УДК 004.7

*О.П. Ігнатенко*

# ОДНА ДИНАМІЧНА КОНФЛІКТНО КЕРОВАНА МОДЕЛЬ ВЗАЄМОДІЇ КОРИСТУВАЧІВ У ВІДКРИТИХ ІНФОРМАЦІЙНИХ СЕРЕДОВИЩАХ

Дана робота присвячена моделям ігрового керування потоками даних у мережах, що використовують TCP алгоритми керування перевантаженнями. В роботі представлено огляд проблеми, підкреслена її актуальність, описані існуючі підходи до вирішення. Для мереж зі змінними протоками даних і наявності конфліктуючих користувачів пропонується новий підхід до аналізу роботи мережі на базі конфліктно-керованих потокових моделей. Побудована модель одного класу мереж з фіксованим вікном доступу та доведене існування оптимального у сенсі швидкодії розв'язку задачі завантаження файлу. Розглянуто гру двох користувачів, що намагаються передавати дані через спільний канал та знайдені умови виникнення втрат.

## Вступ

**Актуальність проблеми.** Сучасні відкриті інформаційні середовища (ВІС) пов'язані практично з усіма областями людської діяльності. Інтернет стрімко увійшов у наше життя об'єднавши понад два мільярди користувачів у найбільшу в історії людства комунікаційну структуру. На сьогодні спостерігається стійкий розвиток ВІС у напрямку зростання розподіленості, інтелектуалізації і складності. Забезпечення стабільної роботи цих систем неможливе без адекватних моделей їх роботи – детерміністичних, стохастичних, імітаційних тощо. Кожна з них розв'язує окремий клас проблем, представляючи роботу ВІС в певному розрізі.

Важливим і актуальним напрямком в моделюванні ВІС є застосування потокових моделей, який дозволяє використати потужний апарат теорії керування і дослідити роботу ВІС «в цілому», відкидаючи деталі. В рамках цього підходу було отримано багато вагомих результатів в області аналізу стійкості поведінки мереж, які описані, в роботах Келі, Кумара, Сріканта, Мейна та інших. Цей підхід дозволяє проаналізувати структурні характеристики ВІС, що обумовлені топологією мережі, однак він, як правило, не враховує поведінку користувачів ВІС і їх потреби.

Розробка потокових моделей, які б враховували поведінку користувачів на основі використання ігрових підходів могла б заповнити цей пробіл. Зрозуміло, що ВІС, як правило, складаються з різнорідних обчислювальних і комунікаційних ресурсів. Відмінності у ресурсах різних частин системи і нерівномірність вхідного потоку завдань можуть призвести до значних коливань завантаженості ресурсів системи і погіршення ефективності роботи. Задача розподілу ресурсів у неоднорідних багатопроцесорних системах при здійсненні обчислень є, у загальному випадку, NP складною проблемою. Підхід потокового моделювання дозволяє знайти прийнятне наближення розв'язку за невеликий час, що є критично важливим у системах реального часу. Цей підхід започатковувався і розвивається наразі у роботах Вейса, Назараті, Прасани.

*Врахування при такому моделюванні конкуренції за ресурси між користувачами ВІС дозволило б більш адекватно вирішувати цю проблему.*

Зростання кількості користувачів ВІС, їх диференціація за завданнями та їх поведінка поставила перед розробниками нові проблеми, головна з яких полягає у необхідності забезпечення стійкої роботи і справедливого розподілу ресурсів між користувачами. Для розв'язання цих проблем були створені спеціальні алгоритми, що регулюють поведінку користувачів у мережі – протоколи. Перші протоколи були інженерними евристичними рішеннями. Пізніше Леонардом Клейнроком на базі теорії масового обслуговування був розроблений перший теоретично обґрунтований протокол. На сьогодні дослідниками запропоновано більше 50 реалізацій алгоритмів керування потоками даних (TCP протоколів). Загальна ситуація ускладнюється тим, що взаємодія різних протоколів часто носить конкурентний характер, що призводить до нерівномірного розподілу ресурсів. Питання стійкості і справедливості розподілу ресурсів досліджувалися Альтманом, Лоу, Паганіні. Слід відзначити, що розвиток IP телефонії, Grid і Cloud обчислень, трансляція відео і аудіо потоків значно загострила проблему гарантованого обслуговування користувачів. Внаслідок цього, особливу важливість набуває розвиток аналітичних методів дослідження роботи мереж з обслуговування різнотипних конфліктуючих користувачів. Конфлікт тут розуміється у сенсі конкуренції за ресурси, при цьому кожен з користувачів зацікавлений у стабільній роботі мережі. Ігрові підходи до аналізу процесів у мережах досліджувалися такими науковцями як Альтманом, Сандхольмом, Лоу та іншими.

*Однак до цього часу невирішеною залишається проблема створення інтегрованої потокової моделі взаємодії різних користувачів за умов різних протоколів і обмежених ресурсів.*

Зовсім інша проблема виникає, коли один з гравців зацікавлений у погіршенні роботи елементів ВІС. Це – ситуація зловмисного використання, так звана атака на відмову. Такі атаки можуть проявлятися у дуже різноманітні способи, але, так чи інакше, наслідком є значне погіршення якості або повне припинення роботи елемента ВІС. На сьогоднішній момент існує досить багато різних видів атак на відмову, кожна з яких використовує певну особливість побудови мережі або вразливості програмного забезпечення. За умов постійного зростання типів і потужності атак були здійснені спроби розробити теоретичні моделі виявлення і протидії, які б дозволили ефективно протидіяти цим негативним явищам. Основними напрямки стали моделі виявлення перевантажень (Іоанідіс, Флойд та інші), особливостей (Гіл, Полетто, Ченг, Кулкарні, Пенг, Дарховський) і аномальної поведінки (Денінг, Магларас, Чен). Задача протидії вирішувалась в рамках евристичних алгоритмів, які відсікали трафік атаки. У якості прикладів можна відзначити схему D-WARD, розроблену Е. Мірковіч та схему протидії Т. Пенга. Значна частина досліджень присвячена розробкам нових протоколів, мереж, засобів аутентифікації, які б унеможливили появу атак взагалі. Ефективність цих підходів, як правило, залежить від впровадження нових схем по всій мережі Інтернет, що значно зменшує їх практичну цінність.

*Застосування ігрових методів до аналітичного моделювання атак на відмову з урахуванням поведінки зловмисних користувачів ВІС та створення теоретично обґрунтованих стратегій протидії їм дозволило б досягти значних успіхів у вирішенні цієї проблеми.*

Все це дає підстави вважати, що розробка методів, моделей та засобів ігрового керування відкритими інформаційними середовищами за умов конфліктної взаємодії користувачів і атак на відмову є актуальною науково-технічною проблемою, вирішення якої сприятиме стійкому та безпечному функціонуванні ВІС.

## 1. Аналітичні моделі керування мережами

В першому розділі проведено аналіз існуючих моделей і методів керування відкритими інформаційними середовищами та виявлено невирішені актуальні задачі керування за умов конфліктної взаємодії та атак на відмову. Аналіз предметної області дозволяє виділити такі основні сутності:

- **користувачі**. Поведінка користувачів визначається задачами, що стоять перед ними. Ці задачі, як правило, полягають у виконанні певних операцій на серверах (завантаження даних, здійснення обчислень та інше). Взаємодія різних користувачів може бути конфліктною з огляду на обмежену кількість ресурсів системи;

- **сервери.** Сервери надають послуги користувачам через мережу. За умов обмеженості ресурсів їх основною задачею є справедливий розподіл своїх можливостей з метою виконання задач користувачів;
- **комунікаційна мережа**. Представляє складне середовище, що включає канали зв'язку, маршрутизатори, протоколи взаємодії користувачів, механізми керування перевантаженнями. До основних задач цієї системи належать: стабільна передача даних користувачів, стабільна і прогнозована робота, рівномірний розподіл ресурсів.

Опишемо основні типи аналітичних моделей, що використовуються для аналізу роботи мереж на сьогодні.

**Потокова модель керування окремим стаціонарним потоком даних.** Історично першим підходом опису мереж були моделі теорії масового обслуговування. Мережі масового обслуговування складаються зі скінченної кількості вузлів і характеризуються вхідним потоком запитів, що потребують обслуговування на вузлах, характеристиками схем маршрутизації і обробки на вузлах. Розглянемо детерміністичну потокову модель [1]. Нехай мережа складається з $N$ вузлів, позначимо $q_i(t)$ – чергу на $i$-тому вузлі. Потік вхідних пакетів, що потрапляє ззовні на вхід $i$-го вузла описується швидкістю $\lambda_i$, можливості обробки даного вузла – потужністю $u_i$, $i=1,...N$. Тоді динаміка мережі описується системою диференційних рівнянь:

$$\dot{\overline{q}}(t)=\overline{\lambda}+B\overline{u}, \qquad (1)$$

де $\overline{u}\in U$ – керування, $B$ – матриця, що описує структуру мережі. На змінні $\overline{u}$, $\overline{q}$, $\overline{\lambda}$ накладені наступні обмеження:

$$\overline{u},\overline{\lambda},\overline{q}\in R_+^N, C\overline{u}\le\overline{1}, \ \overline{q}\in Q, \qquad (2)$$

де $C$ – структурна матриця (визначає спільне використання потужностей на вузлах), $Q$ – компакт з $R_+^N$, що описує область в якій може знаходитися система. Допустимим позиційним керуванням системи (1) називається вимірна функція $u(\cdot):Q\to U$, яка задовольняє обмеженням (2). Керування $u(q(t))$ називається *стабілізуючим*, якщо розв'язок системи (1) для даного керування $q(t;q_0)\to 0$ при $t\to\infty$ для будь-якого $q_0\in Q$. Основною задачею, яка ставиться, зокрема, в роботах [1, 2] є пошук умов існування стабілізуючого керування для системи (1) – (2). Якщо задано критерій якості роботи системи $J(q(\cdot))$, то виникає задача знаходження оптимального керування:

$$J(q(\cdot))\underset{u(\cdot)}{\to}\min$$

$$\dot{\overline{q}}(t)=\overline{\lambda}+B\overline{u}, \ \overline{u},\overline{\lambda},\overline{q}\in R_+^N,$$

$$C\overline{u}\le\overline{1}, \ \overline{q}\in Q. \qquad (3)$$

Якщо критерій якості має вигляд часу приведення траєкторії до нуля (тобто розвантаження всіх черг) задача (3) стає задачею про оптимальну швидкодію.

Для таких моделей існує добре розвинена теорія, однак застосування її до реальних мереж ускладнене наступними факторами:

1) необхідність повної інформованості;
2) припущення про стаціонарність вхідного потоку;
3) відсутність у моделях (1)-(3) втрат і їх впливу на поведінку користувачів;
4) відсутність диференціації між користувачами;
5) умова існування стабілізуючого керування полягає в тому, щоб потужності системи завжди були більші потужності вхідного потоку.

В реальних системах ці умови порушуються майже завжди. Тим не менше моделі (1)-(3) дозволяють аналітично оцінити структуру мережі, описати її характеристики (виявити вузькі місця, обчислити час розвантаження тощо).

**Модель розподілу ресурсів Ф. Келі.** З розвитком Інтернету проблеми дослідження мереж з багатьма конкуруючими учасниками стали привертати увагу дослідників. Робота Ф. Келі [2] була, по суті, першою комплексною математичною моделлю, яка продемонструвала принцип розподілу пакетів у системі, що працює на основі оптимізації функцій корисності користувачів. Застосування теорії опуклого програмування дозволило сформулювати необхідні умови існування розв'язку – точки рівноважного розподілу ресурсів мережі. Опишемо цей результат. Мережа в даній моделі представляється набором ланок, кожна з яких описується максимальною пропускною здатністю $c_l$, $l = 1,...,M$. Ланки мережі спільно використовуються джерелами, кожне з яких характеризується швидкістю передачі пакетів $x_i$, $i = 1,...,P$. З кожним джерелом (користувачем) пов'язана функція корисності $U_i(x_i)$ та фіксований однозначний маршрут – множина ланок, яка використовується потоком пакетів даного користувача. Нехай $r(i, j)$ – функція, яка дорівнює одиниці якщо джерело $i$ використовує $j$-ту ланку, тоді позначимо $R$ – матрицю маршрутизації:

$$R = \left\{ r_{ij} = r(i, j) \right\}_{\substack{i=1,...,P \\ j=1,...,M}} .$$

Тоді задача рівноважного розподілу ресурсів між користувачами може бути сформульована у вигляді нелінійної програми:

$$\max_{\bar{x} \geq 0} \sum_i U_i(x_i), \; R\bar{x} \leq \bar{c} . \qquad (4)$$

Доведено, що якщо $U_i(\cdot)$ – зростаючі строго опуклі вниз функції, то існує єдиний розв'язок задачі (4).

Підхід Ф. Келі дозволив знайти умови існування точки рівноваги мережі з багатьма конкуруючими користувачами, що мало значний вплив на подальший розвиток даного напрямку.

До основних недоліків даної моделі слід віднести:

1) статичний підхід до аналізу мережі – розглядається тільки точка рівноваги;
2) не врахована робота вузлів мережі – наявність затримок, буферизація, різні алгоритми обробки вхідних пакетів;
3) відсутнє моделювання CC алгоритмів.

**Узагальнені моделі розподілу на основі теорії нелінійного програмування.** У роботі Ф. Келі було встановлено зв'язок між прямою і двоїстою задачами та функціями корисності користувачів і вартістю використання мережі. Була доведена стійкість алгоритмів для невеликих збурень за умов відсутності затримок.

Пізніше ряд авторів [3–5] досліджували різні узагальнення цього підходу, та розвинули загальну теорію розподіленого керування, що спричинило до появи нових схем та алгоритмів. Зокрема, Мо і Варланд [3] запропонували алгоритм на основі затримок, який забезпечує справедливий розподіл ресурсів між користувачами (у сенсі пропорційної справедливості). Пізніше ці ідеї були покладені у основу алгоритму TCP FAST [6].

Каніар і Срікант [7] дослідили схему зворотного зв'язку від мережі до користувача. Запропонована ними ідея помічати а не відкидати пакети отримала назву ECN (explicit congestion notification) і вважається перспективним напрямком розвитку Інтеренту.

**Моделі AIMD динаміки та TCP-подібних з'єднань.** Головним недоліком моделей попередніх параграфів є те, що в них не враховано процеси керування перевантаження – те, що складає найважливіший елемент сучасного Інтернету. Керування перевантаженнями – основа TCP алгоритму представляє реалізацію простої ідеї:

Кожен користувач має збільшувати свою швидкість за умови успішного проходження його пакетів і зменшувати у разі виникнення перевантаження мережі.

Наступним кроком до побудови аналітичної моделі роботи TCP мереж є узагальнена модель AIMD динаміки. AIMD (Additive increase, multiplicative decrease) – адитивне збільшення, мультиплікативне зменшення, означає, що джерело нарощує свою швидкість додаючи певну константу протягом кожного періоду часу, коли не було перевантаження, та зменшуючи її у певну визначену кількість раз протягом кожного моменту часу, коли перевантаження виникало. Суттєво новим в даній моделі є поява механізмів керування на основі вікна (congestion window). Вікно – це кількість пакетів, яку джерело може надіслати в мережу без підтвердження. За допомогою вікна обмежується максимальний обсяг пакетів користувача, що може перебувати в мережі. Підтвердження – це спеціальне повідомлення (ACK пакет), яке містить унікальну для з'єднання послідовність ідентифікації наступного пакета, що очікується адресатом. Ідентифікатори призначаються пакетам упорядковано, тому механізм зворотних підтверджень є одночасно способом підтвердження доставки пакета (якщо очікується новий пакет) і способом виявлення втрат (якщо очікується вже надісланий пакет). Механізм ACK пакетів дозволяє адаптувати швидкість передачі відповідно до змін стану мережі – при уповільненні швидкості надходження ACK повідомлень уповільнюється швидкість джерела, хоча вікно при цьому може залишатись незмінним. З механізмом зворотних підтверджень зв'язане поняття RTT – round trip time – час повного повернення. За визначенням RTT – це час необхідний на проходження пакету до адресата та повернення підтвердження про доставку.

Позначимо $w_i(t)$ розмір вікна $i$-го джерела в момент часу $t$, $\tau_i$ – RTT стану рівноваги, яке вважається сталим. Тоді за визначенням $x_i\left(t\right)=\dfrac{w_i\left(t\right)}{\tau_i}$ – швидкість передачі даних у момент часу $t$. При цьому $x_i(t)$ усереднюється на проміжках часу $\tau_i$, тобто динаміка з меншим масштабом не береться до уваги у цій моделі. Нехай $p_l(t)$ – ймовірність маркування пакета у ланці $l$ мережі у момент часу $t$ (в рамках ECN схеми пакети не відкидаються а помічаються спеціальним маркером, що значно спрощує процес виявлення втрат пакетів). Ключовим припущенням в цій моделі [7] є припущення про адитивність маркування:

$$s_i\left(t\right)=\sum_l r_{li}p_l\left(t\right),$$

що є природнім якщо $p_l\left(t\right)$ мале. Якщо ж припущення порушується, то модель може застосовуватися, але вигляд функцій корисності ускладниться. Опишемо поведінку окремого з'єднання з TCP Reno подібною поведінкою (TCP Reno включає AIMD динаміку джерела, керування перевантаженням та зворотній зв'язок на основі втрат). Нехай в момент часу $t$ швидкість передачі пакетів дорівнює $x_i\left(t\right)$. Вважаючи, що підтвердження повертаються з такою ж швидкістю (що вірно в точці рівноваги мережі) користувач отримує $\left(1-s_i\left(t\right)\right)x_i\left(t\right)$ позитивних підтверджень. Кожне позитивне підтвердження збільшує вікно на $\dfrac{1}{w_i\left(t\right)}$. Також джерело отримує (усереднено) $q_i\left(t\right)x_i\left(t\right)$ помічених або негативних підтверджень, кожне з яких зменшує вікно удвічі. Отже динаміка швидкості джерела може бути представлена у вигляді

$$\dot{x}_i\left(t\right)=\frac{1-s_i\left(t\right)}{\tau_i^2}-\frac{1}{2}s_i\left(t\right)x_i^2\left(t\right). \quad (5)$$

Інший варіант моделі (5) для випадку коли вікно детерміністично збільшується на 1 за кожний RTT описано у роботах [8]

$$\dot{x}_i\left(t\right)=\frac{1}{\tau_i^2}-\frac{1}{2}s_i\left(t\right)x_i^2\left(t\right).$$

Ще одна варіація (5) використовується якщо ми припускаємо, що вікно зменшується удвічі за кожний період RTT, який містить хоча б одне негативне підтвердження. Це змінює мультиплікативну частину:

$$\dot{x}_i\left(t\right)=\frac{1-s_i(t)}{\tau_i^2}-\frac{1}{2\tau_i}s_i\left(t\right)x_i\left(t\right).$$

Дана модель досліджувалась у роботах [8, 9].

**Моделі з урахуванням AQM алгоритмів.** Можливості керування завантаженістю на рівні джерела обмежені. Система керування перевантаженнями, реалізована тільки на стороні клієнта не може бути ефективною, оскільки:

- протокол роботи у мережі Інтернет не є обов'язковим стандартом. Навіть у рамках стандарту TCP різні реалізації ведуть себе по різному при виникненні втрат. Інші типи протоколів можуть взагалі не зменшувати свою швидкість при перевантаженні (як, наприклад, CBR мережі);
- деякі протоколи більш «агресивні» і захоплюють більші потужності каналів зв'язку, якщо їх вчасно не обмежити. Інформація для таких обмежень наявні тільки на маршрутизаторі, який опрацьовує всі потоки даних;
- маршрутизатор може стати ціллю атаки на відмову, або бути перевантажений великими потоками трафіка, тому йому потрібно мати власну систему захисту від перевантажень.

Найбільш поширеними механізмами AQM є DropTail та RED. DropTail історично був першим і природнім рішенням. Згідно з ним пакети, що надійшли поміщуються у загальну чергу і опрацьовуються згідно з порядком надходження. Якщо черга переповнюється, пакети відкидаються. Особливістю потокової моделі DropTail є негладкість правої частини відповідного диференційного рівняння:

$$\dot{x}_i(t) = \frac{1 - s_i(t)}{\tau_i^2} - \frac{1}{2} s_i(t) x_i^2(t),$$

$$\dot{q}_l(t) = \begin{cases} \left[\sum_i r_{li} x_i(t) - u_l\right]_- & q_l(t) = q_l^{\max} \\ \left[\sum_i r_{li} x_i(t) - u_l\right] & 0 < q_l(t) < q_l^{\max} \\ \left[\sum_i r_{li} x_i(t) - u_l\right]_+ & q_l(t) = 0 \end{cases}. \quad (6)$$

Модель (6) для узагальненої поведінки $x_i(t)$, яке визначається цільовою функцією $J_i(x)$ проаналізовано в роботах [10, 11].

Довгий час DropTail був єдиним AQM алгоритмом, однак в його поведінці (особливо у комбінації з TCP) було виявлено два суттєвих недоліки:

**залипання.** Один або декілька потоків даних займають чергу і не дозволяють іншим з'єднанням отримати доступ;

**переповнення.** Якщо черги майже повністю заповнені і відбувся стрибок трафіка, то відбудеться втрата багатьох пакетів. У результаті відразу багато з'єднань обмежать свою швидкість і канал буде недовантажено. Оскільки класичний TCP підвищує свою швидкість надсилання аж до виникнення втрат, то черги знову переповняться і ситуація переповнення буде повторюватися періодично.

Беззаперечним плюсом DropTail є простота його реалізації.

RED – це схема раннього ймовірнісного відкидання пакетів. Алгоритм RED (ймовірнісне раннє виявлення) використовується для згладжування піків трафіку і більш рівномірного обслуговування потоків користувачів. Уперше цей алгоритм було запропоновано в роботі С. Флойда і В. Джекобсона [12] як метод, що дозволяє «підтримувати середню завантаженість черг на низькому рівні але допускає тимчасові перепади трафіку». Для цього застосовується ймовірнісне відкидання пакетів. Ймовірність відкидання залежить від середнього значення черги (яке запам'ятовується за певний період часу). Рішення про відкидання пакетів приймається наступним чином:

- якщо розмір черги менше за $q_{\min}$, то пакети не відкидаються;
- якщо розмір черги більше за $q_{\min}$ і менше за $q_{\max}$, то ймовірність відкидання пропорційна до розміру черги з заданим коефіцієнтом;
- якщо розмір черги перевищує $q_{\max}$, то всі вхідні пакети відкидаються.

**Точне моделювання TCP динаміки.** Створення аналітичної моделі, яка б давала змогу прогнозувати поведінку TCP з'єднань у мережі надзвичайно важливе для дослідження динаміки сучасних мереж. На перший план тут виходять стійкість і справедливість розподілу ресурсів. Проблема полягає у постійних стохастичних збуреннях, які впливають на затримки і втрати пакетів користувачів. Тому необхідно знати чи є алгоритм керування мережею стійким, тобто таким, що при появі збурень повертає систему до стану рівноваги. Ще одним джерелом нестабільності є поведінки користувачів – їх кількість і запити можуть непередбачувано змінюватися. Ці зміни утворюють новий рівноважний розподіл ресурсів мережі і алгоритми керування повинні достатньо швидко і плавно переводити систему у нову точку. Важливою характеристикою розподілу ресурсів у мережі є справедливість. Питання стійкості справедливого розподілу і справедливості стійкої рівноваги мережі складають другу проблему, якою наразі посилено займаються дослідники.

Умови існування і єдиності точки рівноваги (для ідеалізованої математичної моделі мережі) були знайдені Ф. Келі [2]. Пізніше для ідеалізованих схем TCP (без затримок, з рівномірно розподіленими підтвердженнями) було застосовано методи аналізу стійкості за Ляпуновим теорії керування (Лоу 1999, Канніур Срікант 2002, Алтман 1998, Вен Арак 2004). В цих роботах не враховувалась наявність затримок, що критично важливо для стійкості реальних систем. Локальна стійкість схеми TCP Reno з AQM RED з урахуванням затримок досліджена у роботах [13, 14]. Аналіз показав, що даний протокол стає нестійкий при зростанні затримок та (що було досить несподівано) потужностей мережі. Цей результат спричинив до появи нового напрямку досліджень – розробки протоколів, локально стійких у високопродуктивних мережах.

Інший напрямок – визначення умов глобальної стійкості за умов наявності затримок пов'язаний з функціоналами Красовського – Ляпунова [15, 16].

У 2007 році Вей [17] вказав на суттєвий недолік існуючих потокових моделей – відсутність моделювання ефектів пакетного рівня. Вся ідея потокового моделювання полягає у заміні потоку пакетів на непевну «субстанцію», що струмує через канали зв'язку. Однак, алгоритм TCP суттєво використовує два пакетно-орієнтовані механізми – керування на основі зворотних підтверджень та керування на основі вікна, тому теоретичні моделі не могли передбачити і проаналізувати ефекти, що виникали у реальних системах.

У роботах [18 – 20] здійснена спроба побудувати адекватну модель TCP з'єднання. При потоковому моделюванні мереж центром уваги є швидкість передачі даних джерелами. Причина тут полягає у тому, що стан кожного вузла мережі визначається різницею між вхідним трафіком та потужністю цього вузла. Знаючи швидкість надходження пакетів у мережу можна зробити висновки про стійкість, черги та завантаженість системи. Основна проблема з TCP алгоритмами полягає в нетривіальній залежності майбутньої швидкості джерела від поточної. На додачу, в TCP головною внутрішньою змінною і керуючим параметром є розмір вікна, який неявно (в залежності від швидкості повернення ACK пакетів) визначає швидкість передачі даних. Моделювання цих двох механізмів на сьогодні активно розвивається дослідниками.

Позначимо $f(t)$ загальна кількість пакетів, що надіслані джерелом у мережу але ще не підтверджені. Ці пакети або відповідні їм ACK пакети можуть перебувати у чергах або лініях зв'язку. Якщо припустити, що $f(t)=const$, то зрозуміло, що за інтервал часу $(t,t+\tau(t))$ джерело має надіслати у мережу рівно $f(t)$ пакетів; тут $\tau(t)$ – час необхідний щоб останній пакет, надісланий у мережу до моменту часу $t$ включно був підтверджений. В реальних системах звісно $f(t)$ змінюється протягом часу. Важливим однак є значення $f(\cdot)$ на кінці інтервалу. Загальна кількість пакетів, яку джерело має надіслати у систему для збереження балансу дорівнює $f\left(t+\tau\left(t\right)\right)$. Нехай швидкість джерела описується функцією $\lambda\left(\cdot\right)$, тоді за визначенням функції $f\left(\cdot\right)$ для кожного моменту часу $t$ виконується співвідношення

$$\int_{t}^{t+\tau(t)} \lambda\left(\theta\right)d\theta = f\left(t+\tau\left(t\right)\right). \qquad (7)$$

Складність аналізу (7) полягає у невідомій природі затримки $\tau\left(t\right)$. Ця затримка залежить від стану системи у момент часу $t$ та дій гравців протягом усього інтервалу $\left[t, t+\tau\left(t\right)\right)$. У зв'язку з цим в сучасних дослідженнях застосовується різні наближення. Найбільш розповсюджена лінійна форма $\tau\left(t\right) = d + \frac{q\left(t\right)}{u}$, де $d$ – постійна затримка, пов'язана з каналами зв'язку, $\frac{q\left(t\right)}{u}$ – затримка, пов'язана з заповненням вузького місця з чергою $q\left(t\right)$ і пропускною спроможністю $u$ (вважається фіксованою).

У роботі К. Якобсона [19] досліджена модель DAE, яка для випадку лінійних затримок і мережі з одним вузьким місцем описується наступними співвідношеннями

$$\dot{q}\left(t\right) = \lambda\left(t\right) + \lambda_c\left(t\right) - u\,,$$

$$\int_{t}^{t+\tau(t)} \lambda\left(\theta\right)d\theta = f\left(t+\tau\left(t\right)\right),$$

$$\tau\left(t\right) = d + \frac{q\left(t\right)}{u},\; q\left(t\right) \ge 0\,,\; f\left(t\right) \ge 0\,, \qquad (8)$$

де $\lambda_c$ – швидкість фонового трафіку. В роботі К. Якобсона встановлені умови стійкості, побудована модель DAE протоколу FAST, встановлена стійкість для системи багатьох користувачів з різнорідними RTT. Проведене експериментальне підтвердження з використанням середовища імітаційного моделювання NS-2.

Дещо інша модель запропонована у роботі Н. Мьоллера [20]. Усереднюючи за інтервалом $\left[t, t+\tau\left(t\right)\right]$ можемо вважати, що у стані рівноваги $\lambda\left(t\right) = \frac{f\left(t\right)}{\tau\left(t\right)}$. Слід зазначити, що в реальних системах вікно алгоритму TCP $w\left(t\right)$ буде відрізнятися від $f\left(t\right)$, однак в рамках даної моделі вони вважаються рівними. Отже динаміка черги вузького місця системи описується рівнянням

$$\dot{q}\left(t\right) = \frac{w\left(t\right)}{d + \frac{q\left(t\right)}{u}}.$$

Це – так звана *інтегруюча модель* (оскільки буфер вузького місця виступає у ролі інтегратора трафіку). З іншого боку, незалежно від внутрішньої роботи мережі вона повертає підтвердження зі швидкістю вузького місця системи. Використовуючи дану властивість побудована *статична модель* керованої мережі:

$$q\left(t\right) = \frac{w\left(t\right)}{\gamma} - u\tau\,,$$

$$\lambda\left(t\right) = \gamma u + \frac{\dot{w}\left(t\right)}{\gamma},$$

де τ – RTT, яке вважається сталим, $\gamma = 1 - \frac{\lambda_c}{u}$ – доступна потужність системи. Інтегруюча і статична моделі суперечать одна одній, кожна з них описує певну частину TCP механізму. Розвитком стала побудова об'єднаної моделі (Н. Мьоллер). Нехай $w\left(t\right) = \int_{t_0}^{t_1} \lambda\left(s\right)ds$ – розмір вікна. Вважаючи , що функції $w\left(t\right)$, $\lambda\left(t\right)$ задовольняють умовам теореми про середнє можна записати:

$$w\left(t_0\right) = \int_{t_0}^{t_1} \left(\lambda\left(s\right) - \dot{w}\left(s\right)\right)ds = \left(t_1 - t_0\right)\left(\lambda\left(t\right) - \dot{w}\left(t\right)\right),$$

$$\lambda(t) = \frac{w(t_0)}{\tau + \frac{q(t_0)}{u}} + \dot{w}(t).$$

Розглядаючи систему без затримок модель поведінки джерела записується

$$\dot{q}(t) = \frac{w(t)}{\tau + \frac{q(t)}{u}} + \dot{w}(t) + \lambda_c(t) - \gamma u.$$

## 2. Узагальнена модель оптимального керування потоками даних у мережі Інтернет

Опишемо динамічну потокову модель мережі. При побудові аналітичної моделі будемо вважати, що:

- процес передачі пакетів у мережі апроксимується неперервними потоками;
- кожен користувач пов'язаний з фіксованим з'єднанням та маршрутом передачі даних, які не змінюються протягом усього часу розгляду моделі;
- втрата окремого пакету вважається незначною подією, якою можна знехтувати. В рамках моделі розглядаються поведінка мережі «в цілому»;
- кожна ланка мережі однозначно пов'язана з чергою. Черги функціонують за принципом FIFO. При переповненні черги нові пакети втрачаються;
- пакети підтвердження і інформація про втрату пакетів доставляються миттєво і не завантажують мережу (нульові інформаційні затримки);
- з кожним користувачем пов'язана його функція корисності $J_i(\cdot)$, яку він намагається мінімізувати своїми діями.

Будемо вважати, що мережа складається з вузлів і з'єднуючих ланок. На кожному вузлі розташовані одна або більше черг з якими пов'язані обслуговуючі ресурси. Позначимо $I = \{1, 2, ..., N\}$ множину індексів користувачів. Користувач надсилає у мережу потік своїх пакетів, керуючи швидкістю передачі – функцією $\lambda_i(t)$, $i \in I$. Введемо множину індексів черг $J = \{1, 2, ..., M\}$ та відповідний вектор черг $q_j(t)$. Побудуємо матрицю $A$ наступним чином – якщо пакети $i$-го користувача потрапляють на вхід $j$-ї черги, то відповідний елемент матриці $a_{ij} = 1$, в іншому випадку $a_{ij} = 0$. Кожна черга зв'язана з обчислювальним ресурсом $u_j(t)$, $j \in J$, який обслуговує потік пакетів. Оскільки потік пакетів складається з потоків різних користувачів, то, взагалі кажучи, кожна функція $u_j(t) = u_j^1(t) + ... + u_j^n(t)$ (звичайно деякі з цих компонентів можуть бути рівні нулю). Питання визначення стратегії розподілу ресурсів $u(t)$ буде розглядатися пізніше.

Кожен вузол може містити одну або більше черг. Позначимо $K = \{1, 2, ..., p\}$ множину індексів вузлів. Введемо функцію відповідності $s(j) \in K$, $j \in J$, яка визначає належність черги $j$ до вузла $k$. Матриця відповідності $C$ визначається наступним чином:

$$c_{ij} = \begin{cases} 1 & s(j) = k \\ 0 & s(j) \neq k \end{cases}.$$

Матриця $C$ описує структуру розташування черг на вузлах. Після обробки пакети залишають вузол мережі і можуть або залишити її межі або потрапити на інший вузол. Шлях переходу пакету з вузла на вузол називається маршрутом і задається матрицею $R$:

$$r_{ij} = \begin{cases} 1 & r(j) = i \\ 0 & r(j) \neq i \end{cases},$$

де $r(j) \in J$ – функція, що задає чергу у яку потрапляють пакети з $j$-ої черги.

При переповненні черги пакети, що надходять втрачаються. Цей процес описується функціями втрат

$$l_j(\lambda, q) = \begin{cases} 0 & q < q^{\max} \\ \min\{-[A\lambda + Bu]_j, 0\} & q \geq q^{\max} \end{cases},$$

$$j \in J.$$

Введемо також функцію сумарних втрат $i$-го користувача по всім чергам $\Lambda_i(t)$. Після закінчення обслуговування пакетів користувача сервер надсилає йому підтвердження про успішну обробку. Позначимо $v_i(t)$ функції, що описують доставку підтверджень.

Для кожного користувача заданий цільовий функціонал $J_i(\cdot)$, який той намагається мінімізувати. Зокрема, таким функціоналом може бути час закінчення обробки певного обсягу пакетів

$$J_i(\lambda(t)) = \min\left\{ t \ge 0 : \int_0^t v_i(\tau)d\tau \ge \lambda_i^{\text{int}} \right\}.$$

Таким чином, динаміка роботи мережі описується наступною системою звичайних диференційних рівнянь:

$$\frac{d\overline{q}(t)}{dt} = A\overline{\lambda}(t) + B\overline{u}(t) + l\left(\overline{\lambda}(t), \overline{q}(t)\right). \quad (9)$$

На систему (1) накладені наступні обмеження, що випливають з природи реальних мереж і мають ключове значення для розв'язання задач ігрового керування.

- пропускні здатності ланок мережі є обмеженими, отже обсяг сумарного потоку кожної черги не може перевищувати задану величину $\left[A\lambda + Bu\right]_j \le d_j$, $j \in J$;
- обсяги черг більші за нуль та обмежені за величиною – $0 \le q_j(t) \le q_j^{\max}$, $j \in J$;
- керування $u(t)$ будемо вважати кусково гладким зі значеннями з опуклого компакту відповідного простору;
- функції $\lambda_i(t)$ – невід'ємні кусково неперервні функції, обмежені за величиною $0 \le \lambda_i(t) \le \lambda_i^{\max}$, $j \in J$.

Якщо користувач працює в рамках TCP мережі (як буде вважатися у більшій частині роботи), то існує додаткове обмеження, пов'язане з керуванням на основі вікна доступу. Позначимо $w(t) \ge 0$ – величину вікна у момент часу $t$, тоді для кожного моменту часу $T \ge 0$ має виконуватися нерівність:

$$\int_0^T \left[\lambda_i(t) - v_i(t) + \Lambda_i(t)\right]dt \le w(T). \quad (10)$$

Нерівність (10) означає, що кількість пакетів, які можуть потрапити у мережу на момент часу $T$ не може бути більша за сумарну кількість підтверджених і втрачених пакетів плюс розмір вікна в цей момент часу. Динаміка з обмеженням (10) досить складна для дослідження, тому в даній роботі ми обмежимося простою топологією мережі (яка тим не менше дозволить зрозуміти важливі властивості взаємодії потоків даних користувачів).

**Проста динаміка.** Розглянемо систему з одним сервером та двома користувачами, яка описується системою диференціальних рівнянь:

$$\dot{q}(t) = \lambda_1(t) + \lambda_2(t) - u(t) - l(t),$$

$$q(t) \in \left[0, q^{\max}\right],\ u(t) \in \left[0, u^{\max}\right],$$

$$\lambda_i(t) \in \left[0, \lambda_i^{\max}\right],\ i = 1,2. \quad (11)$$

За аналогією з теорією диференціальних ігор такий тип динаміки можна назвати *простою динамікою*.

Введемо допоміжні змінні $q_i(t)$, що описують обсяг черги, який займають пакети $i$-го користувача у момент часу $t$ та $u_i(t)$ – частина ресурсів сервера, що використовується для обслуговування пакетів $i$-го користувача у момент часу $t$. Будемо вважати, що час обертання пакетів (затримка) описується функцією $T(t) = \dfrac{q(t)}{u^{\max}}$. Іншими словами це час, необхідний на повне звільнення поточної черги від пакетів, що потрапили туди до часу $t$. Визначимо функції обробки та відкидання пакетів. Якщо швидкості і розміри пакетів достатньо близькі можна вважати, що пакети потрапляють у чергу пропорційно швидкості потоків:

$$u_i(t)=\frac{u^{\max}\lambda_i\left(t-T(t)\right)}{\sum\limits_{i=1,2}\lambda_i\left(t-T(t)\right)},$$

$$l_i(t)=\begin{cases}\max\left\{0,\lambda_i(t)-u_i(t)\right\}, & q(t)=q^{\max}\\ 0, & q(t)<q^{\max}\end{cases}. \quad (12)$$

При цьому якщо $\lambda_1(t)=\lambda_2(t)=0$ покладемо $u_i(t)=0$.

Задача передачі фіксованого обсягу даних за мінімальний час має вигляд

$$J_i\left(\lambda(t)\right)=\min\left\{t\ge 0:\int\limits_0^t u_i(\tau)d\tau\ge\lambda_i^{\text{int}}\right\}\to\min.$$

Задача (11) з функціями (12) може бути записана у вигляді

$$\dot{\bar{q}}=f\left(\bar{q}(t),\bar{\lambda}(t)\right),\ q(t_0)=0. \quad (13)$$

Функція $f(t)$ визначена для $t\in[0,+\infty]$, обмежена та інтегрована. Існує розв'язок рівняння $q(t)$ для будь-яких допустимих функцій $\lambda_1(t)$, $\lambda_2(t)$.

Покажемо тепер, що в задачі оптимального керування існує розв'язок. Побудуємо множину досяжності першого гравця

$$U_1(t,t_0)=\left\{u:u=\int\limits_{t_0}^t u_1(\tau)d\tau\right\},$$

де інтеграл береться по всім вимірним допустимим функціям. Зафіксуємо стратегію $\lambda_2(\cdot)$ і розглянемо можливі $u_1(t)$ для різних значень $\lambda_1\in\left[0,\lambda_1^{\max}\right]$.

Оскільки

$$0\le\frac{u^{\max}\lambda_1}{\lambda_1+\lambda_2(t)}\le\frac{u^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2(t)},$$

то $U_1(t,t_0)=\left[0,\int\limits_{t_0}^t\frac{u^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2(\tau)}d\tau\right]$, де

$$\int\limits_{t_0}^t\frac{u^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2(\tau)}d\tau\ge(t-t_0)\frac{u^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}}.$$

Отже існує $T^*$, таке, що $\lambda_1^{\text{int}}\in U_1(t,t_0)$. З компактності $U_1(t,t_0)$ випливає існування мінімального часу.

**Твердження 1.** Для $\lambda_2(t)=const$ мінімальний час завершення першого користувача дорівнює

$$T_1^{\min}=\frac{\lambda^{\text{int}}\left(\lambda_1^{\max}+\lambda_2\right)}{\lambda_1^{\max}u^{\max}}.$$

Для довільної $\lambda_2(t)$ мінімальний час можна оцінити наступною формулою:

$$T_1^{\min}\le\frac{\lambda^{\text{int}}\left(\lambda_1^{\max}+\lambda_2\right)}{\lambda_1^{\max}u^{\max}}.$$

**Доведення.** Для будь-якої $\lambda_2(\cdot)$ стратегія $\lambda_1(t)\equiv\lambda_1^{\max}$ дає не більший час завантаження порівняно з іншими $\lambda_1(\cdot)$: $T_1^{\min}\left(\lambda_1^{\max}\right)\le T_1^{\min}\left(\lambda_1(\cdot)\right)$ для всіх $\lambda_1(\cdot)$.

Далі, при $\lambda_2(t)=\lambda_2^{\max}$ значення часу є максимальним.

**Наслідок.** Пара керувань $\lambda_i=\lambda_i^{\max}$ є точкою рівноваги за Нешем у грі завантаження.

Пара стратегій $\left(\lambda_1^{\max},\lambda_2^{\max}\right)$ може бути неефективною точкою рівноваги через високі рівні втрат.

Достатні умови виникнення втрат.

**Твердження 2.** Якщо виконується одна з наступних умов то для системи (13) $l_1(t)\equiv 0$ для $t\in\left[t_0,+\infty\right]$:

1) $\lambda_1^{\max}+\lambda_2(t)\le u^{\max}$;

2) $\dfrac{q^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}-u^{\max}} \ge \lambda^{\text{int}}\left(\lambda_1^{\max}+\lambda_2^{\max}\right)$.

**Доведення.** Якщо виконується умова 1), то $\dot{q}(t)=0$ і $l_1(t)=0$. Виконання умови 2) означає, що за час заповнення черги $t^*=\dfrac{q^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}-u^{\max}}$ загальна кількість пакетів першого користувача які потрапили у систему $t^*\dfrac{\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}}$ більша за $\lambda^{\text{int}}$.

Якщо умови 1), 2) не виконуються в системі можуть виникнути втрати. Оцінимо втрати у точці рівноваги $\left(\lambda_1^{\max},\lambda_2^{\max}\right)$. Час заповнення черги $t^*$ визначено вище, загальний час завершення завантаження дорівнює $T^*=\dfrac{\lambda^{\text{int}}\left(\lambda_1^{\max}+\lambda_2^{\max}\right)}{\lambda_1^{\max}u^{\max}}$.

Загальні втрати

$$\int_{t^*}^{T^*} l_1(t)\,dt=\int_{t^*}^{T^*}\left[\lambda_1^{\max}-u_1\right]dt=$$

$$=\int_{t^*}^{T^*}\left[\lambda_1^{\max}-\frac{u_1^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}}\right]dt=$$

$$=\left(T^*-t^*\right)\left[\lambda_1^{\max}-\frac{u_1^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}}\right]=$$

$$=\frac{\lambda^{\text{int}}}{u^{\max}}\left(\lambda_1^{\max}+\lambda_2^{\max}-u^{\max}\right)-\frac{q^{\max}\lambda_1^{\max}}{\lambda_1^{\max}+\lambda_2^{\max}}.$$

**Динаміка з фіксованим вікном доступу.** Розглянемо динаміку системи (13) з одним користувачем та додатковим обмеженням у вигляді вікна доступу:

$$\dot{q}(t)=\lambda_1(t)-u(t)-l_1(t),$$

$$q(t)\in[0,q^{\max}],\ \lambda_1(t)\in\left[0,\lambda_1^{\max}\right],$$

$$\int_0^T\left[\lambda_1(t)-u(t)-l_1(t)\right]dt\le w,$$

$$J\left(\lambda(\cdot)\right)=$$

$$=\min\left\{t\ge 0:\int_0^t u_i(\tau)d\tau\ge\lambda_i^{\text{int}}\right\}\to\min.\quad(14)$$

Оскільки для системи з одним користувачем

$$u(t)=\begin{cases}0, & \lambda_1(t)=0\\ u^{\max}, & \lambda_1(t)>0\end{cases},$$

то $J\left(\lambda(\cdot)\right)$ мінімальний якщо $\lambda_1(t)$ максимальна.

Визначимо функцію $\lambda^*(t)$ наступним чином:

$$\lambda^*(t)=\begin{cases}\lambda^{\max}, & t\in\left[0,\dfrac{w}{\lambda^{\max}-u^{\max}}\right]\\ u^{\max}, & t\in\left(\dfrac{w}{\lambda^{\max}-u^{\max}},\dfrac{\lambda^{\text{int}}}{u^{\max}}\right]\end{cases},$$

якщо $\dfrac{w}{\lambda^{\max}-u^{\max}}\le\dfrac{\lambda^{\text{int}}}{u^{\max}}$, то $\lambda^*(t)=\lambda^{\max}$.

**Твердження 3.** Задача (14) має мінімум. Мінімум досягається, зокрема, на функції $\lambda^*(t)$.

**Доведення.** Існування мінімуму випливає з загальних теорем існування розв’язку диференціального рівняння та того факту, що множина $\int_0^t u_i(\tau)d\tau$ є опуклим компактом, який неперервно (у метриці Хаусдорфа) залежить від $t$ та зростає у сенсі вкладення ($\int_0^t u_i(\tau)d\tau\subset\int_0^T u_i(\tau)d\tau$, якщо $t\le T$). Розглянемо множину точок, в яких втрати ненульові. Позначимо момент закінчення завантаження $T^*$, тобто

$\int_0^{T^*} u_i(\tau)d\tau = \lambda_i^{\text{int}}$ і такий момент часу мінімальний, $T = \left[0, T^*\right]$, $T_l = \left\{t \geq 0 : l_1(t) > 0\right\}$. Функція втрат визначається наступним чином:

$$l_1(t) = \begin{cases} \max\left\{0, \lambda_1(t) - u_1(t)\right\}, & q(t) = q^{\max} \\ 0, & q(t) < q^{\max} \end{cases}.$$

Оскільки розв'язок $q(t)$ є абсолютно неперервна функція, то множина $T_q = \left\{t \geq 0 : q(t) = q^{\max}\right\}$ є об'єднання замкнутих множин в $R^1$ (тобто об'єднання точок та відрізків). Оскільки функція $\lambda_1(\cdot)$ – вимірна, то і $l_1(\cdot)$ також, тому $T_l \subset T_q$ – вимірна множина. Введемо функцію $\hat{\lambda}_1(t)$ за наступною формулою:

$$\hat{\lambda}_1(t) = \begin{cases} \lambda_1(t), & t \notin T_l \\ u^{\max}, & t \in T_l \end{cases}.$$

Зрозуміло, що $\hat{\lambda}_1(t)$ допустима функція (вимірна та задовольняє умовам системи (14)), і $\dot{q} = \hat{\lambda}_1(t) - u(t) = \lambda_1(t) - u(t) - l_1(t)$. Тому час закінчення завантаження співпадає – $T^* = \hat{T}$. Отже, якщо мінімум досягається для функції $\lambda_1(t)$, то і для $\hat{\lambda}_1(t)$ також. Для системи (14) мінімальний час закінчення завантаження дорівнює $\dfrac{\lambda^{\text{int}}}{u^{\max}}$. Пряма перевірка показує, що він досягається на функції $\lambda^*(t)$.

**Гра двох учасників з фіксованим вікном доступу**.

Нехай динаміка системи з одним сервером та двома користувачами, описується системою диференціальних рівнянь:

$$\dot{q}(t) = \lambda_1(t) + \lambda_2(t) - u(t) - l(t),$$

$$q(t) \in [0, q^{\max}],\ u(t) \in [0, u^{\max}],$$

$$\int_0^T \left[\lambda_i(t) - u_i(t) - l_i(t)\right] dt \leq w_i,$$

$$\lambda_i(t) \in \left[0, \lambda_i^{\max}\right],$$

$$J_i(\lambda_i(\cdot)) = \min\left\{t \geq 0 : \int_0^t u_i(\tau)d\tau \geq \lambda_i^{\text{int}}\right\} \to \min,$$

$$i = 1,2. \qquad (15)$$

Функції $u_i(t)$, $l_i(t)$ визначені формулами (12). Введемо допоміжні змінні $q_i(t)$, $l_i(t)$.

**Твердження 4.** Якщо виконується одна з наступних умов то для системи (15) $l(t) \equiv 0$ для $t \in [t_0, +\infty]$:

1) $\lambda_1^{\max} + \lambda_2^{\max}(t) \leq u^{\max}$;

2) $w_1 + w_2 \leq q^{\max}$;

3) $\dfrac{q^{\max}\lambda_i^{\max}}{\lambda_1^{\max} + \lambda_2^{\max} - u^{\max}} \geq \lambda^{\text{int}}\left(\lambda_1^{\max} + \lambda_2^{\max}\right)$,

$i = 1,2$.

**Доведення.** Доведення для пунктів 1), 3) повторює доведення з твердження 2. Нехай виконується умова 2), тоді сумарна кількість пакетів, що можуть перебувати у черзі $w_1 + w_2$ менше за $q^{\max}$, отже втрати не виникають.

## Висновки

Дана робота присвячена моделям ігрового керування потоками даних у мережах, що використовують TCP алгоритми керування перевантаженнями. В роботі представлено огляд проблеми, підкреслена її актуальність, описані існуючі підходи до вирішення. Для мереж зі змінними протоками даних і наявності конфліктуючих користувачів пропонується новий підхід до аналізу роботи мережі на базі конфліктно-керованих потокових моделей. Побудована модель одного класу мереж з фіксованим вікном доступу та доведене існування оптимального у сенсі швидкодії розв'язку задачі завантаження файлу. Розглянуто гру двох користувачів, що намагаються передавати дані через спільний канал та знайдені умови виникнення втрат.


1. *Meyn S.* Control Techniques for Complex Networks. – Cambridge University Press. – 2007. – 582 p.
2. *Kelly F.P.* Charging and rate control for elastic traffic // European Trans. on Telecommunications. – 1997. – **8**. – P. 33 – 37.
3. *Mo J., Walrand J.* Fair end-to-end window-based congestion control // IEEE/ACM Transactions on Networking. – 2000. – **8**. – P. 556 – 567.
4. *La R.J., Anantharam V.* Charge-sensitive TCP and rate control in the Internet // Proc. IEEE Infocom. – 2000. – P. 1166 – 1175.
5. Low S.H., Lapsley D.E. Optimization flow control: Basic algorithm and convergence // IEEE/ACM Transactions on Networking. – 1999. – **7**, N 6. – P. 861 – 874.
6. *Paganini F., Doyle J.C., Low S.H.* Scalable laws for stable network congestion control // Proc. of IEEE Conference on Decision and Control. – 2001. – 1. – P. 185 – 190.
7. *Kunniyur S. and Srikant R.* "A time-scale decomposition approach to adaptive explicit congestion notification (ECN) marking," // IEEE Transactions on Automatic Control, June 2002. – Vol. 47(6). – P. 882–894.
8. *Low S.H., Srikant R.* A Mathematical Framework for Designing a Low-Loss, Low-Delay Internet // Network and Spatial Economics. – 2004. – **4** (1). – P. 75 – 102.
9. *Low S. Paganini F., Doyle J.* Internet congestion control // IEEE Control Syst. Mag. – 2002. – **22** (1). – P. 28–43.
10. *Alpcan T. and Başar T.* Network Security: A Decision and Game Theoretic Approach. Cambridge University Press, 2010.
11. *Alpcan T. and Başar T.* "A game theoretic analysis of intrusion detection in access control systems," in Proc. of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, December 2004. – P. 1568–1573.
12. *Floyd S. and Jacobson V.* Random early detection gateways for congestion avoidance // IEEE/ACM Transactions on Networking. – 1993. –1(4). – P. 397–413.
13. *Hollot C. V., Misra V., Towsley D., and Gong W.* Analysis and design of controllers for AQM routers supporting TCP flows // IEEE Transactions on Automatic Control. – 2002. – 47(6). – P. 945–959.
14. *Tan L., Zhang W., Peng G., and Chen G.* Stability of TCP/RED systems in AQM routers // IEEE Transactions on Automatic Control. – 2006. – 51(8). – P. 1393–1398.
15. *Deb S. and Srikant R.* Global stability of congestion controllers for the Internet // IEEE Transactions on Automatic Control. – 2003. – 48(6). – P. 1055–1060.
16. *Ying L., Dullerud G. E., and Srikant R.* Global stability of Internet congestion controllers with heterogeneous delays // IEEE/ACM Transactions on Networking. – 2006. – 14(3). – P. 579–591.
17. *Wei. X.* Microscopic behavior of TCP congestion control. PhD thesis, California Institute of Technology. 2007.
18. *Welzl M.* Network Congestion Control: Managing Internet Traffic. Wiley. – 2005. – 263 p.
19. *Jacobsson K.* Dynamic modeling of Internet congestion control. PhD thesis, KTH school 2008 of electrical engeneering.
20. *Möller. N.* Window-based congestion control. PhD thesis, The Royal Institute of Technology, KTH. 2008.


Одержано 23.11.2011

***Про автора:***

*Ігнатенко Олексій Петрович*,
кандидат фізико-математичних наук,
старший науковий співробітник,
докторант.

***Місце роботи автора:***

Інститут програмних систем
НАН України,
03187, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 526 6025.
e-mail: o.ignatenko@gmail.com