end

% Build OPTI Problem
Opt = opti('f',f,'ineq',A,b,'lb',lb,'int',int)
% Solve
x = solve(Opt);
F(i)=f*x;

% Визначення середнього часу завантаження всіх завдань на всіх машинах з
% оптимізацією
a(1)=x(1);
a(2)=x(4);
a(3)=x(7);
a(4)=x(10);
t1=max(a)-round(b(1));
c(1)=x(2);
c(2)=x(5);
c(3)=x(8);
c(4)=x(11);
t2=max(c)-round(b(2));
d(1)=x(3);
d(2)=x(6);
d(3)=x(9);
d(4)=x(12);
t3=max(d)-round(b(3));
V(1)=t1;

```

```

V(2)=t2;
V(3)=t3;
TO(i)=max(V);
tn=0;
for k=1:12
    tn= tn+ b(k);
end
TN(i)=-tn;
end
% побудова гістограми
ksi=F;
N=1000;% довжина вибірки
t=1:N;% час
figure(2)
plot(t,ksi)
title('Время считывания всех файлов');% Підпис графіка
xlabel('время');% Підпис осі x графіка
ylabel('значение');% Підпис осі y графіка
display('матожидание')
m=mean(ksi)
display('дисперсия')
d=mean((ksi-m).^2)
% Побудова гістограми
[aa,bb]=hist(ksi,20);
dx=(max(ksi)-min(ksi))/20;
p1=aa/(N);%*dx
mx=bb*p1';
sigm=sqrt(mean ((ksi-mx).^2));
p2=exp(-(bb-mx).^2/(2*sigm^2))/(sigm*sqrt(2*pi))*dx;
figure (3)

```

```

bar(bb,p1)
hold on,
plot(bb,p2,'r')
title('Гистограмма распределения вероятностей');% Підпис графіка
xlabel('значение процесса');% Підпис осі x графіка
ylabel('вероятность');% Підпис осі y графіка
% перевірка гіпотези правдоподібності о гаусовій щільності розподілу
ймовірності
xu2=sum((aa-N*p2).^2./(N*p2))
svob=20-1-1-1;% кількість степеней свободи
lam=0.02;% рівень значущості;
li=2;
while tabl(li,1)<svob
    li=li+1;
end
lj=2;
while tabl(li,lj)<xu2 & lj<10,
    lj=lj+1;
end
ppi=tabl(1,lj);
if lam<ppi
    disp('Выдвинутая нами гипотеза не противоречит опытным данным')
else
    disp('Выдвинутая нами гипотеза противоречит опытным данным и ее надо
отбросить')
end
% Фільтр Робінса-Монро
t=1:1000;
H=1;

```

```

y=TO;% Спостережуваний процес
x0(1)=y(1);
P1(1)=(mean(TO)-x0(1))^2;
% Робим оцінку випадкової величини
%Фільтр Робінса-Монро
for k=1:999
K(k+1)=1/k; % коефіцієнт посилення фільтра
% оцінка на k + 1 - м кроці
x0(k+1)=x0(k)+K(k+1)*(y(k+1)-H*x0(k));
P1(k+1)=(mean(TO)-x0(k+1))^2;
end
figure(4)
plot(t,TO)
title('оптимальное время считывания файлов');% Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка
hold on
plot(t,x0,'r')
title('Оценка оптимального времени считывания файлов');%Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка
figure(5)
plot(t,P1,'g')
hold on
title('Апостериорная дисперсия ошибки оценки');% Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка

% Час зчитування без оптимізації
u=TN;% спостережувальний процес

```

```

x0(1)=y(1);
P1(1)=(mean(TN)-x0(1))^2;
% Робимо оцінку випадкової величини
%Фільтр Робінса-Монро
for k=1:999
    K(k+1)=1/k; % коефіцієнт посилення фільтра
    % оцінка на k + 1 - м кроці
    x0(k+1)=x0(k)+K(k+1)*(y(k+1)-H*x0(k));
    P1(k+1)=(mean(TN)-x0(k+1))^2;
end
figure(6)
plot(t,TN)
title('время считывания файлов');% Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка
hold on
plot(t,x0,'r')
title('Оценка времени считывания файлов');% Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка
figure(7)
plot(t,P1,'g')
hold on
title('Апостериорная дисперсия ошибки оценки');% Підпис графіка
xlabel('Время');% Підпис осі x графіка
ylabel('Значение');% Підпис осі y графіка

```

Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації:

1. Теплицкая С.Н. Анализ качества оценки случайных величин / С.Н. Теплицкая, Е.О. Поповская, Я.Т. Хусейн // Радиотехника: Всеукр. межвед. науч.- техн. сб. – 2010. – Вып. 163. – С. 204-208.
2. Saburova S.A. Prospects for service platform pre – IMS / S.A. Saburova, E.V. Bondar, E.O. Popovskaya // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2010. – Вып. 163. – С.13 -19.
3. Мельникова Л.И. Управление перегрузкой сетевых ресурсов с использованием процедуры Калмана-Бьюси / Л.И. Мельникова, С.Н. Горяева, Е.О. Поповская // Радиотехника: Всеукр. межвед. науч. - техн. сб. – 2009. – Вып. 159. – С. 35-41.
4. Поповская Е.О. Анализ качества двумерного рекурсивного фильтра / Е.О. Поповская, Б.С. Тур, Я.Т. Хуссейн // Восточно-Европейский журнал передовых технологий. – 2012. – Том 1. – № 9(55). – С. 26-28.
5. Saburova S.A. Development of services in the IP multimedia subsystem / S.A. Saburova, E.O. Popovska, Y.T. Hussein // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2009. – Вып. 159. — С. 229 - 233.
6. Москалец Н.В. Модели одноранговых пиринговых сетей для передачи потоков IPTV/ Н.В. Москалец, Е.О. Поповская // "Международный научно-исследовательский журнал". – 2014. – Ч. 2. – № 4(23). – г. Екатеринбург. – С. 46-50.
7. Ощепков М.Ю. Разработка математических моделей одноранговых сетей с передачей IPTV потоков / М.Ю. Ощепков, Е.О.Поповская // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. – Ростов-на-Дону: ПЦ "Университет" СКФ МТУСИ'2014.– Ч.1. – С. 304 - 307.
8. Ощепков М.Ю. Математическое моделирование пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // «Science and world» International scientific journal. – 2014. – № 3(7). –Vol.1. – С. 195 – 197.

9. Москалец Н.В. Оптимизация суммарного взвешенного времени обслуживания в пиринговой сети / Н.В. Москалец, Е.О.Поповская, К.А. Тарасов // Международный научно-исследовательский журнал. – 2016. – № 6-2(20). – С. 110-116.
10. Москалец Н.В. Функциональная модель управления в сетях пиринговой связи / Н.В. Москалец, Е.О.Поповская // Радиоэлектроника и информатика. – ХНУРЕ. – 2017. – №1. – С. 48-52.
11. Москалец Н.В. Фрагментация TV-контента в P2P-сетях с помощью процедуры динамического программирования / Н.В. Москалец, Е.О.Поповская // Зв'язок. – 2017. – №2(126). – С.14-17.
12. Поповская Е.О. Анализ модели трафика реального времени и данных / Сб. науч. тр. 4-го Международного радиоэлектронного форума "Прикладная радиоэлектроника" МРФ-2011– Х.: АНПРЭ, Украина, Харьков. – 2011. – С. 130–131. (очна участь із доповіддю)
13. Поповская Е.О. Математические модели одноранговых сетей // Материалы XVIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – 2014. – Том 4. – С. 79 – 80. (очна участь із доповіддю)
14. Oshchepkov M. Ways for organization of IPTV services over peer-to-peer networks / M. Oshchepkov, K. Popovskaya // Modern problems of radio engineering, telecommunications and computer science, Proceedings of the XII-th International Conference TCSET' 2014, February 25- March 1, Slavske. – P. 568. (очна участь із доповіддю)
15. Способы организации услуги IPTV поверх пиринговых сетей / М.Ю. Ощепков, Е.О.Поповская // 10-я Международная молодежная научно-техническая конференция РТ-2014 «Современные проблемы радиотехники и телекоммуникаций». – Севастополь. –12-17 мая 2014. – С. 95. (заочна участь)
16. Moskalets N. Mathematical scheduling models of IPTV peer to peer networks / N. Moskalets, E.Popovskaya // Матеріали 3-ї Міжнародної наукової конференції «Інформація, комунікації, суспільство». – Славське. – 21-24 травня. – 2014. – С. 20-21. (заочна участь)

17. Linnik E.V. Optimization of the total weighted holding time in a p2p network / E.V. Linnik, K.O. Popovskaya // Материалы IV международной научно-практической конференции «Фундаментальные и прикладные науки сегодня». – 20-21 октября 2014 г. – North Charleston, USA, Том 3. – С. 160 -163. (заочна участь)
18. Popovsky V.V. Mathematical model of P2PTV-network dynamics / V.V. Popovsky, K.O. Popovska // 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2014), october 14-17, 2014. – Kharkiv, Ukraine. – С. 23 - 25. (очна участь із доповіддю)
19. Москалец Н.В. Методы управления потоком фрагментов в пиринговых сетях / Н.В. Москалец, Е.О. Поповская // Збірник матеріалів Міжнародної науково-технічної конференції «Проблеми телекомунікацій». – Київ: НДІ телекомунікацій ІТС КПІ ім. Ігоря Сікорського, 2017. – С.1-3. (очна участь із доповіддю)
20. Kuzminich I. Minimization process of the total weighted holding time in a P2P network / I. Kuzminich, K. Popovska // 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2016), October 4-6, 2016. – Kharkiv, Ukraine. – С. 154 – 155. (очна участь із доповіддю)
21. Popovsky V. Searching optimization method for requested media data in peer-to-peer network / V. Popovsky, K. Popovska // 2017 Fourth International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T'2017), October 10-13, 2017. – Kharkiv, Ukraine. –P.1-4. (очна участь із доповіддю)

