УДК 004.7

ПОТОКОВІ МОДЕЛІ МЕРЕЖІ ІНТЕРНЕТ ЗА УМОВ АТАК НА ВІДМОВУ

П.І. Андон, О.П. Ігнатенко

Інститут програмних систем НАН України,
03187, Київ, проспект Академіка Глушкова, 40.
Тел.:+380 (44) 526 3309, факс: 380 (44) 526 6025,
e-mail: ignat@isofts.kiev.ua

Дана робота присвячена потоковому моделюванню процесів керування потоками даних у мережах за умов поглинаючих атак на відмову. Описано алгоритм керування перевантаженням TCP (протокол керування передачею), побудована потокова модель TCP з'єднання. Розглянута задача оптимального завантаження файлу та ігрової взаємодії двох користувачів. Для моделі поглинаючої атаки отримані аналітичні оцінки погіршення якості зв'язку в залежності від обсягу атаки.

This paper deals with fluid controlled models of networks under flooding denial of service attacks. We proposed a new approach to analyzing network work using fluid models and conflict control theory domain. Using analytic model there is found solution for certain class of networks and Nash equilibrium. We consider special case of network activity – denial of service attack and found dependency on download time depending of attack direction and traffic volume. The theoretical results were tested in simulation environment NS-2.

Вступ

Сучасні комп'ютерні мережі пронизують майже всі області 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 мереж 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 призначена для розв'язання окремого класу проблем, представляючи роботу системи в певному розрізі.

Особливий інтерес представляє дослідження процесів керування потоками даних, оскільки розвиток мережі Інтернет 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 Л. Клейнроком на базі теорії масового обслуговування був розроблений перший теоретично обґрунтований протокол (TCP – протокол керування передачею). На сьогодні дослідниками запропоновано більше 50 реалізацій алгоритмів керування потоками даних (TCP алгоритмів). 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 Ф. Келі [1], С. Лоу та Ф. Паганіні [2], Р. Срікантом [3]. 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 С. Лоу [4 – 5] 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 системи. Це – ситуація зловмисного використання, так звана атака на відмову [6]. Такі атаки можуть проявлятися у дуже різноманітні способи, але, так чи інакше, наслідком є значне погіршення якості або повне припинення роботи мережі. 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. 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 врахуванням поведінки зловмисних користувачів та створення теоретично обґрунтованих стратегій протидії їм дозволило б досягти значних успіхів у вирішенні цієї проблеми.

Ефективна робота такої складної і масштабної структури як Інтернет залежить від багатьох факторів. Алгоритми, що забезпечують надійний обмін даними між різними користувачами (протоколи) складаються з

різних рівнів реалізації, кожен з яких призначений для виконання певних задач. Структура рівнів моделі TCP/IP показана на рис. 1.

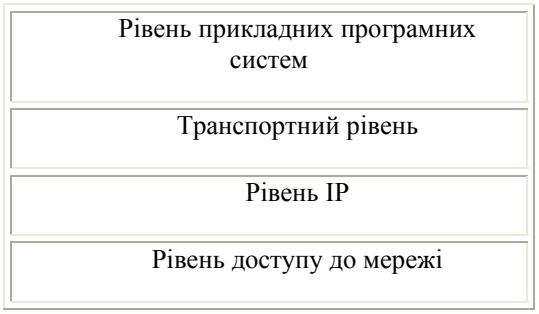

Рис. 1. Стек TCP/IP

Найнижчим рівнем є рівень доступу до мережі. На цьому рівні визначається використання фізичних елементів мережі для передачі IP датаграм (пакетів даних) таких як кабелі, повторювачі та інтерфейсні плати. На сьогодні найбільш поширеною технологією фізичного рівня мереж є Ethernet.

Рівень IP. Даний рівень має ключове значення для роботи мережі Інтернет і призначений для передачі пакетів даних з одного комп'ютера на інший. IP надає єдиний механізм роботи з різнорідним устаткуванням. Всі служби і протоколи вищих рівнів використовують IP як засіб передачі пакетів.

IP виконує наступні функції:

- визначає датаграму – пакет, що є основним елементом процесу передачі даних в мережі;
- визначає схему адресації Інтернету;
- забезпечує передачу даних між рівнем доступу до мережі і транспортним рівнем;
- пересилає датаграми на заданий комп'ютер;
- забезпечує фрагментацію і зборку даних.

IP відповідає за надійну передачу даних за заданою адресою але не надає гарантій щодо отримання вірних даних (дані можуть прийти з помилкою).

Транспортний рівень. Транспортний рівень відповідає за надійну передачу даних з одного вузла на інший. На даному рівні вперше з'являється поняття гарантування якості (за успішного завершення роботу протоколу вузол отримує заплановані дані). TCP є найбільш поширеним представником алгоритмів цього рівня [7]. TCP забезпечує двонаправлений потік пакетів між двома вузлами. При цьому для синхронізації використовуються спеціальні ідентифікатори послідовності пакетів і пакти підтвердження (так звані ACK пакети або повідомлення). За їх допомогою виявляється втрати, зміни у порядку надходження та дублювання пакетів. Ще одним призначенням ACK повідомлень є керування обсягом потоку джерела (механізм ACK clocking – зворотній зв'язок на основі підтверджень), який запобігає перевантаженню вхідної черги одержувача. Інший ключовий механізм TCP – керування на основі вікна доступу запобігає перевантаженню проміжних вузлів мережі.

Гарантований зв'язок, який надає TCP не завжди підходить для потреб користувача. Наприклад, аудіо і відео потоки, програми реального часу більш чутливі до рівня затримок ніж до втрат окремих пакетів. Для них можуть бути більш прийнятними протоколи найкращого можливого зв'язку, зокрема UDP.

Рівень прикладних програмних систем. Існує значна кількість протоколів, що використовують транспортний рівень для організації сесій, служб та допоміжних програм, наприклад:

- DNS – Domain Name System – переводить адресацію мережі (наприклад, IP адреси) у зрозумілі людям назви (сайтів, доменів) і навпаки;
- DHCP – Dynamic Host Configuration Protocol – автоматично призначає Інтернет адреси комп'ютерам та користувачам;
- FTP – File Transfer Protocol – протокол, що використовується для передачі файлів;
- HTTP – HyperText Transfer Protocol – Інтернет протокол для надсилання і прийому веб-сторінок.

Керування перевантаженнями. Одним з найважливіших елементів, який впливає на роботу всієї системи є алгоритм керування перевантаженнями. Алгоритми керування перевантаженням мають регулювати обсяги потоків даних, що генеруються кінцевими вузлами та мінімізувати затримки і втрати. Наявна на сьогодні висока ефективність мережі Інтернет пов'язана, насамперед, з TCP, який наразі є найбільш поширеною і розвиненою схемою керування перевантаженнями.

TCP надає розподілений і децентралізований механізм, згідно до якого кожен користувач має оптимально використовувати спільні ресурси мережі. Оптимально в даному контексті означає, що обсяг потоків даних користувачів має бути максимальним але не допускати перевантаження мережі. За умов наявності великої кількості користувачів, які не мають фізичної можливості узгоджувати свої дії та відсутності повної інформованості про стан завантаженості та наявності вільних ресурсів мережі розробка таких алгоритмів представляє складну проблему. При розгляді задачі оптимізації необхідно виділити критерії, які має максимізувати або мінімізувати алгоритм (або принаймі гарантувати певний рівень).

Уникнення перевантаження. Головна причина розробки алгоритму TCP полягала у запобіганню перевантаження мережі. За відсутності достовірної інформації про стан мережі алгоритм вважає ознакою перевантаження втрату пакетів. Це необхідна умова (якщо відбулось перевантаження, то пакети відкидаються, тобто втрачаються) але іноді втрати пакетів можуть спричинятися збоями обладнання, помилками і навіть погодою (для безпровідних мереж). В сучасних схемах також використовується такий неявний індикатор можливого перевантаження, як зростання часу на доставку пакета.

Ефективне використання ресурсів. Працездатність мережі вимірюється пропускною здатністю її вузького місця. Якщо вузьке місце працює не у повну потужність, то ресурси втрачаються. Тому, якщо у процесі роботи системи у вузькому місці вивільнилися додаткові ресурси, алгоритм має вміти використати їх у розумно найшвидший спосіб.

Справедливість розподілу. Спільне використання потужностей мережі різними користувачами вимагає здійснювати явний або неявний розподіл ресурсів поміж ними. Алгоритм, таким чином, має забезпечувати прогнозовану та справедливу схему поділу наявних потужностей.

1. Опис TCP алгоритмів

Найважливішим елементом алгоритму TCP, який вирізняє його поміж інших є 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.

Керування на основі вікна та механізм ACK повідомлень не допускає тотального перевантаження мережі і адаптується до змін в обсязі вільних ресурсів (у сторону збільшення або зменшення) та дозволяє у поєднанні з AQM алгоритмами розподіляти ресурси між користувачами. Як уже зазначалось, основною інформацією про стан мережі є підтвердження доставки пакета або його втрати. Взагалі кажучи причини втрати пакета можуть бути різними, однак ми зосередимося на втратах, спричинених перевантаженнями мережі. Кожний пакет підтвердження включає ідентифікатор пакета, що наразі очікується адресатом. Іншими словами отримання ACK пакета з ідентифікатором N означає успішну доставку всіх пакетів з номером $N-1$. Якщо пакет M було втрачено, то джерело буде отримувати ACK пакети з ідентифікатором M за кожний пакет, що надійде після факту втрати до того часу поки пакет M не буде надіслано знову. Таким чином, втрата пакета спричиняє як мінімум дублювання підтвердження. Однак причин появи дублюючих ACK пакетів більше – це може статись внаслідок затримки пакета і надходження його не у порядку слідування (внаслідок зміни маршруту руху) або через дублювання ACK повідомлення мережею. Тому рішення про втрату пакета приймається на стороні джерела. Це стається внаслідок настання однієї з двох подій: перевищено час на очікування підтвердження (змінна з'єднання RTO) або отримано три дублюючих ACK підтвердження. Значення RTO не є постійним а визначається на основі усередненого RTT. Також якщо відбувається втрата пакета після перевищення RTO застосовується експонентний механізм збільшення.

Опис основних станів системи TCP. Виділяють чотири окремих стани TCP з'єднання. З кожним з'єднанням пов'язані змінні: поточне вікно – *cwnd* та *ssthresh* – параметр переходу в стан уникнення перевантаження.

При встановленні TCP з'єднання *cwnd* дорівнює 2 а *ssthresh* нескінченності.

Повільний старт. Перший стан в якому знаходиться з'єднання у момент встановлення або після відновлення. Якщо $cwnd < ssthresh$ то розмір вікна зростає на одиницю за кожний пакет підтвердження. Іншими словами протягом RTT періоду вікно подвоюється. Якщо вважати RTT постійне, то вікно зростає за експонентою. Стан повільного старту продовжується до виникнення однієї з наступних подій:

- змінна *cwnd* стає більшою за $ssthresh$. TCP з'єднання переходить у стан уникнення перевантаження;
- затримка підтвердження перевищує RTO (пакет вважається втраченим за таймаутом). Перехід у стан очікування;
- виникнення дублюючого підтвердження. На відміну від попередньої події, яка свідчить про «значне» перевантаження доставка дублюючого ACK пакета означає, що пакет втрачено через тимчасовий збій або мережа підійшла до стану перевантаження і слід зменшити швидкість. Перехід у стан швидкого відновлення.

Уникнення перевантаження. В даному режимі *cwnd* збільшується на одиницю за кожний RTT. Як і в попередньому стані перевищення RTO або дублюючий ACK переводить з'єднання у відповідний стан.

Очікування. Після виникнення тайм-ауту (перевищення часу очікування поточної величини RTO) виконуються наступні дії:

- втрачений пакет надсилається повторно;

- оновлюються значення змінних $ssthresh = \frac{cwnd}{2}$, $cwnd = 1$;

- значення RTO подвоюється.

Стан очікування виникає при суттєвій затримці доставки пакета, що означає значне перевантаження мережі. В такій ситуації доцільно тестувати мережу тільки через зростаючі періоди часу, які визначаються змінною RTO. В кінці кінців або буде підтверджено доставку пакета і з'єднання перейде у стан повільного старту або користувач закриє з'єднання.

Швидке відновлення. TCP з'єднання потрапляє у стан швидкого відновлення після виникнення трьох дублюючих АСК повідомлень. При цьому оновлюється значення $ssthresh = \frac{cwnd}{2}$ і відбувається повторне надсилання пакета. Якщо час очікування перевищує RTO то з'єднання переходить у стан очікування; якщо АСК успішно отримано, то з'єднання переходить у стан уникнення перевантаження з $cwnd = ssthresh$. Зменшення вікна наполовину означає (за умови рівномірного надходження пакетів підтвердження), що протягом половини RTT з'єднання буде очікувати на підтвердження отримання пакетів і лише після досягнення кількості пакетів у мережі розміру вікна зможе відновити надсилання.

2. Потокове моделювання мереж TCP

Опишемо динамічну потокову модель мережі [8]. 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. Користувач надсилає у мережу потік своїх пакетів, керуючи швидкістю передачі – функцією $\lambda(t)$. Введемо множину індексів черг $J = \{1,2,...,M\}$ та відповідний вектор черг $q_j(t)$. Кожна черга зв'язана з обчислювальним ресурсом $u_j(t)$, $j \in J$, який обслуговує потік пакетів.

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 пакета 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 в яку потрапляють пакети з j - ої черги.

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

$$l_j(\lambda, q) = \begin{cases} 0 & q < q^{\max} \\ \min\left\{-\left[\lambda + Bu\right]_j, 0\right\} & q \geq q^{\max} \end{cases}, \; j \in J.$$

Введемо також функцію сумарних втрат користувача за всіма чергами $\Lambda(t)$. Керування $u(t) = u(\lambda, q)$ будемо вважати функцією, неперервною за λ та за q для $q \in \operatorname{int} Q$. В точках границі Q $u(\lambda, q)$ може мати розриви першого роду.

Після закінчення обслуговування пакетів користувача сервер надсилає йому підтвердження про успішну обробку. Позначимо $v(t)$ 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

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

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

$$\frac{d\overline{q}(t)}{dt} = \overline{\lambda}(t) + B\overline{u}(t) + l(\overline{\lambda}(t), \overline{q}(t)). \qquad (1)$$

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 $\left[\lambda + Bu\right]_j \le d_j$, $j \in J$;
- обсяги черг більші за нуль та обмежені за величиною – $0 \le q_j(t) \le q_j^{\max}$, $j \in J$;
- керування $u(t,q)$ будемо вважати кусково неперервною функцією зі значеннями з опуклого компакту відповідного простору. Особливістю схем керування мережами є залежність функції від стану $q(t)$ та розривність за цією змінною;
- $\lambda(t)$ – невід'ємна кусково неперервна функція, обмежена за величиною $0 \le \lambda(t) \le \lambda^{\max}$, $j \in J$. Якщо користувач працює в рамках TCP мережі, 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:

$$\int_0^T \left[\lambda(t) - v(t) + l(t)\right]dt \le w(T). \tag{2}$$

Нерівність (2) 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.

Запишемо (1) у скороченій формі $\dot{q} = f(\lambda(t), q)$. Ця система диференціальних рівнянь визначена в області $Q \times L$, де $Q = \left\{ q \in R^M : 0 \le q_i \le q_i^{\max} \right\}$, $L = \left\{ (x,0,...,0), \quad 0 \le x \le \lambda^{\max} \right\}$. Функцію $\lambda(t)$ будемо вважати кусково неперервною. Зафіксуємо $\lambda(\cdot)$, тоді $f(\lambda(t), q) = f(t, q)$. Будемо говорити, що функція задовольняє умовам Каратеордорі, якщо в області $Q \times T$:

1) $f(t,q)$ визначена для майже всіх t і неперервна за q;

2) $f(t,q)$ вимірна за t для кожного q;

3) $\left|f(t,q)\right| \le m(t)$, де $m(t)$ інтегрована за Лебегом на кожному скінченому відрізку.

Теорема 1. (Філіпов, [9]) Якщо функція $f(t,q)$ задовольняє умовам Каратеодорі при $t_0 \le t \le t_0 + a$, $\left|q - q_0\right| \le b$, то на відрізку відрізку $[t_0, t_0 + d]$ існує розв'язок задачі $\dot{q} = f(t,q)$, $q(t_0) = q_0$. При цьому число d таке, що $0 < d \le a$, $\phi(t_0 + d) \le b$, $\phi(t) = \int_{t_0}^{t} m(s)ds$.

Зауваження. Система (1) не задовольняє умовам Каратеодорі, оскільки $f()$ розривна за q.

Теорема 2. (Філіпов [9]) Нехай $\dot{q} = f(t,q)$ – рівняння Каратеодорі в замкнутій обмеженій області D. Тоді кожний розв'язок, що проходить всередині D можна продовжити в обидві сторони до виходу на границю області.

Твердження 1. Якщо $\lambda(t)$, $t \in [t_0, t_1]$ кусково-неперервна, обмежена функція, і $f(t,q)$ неперервна у $\operatorname{int} Q$ то існує розв'язок системи (1) в області $Q \times L$.

Доведення приведене у роботі [10].

Покажемо тепер, що задача завантаження файлу, тобто задача (1) з функціоналом якості $J(\lambda(t)) = \min\left\{ t \ge 0 : \int_0^t v(\tau)d\tau \ge \lambda^{\text{int}} \right\}$ має розв'язок. Введемо додаткову змінну q_0 таку, що $q_0(t) = \lambda^{\text{int}} - \int_0^t v(\tau)d\tau$. Моментом закінчення завантаження назвемо перший момент часу t для якого $q_0(t) = 0$. Зауважимо, що

$$\dot{q}_0 = v(\tau), \; q_0(t_0) = \lambda^{\text{int}}. \tag{3}$$

Формалізуємо визначення функції $v(\tau)$. Оскільки $v(\tau)$ – потік підтверджених пакетів, що покидають один з вузлів системи, то існує індекс $i \in I$, такий, що $u_i(t) = v(t)$. Позначимо L – множину можливих значень функції $\lambda(t)$ для будь якого $t \ge 0$. Будемо вважати, що L – опуклий компакт.

Твердження 2. Якщо для системи (1)-(3) виконується $BU \underline{*} L = \bigcap_{\lambda \in L} (BU - \lambda) \ne \varnothing$ і момент закінчення завантаження $T(L, q(t_0)) < \infty$, то для кожної $\lambda^*(t) \in L$ існує $u^*(t) \in U$, така, що $T(\lambda^*(\cdot), u^*(\cdot), q(t_0)) = T(L, q(t_0))$, черга $q_0(t)$ спадають монотонно і втрати пакетів нульові.

Доведення. Розглянемо множину досяжності системи (1). Нехай задана функція $\lambda^*(t) \in L$, тоді $Q(t) = \{q \in Q : q = q(t_0) + \int_0^t (\lambda^*(\tau) + Bu(\tau))d\tau\}$, де $u(\tau)$ – будь-яка допустима функція. Зрозуміло, що $Q(t) = q(t_0) + \int_0^t (\lambda^*(\tau) + BU)d\tau$ компактна, опукла множина, що неперервно залежить від часу. Позначимо лінійний підпростір, що породжений вектором q_0 через M, π – оператор ортогонального проектування на його ортогональне доповнення $M^{\perp}$. Умова $T(L, q(t_0)) < \infty$ означає існування моменту часу коли $\{0\} \in \pi Q(T)$. Мінімальний такий час існує і дорівнює $T = T(L, q(t_0))$.

Тоді можна вибрати селектор $w \in [BU^*_- L]$, такий, що $\int_{t_0}^T w d\tau = -q(t_0)$. Визначимо $u^*(t)$ з умови $Bu^*(t) = -\lambda^*(t) + w$, тоді рівно в момент часу T виконується $q_0(T) = 0$ і швидкість зменшення стаціонарна і дорівнює w. Доведення закінчене.

На практиці, однак, часто виникає ситуація, коли обсяг вхідного потоку перевищує можливості вузлів з обробки даних (можливо тимчасово). В цьому випадку зайві пакети очікують на обробку в черзі, а в разі переповнення черги – втрачаються. Виявляється, що за певних умов для задачі (1)-(3) можна підібрати таку функцію втрат, щоб існував скінчений час закінчення завантаження і відповідна стратегія.

Твердження 3. Якщо для системи (1) – (3) виконується умова $BU^*_- L = \bigcap_{\lambda \in L} (BU - \lambda) \neq \varnothing$ і існує момент закінчення завантаження $T(L, q(t_0)) < \infty$, Нехай задана функція вхідного потоку даних $\lambda(t) \in CL$, для якогось $C > 1$, тоді існує функція втрат $l(\lambda) \in R^{M+1}$, така, що $\bigcap_{\lambda \in L} ((BU + l(\lambda)) - \lambda) \neq \varnothing$ і момент закінчення завантаження $T(\lambda(\cdot), q(t_0)) < \infty$.

Доведення. Розглянемо $l(\lambda) = \frac{C-1}{C}\lambda$, тоді $l(\lambda) - \lambda \in L$ і можна застосувати твердження 2. Покажемо, що моменти закінчення завантаження співпадають. Для кожної функції $\lambda(t) \in CL$ існують $l(\lambda(t))$, $u(\lambda(t))$ такі, що $\lambda(t) + Bu(\lambda(t)) + l(\lambda(t)) \in BU^*_- L$. Тоді

$$q(t) = q(t_0) + \int_0^t (\lambda(\tau) + Bu(\tau) + l(\tau))d\tau \in q(t_0) + tBU^*_- L.$$

Отже, час закінчення не може бути меншим за $T(L, q(t_0))$. З іншого боку, завжди можна підібрати $l(\lambda(t))$, щоб виконувалась рівність $\lambda(t) + Bu(\lambda(t)) + l(\lambda(t)) = w$, де w вибране з попереднього твердження, тобто так, щоб $\int_{t_0}^T w d\tau = -q(t_0)$. Таким чином швидкість зменшення $q_0(t)$ збігається і $T(\lambda(\cdot), q(t_0)) = T(L, q(t_0))$.

Наслідок. Для пари функцій $\lambda(t)$, $l(\lambda)$ моделі з втратами існує $\lambda^*(t) \in L$ моделі без втрат, що моменти закінчення завантаження збігаються.

Розглянемо модель мережі з одним користувачем, маршрутизатором і сервером. Обмежимося розглядом системи з двома режимами – уникнення перевантаження і швидким відновленням. Керування на основі вікна означає, що в стані рівноваги вікно переключається між цими двома режимами. Спочатку вікно лінійно зростає до максимального значення $w^{\max}$. При виникненні переповнення черги виникає втрата, внаслідок якої вікно зменшується удвічі.

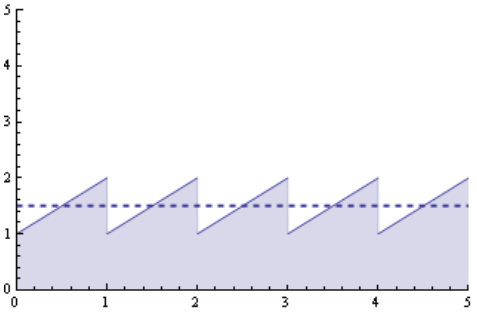

Рис. 2

Усереднене вікно $w_{cp} = \frac{1}{RTT}\int_{t}^{t+RTT} w(t)dt = \frac{3}{4}w^{\max}$. Припущення щодо динаміки процесу:

- будемо вважати RTT сталими і рівним $\frac{w^{\max}}{2}$;
- система має одне вузьке місце, обробка пакетів в інших чергах відбувається «миттєво»;
- повідомлення про доставку також надходять миттєво.

Тоді швидкість передачі даних дорівнює $\lambda^* = \frac{3}{4}w^{\max}RTT^* = \frac{3}{4}w^{\max}\frac{u^*}{q^*}$, де $q^* = \frac{q^{\max}+q^{\min}}{2}$ – усереднена черга. Значення $q^{\max}$ визначене. Розглянемо динаміку зміни обсягу черги $q(t)$ після факту зменшення вікна удвічі. Оскільки джерело не може надсилати нічого до часу t^* поки кількість пакетів у системи не буде рівна розміру вікна. Для часу t^* виконуються рівності:

$$q^{\min} = q^{\max} - u^*t^* = w(t^*) = \frac{w^{\max}}{2} + t^*,$$

$$t^* = \frac{q^{\max} - \frac{w^{\max}}{2}}{u^*+1}.$$

Тоді усереднена черга дорівнює $q^* = q^{\max}\frac{u^*+2}{2u^*+2} + \frac{u^*w^{\max}}{4(u^*+1)}$, $\lambda^* = \frac{3}{4}w^{\max}\frac{u^*}{q^*}$.

Розглянемо задачу завантаження файлу. Нехай задана загальна кількість пакетів λ_{int}, які потрібно завантажити на сервер за найкоротший час. Завантаження вважається закінченим, якщо кількість підтверджених пакетів досягає заданої. Динаміка системи описується системою диференціальних рівнянь:

$$\begin{aligned}\dot{q}_1(t) &= \lambda(t) - u_1(t),\\ \dot{q}_2(t) &= u_1(t) - u_2(t),\\ \dot{q}_3(t) &= u_2(t) - u_3(t),\end{aligned} \tag{4}$$

при цьому на фазові змінні і функції накладені наступні обмеження:

$$0 \le \lambda(t) \le \lambda^{\max},$$

$$u_i(t) \ge 0,\ \frac{u_1(t)}{\mu_1} + \frac{u_3(t)}{\mu_3} \le 1,\ u_2(t) \le \mu_2,$$

$$0 \le q_i(t) \le q_i^{\max},\ q_i(t_0) = 0,\ i = 1,2,3.$$

Функції обробки пакетів на маршрутизаторах $u_i(t)$ в даному прикладі працюють за принципом *FIFO*. Крім цього на першому вузлі більший пріоритет надається пакетам підтвердження. Загальноприйняті [16] визначення функцій керування призводять до виникнення ковзних режимів, що ускладнює їх аналіз. В роботах [11 – 16] розглянуто та обґрунтовано підхід до визначення $u_i(t)$, який усуває ковзні режими.

$$u_1(t) = \begin{cases} \min\left(\lambda(t), \mu_1\left(1 - \frac{u_3(t)}{\mu_3}\right)\right), & q_1(t) = 0 \\ \mu_1\left(1 - \frac{u_3(t)}{\mu_3}\right), & q_1(t) > 0 \end{cases};\ u_2(t) = \begin{cases} \min\left(\mu_2, u_1(t)\right), & q_2(t) = 0 \\ \mu_2, & q_2(t) > 0 \end{cases};$$

$$u_3(t) = \begin{cases} \min\left(\mu_3, u_2(t)\right), & q_3(t) = 0 \\ \mu_3, & q_3(t) > 0 \end{cases}. \tag{5}$$

З тверджень 1 – 3 випливає існування оптимального керування для задачі (4) з функціями (5). Характерною особливістю задач даного типу є наявність «значної» кількості оптимальних керувань $u^*(t)$, що обґрунтовується наступним твердженням.

Твердження 4. Якщо керування $\lambda^*(t)$ оптимальне, то вузьке місце максимально завантажене протягом усього часу завантаження файлу. Якщо за керування $\lambda^*(t)$ вузьке місце максимально завантажене протягом усього часу завантаження файлу, то відповідний час оптимальний.

Твердження доведене у роботі [17].

Наслідок. Керування $\lambda(t)=\lambda^{\max}$ завжди є оптимальним.

3. Ігрові потокові моделі мереж

Розглянемо систему з двома користувачами що завантажують файли на один сервер [25, 26]. При цьому вони використовують спільну мережу. Для моделювання процесу їх взаємодії в рамках підходу потокових моделей і теорії ігор, введемо *віртуальні з'єднання* $q_i^k(t)$, де індекс k означає користувача, індекс i відповідає черзі. Віртуальне з'єднання означає, що ресурси одного вузла розподіляються (за допомогою функції $u_k(t)$) між обслуговуванням різних потоків пакетів користувачів, тобто $\sum_k q_i^k(t)=q_i(t)$. Таким чином, задача зводиться до системи з динамікою виду (1) для функцій $\lambda_i(t)$, кожна з яких зв'язана з окремим користувачем та використовується для мінімізації часу завантаження.

$$\frac{d\overline{q}(t)}{dt}=\overline{\lambda}_1(t)+\overline{\lambda}_2(t)+B\overline{u}(t)+l(\overline{\lambda}(t),\overline{q}(t)). \tag{7}$$

При цьому функції втрат мають вигляд

$$l_i(t)=\begin{cases}0, & q_i(t)<q_i^{\max}\\ [\overline{\lambda}(t)+B\overline{u}(t)]_i, & q_i(t)=q_i^{\max}\end{cases},$$

функції обробки $u_i^k(t)$ розподіляються пропорційно до вхідних потоків.

Існування розв'язку для допустимих функцій $\lambda_1(t)$, $\lambda_1(t)$ випливає з попередніх тверджень.

Твердження 5. Нехай для моделі з двома користувачами задані функції $\lambda_1^*(t)$, $\lambda_2^*(t)$ і $\lambda_1^*(t)\neq\lambda_1^{\max}$. Позначимо час закінчення завантаження T_1^*, T_1^* відповідно. Тоді перший користувач може за допомогою керування $\hat{\lambda}_1(t)\equiv\lambda_1^{\max}$ зменшити свій час завантаження. Існує рівновага за Нешем у грі завантаження (1). Зокрема, керування $\lambda_1^*(t)=\lambda_1^{\max}$, $\lambda_2^*(t)=\lambda_2^{\max}$ дають точку рівноваги.

Доведення. Зафіксуємо $\lambda_1^*(t)$, $\lambda_2^*(t)$. Розглянемо $p(t)=q_0^1(t)+q_0^2(t)$, де

$$\dot{q}_0^1(t)=v_1(t),\ \dot{q}_0^2(t)=v_2(t).$$

Якщо вузьке місце не завантажене пакетами, то збільшення $\lambda_1(t)$ призведе до збільшення частки пакетів першого користувача, що проходять і тим самим зменшать час закінчення.

Якщо вузьке місце системи вже працює на максимальному режимі, то час закінчення буде визначатись часткою, яка припадає на пакети кожного користувача (ця частка може залежати від часу). Однак і в цьому випадку збільшення $\lambda_1(t)$ до максимуму збільшить частку пакетів, що обробляються (можливо збільшить і втрати також), що, для фіксованої функції $\lambda_2(t)$ зменшить час завантаження.

Наслідок. Якщо користувач зменшує свою швидкість $\lambda_1(t)$, то час завантаження принаймі не зростає. Тоді існує рівновага за Нешем у грі завантаження. Зокрема, керування $\lambda_1^*(t)=\lambda_1^{\max}$, $\lambda_2^*(t)=\lambda_2^{\max}$ дають точку рівноваги.

Модель завантаження з одним користувачем за умов атаки на відмову

Розглянемо роботу системи (7) за наявності атаки. В даній роботі ми розглядаємо спрощену модель поглинаючої атаки [6]. Трафік атаки потрапляє на сервер, вимагає ресурсів на обробку та, після обробки, відкидається як помилковий. Виявляється, що атака різних ланок мережі суттєво відрізняється за наслідками для роботи мережі. Будемо розглядати атаку мережі з одним користувачем, та за виконання наступних припущень:

- маршрутизатор є вузьким місцем, потужність сервера набагато більша;
- атака фільтрується під час обробки на маршрутизаторі.

Динаміку процесу атаки мережі запишемо у вигляді рівняння:

$$\dot{\overline{q}}(t)=\overline{\lambda}(t)+\overline{\alpha}+B\overline{u}(t)-\overline{l}(t), \tag{8}$$

де $\overline{\alpha}=(\alpha_1,...,\alpha_3)$ – трафік атаки.

Проаналізуємо атаку вигляду $(\alpha_1,0,\alpha_3)$ системи (8). Нехай система з одним користувачем перебуває у стані рівноваги, тоді відомі значення середнього вікна, швидкості і часу завантаження файлу:

$$T^*(q^{\max},u^*,w^{\max})=\frac{\lambda_{\text{int}}}{\lambda^*}=\frac{4\lambda_{\text{int}}}{3(u^*+1)}\left[\frac{q^{\max}}{w^{\max}u^*}+\frac{q^{\max}}{2w^{\max}}+\frac{1}{4}\right]. \tag{9}$$

Твердження 6. Атака вигляду $(\alpha_1,0,0)$ призведе до нового мінімального часу закінчення завантаження $T(\alpha) = T^*(q^{\max}, u^*, w^{\max} - \alpha_1 \frac{q^{\max}}{u^*})$, якщо $\alpha_1 \le \frac{w^{\max} - 2}{q^{\max}} u^*$ і $T = +\infty$ інакше.

Доведення. Оскільки ресурси діляться пропорційно до потоків користувачів, то для опрацювання пакетів атаки знадобляться ресурси α_1. Якщо α_1 більше за u^*, то черга переповнюється і, оскільки джерело атаки не зменшує свою швидкість при втратах, швидкість користувача впаде до нуля.

Якщо $\alpha_1 < u^*$, то у черзі буде знаходитись $\alpha_1 t^*$ пакетів атаки, де $t^* = \frac{q^{\max}}{u^*}$ – час звільнення черги. Тоді максимальна кількість пакетів, що може перебувати в системі скорочується до $w^{\max} - \alpha_1 \frac{q^{\max}}{u^*}$. Оскільки справжнє вікно коливається навколо середньої величини, то для оцінки працездатності достатньо вимагати, щоб $w_{\min} \ge 1$. Виконуються співвідношення

$$1 \le w_{\min} = \frac{w^{\max} - \alpha_1 \frac{q^{\max}}{u^*}}{2},$$

$$\alpha_1 \le \frac{w^{\max} - 2}{q^{\max}} u^*.$$

Скористаємося формулою (9) для нового значення максимального вікна.

На рис. 3 показані залежності часу виконання від обсягу трафіка атаки для різних значень потужності сервера u^*.

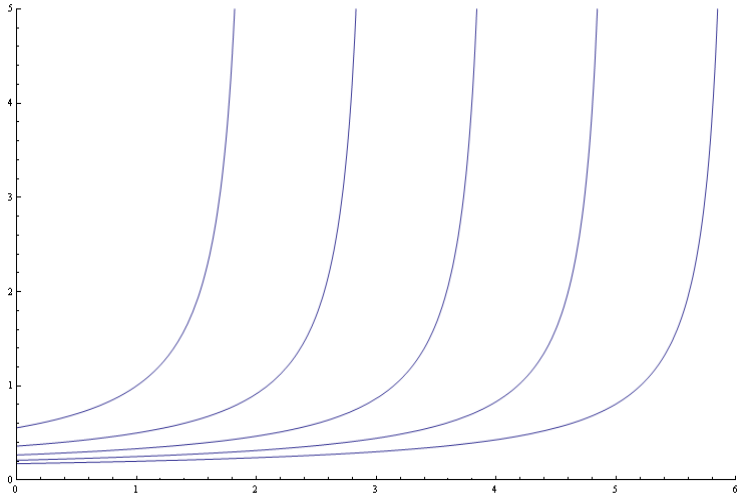

Рис. 3. Вплив атаки на час завантаження

Висновки

В роботі здійснено огляд механізмів керування у мережах Інтернет, описані основні елементи TCP протоколу та розроблена ігрова потокова моделі роботи мережі. Ця модель представляє систему керованих диференційних рівнянь з розривною правою частиною. Поставлена задача оптимального завантаження файлу та знайдені необхідні умови існування її розв'язку. Розглянуто задачу взаємодії двох користувачів, показано існування рівноваги за Нешем.

Для моделі з топологією користувач-сервер-маршрутизатор розв'язується задача впливу атаки на відмову. Знайдені аналітичні залежності погіршення якості (часу завантаження файлу) в залежності від обсягу трафіка атаки. Проведене імітаційне моделювання для різних типів протоколів. Таким чином, в результаті виконання роботи побудована аналітична потокова модель керування потоками даних у комп'ютерних мережах. Розроблена модель дозволяє провести аналіз поведінки користувачів в рамках диференційної гри та оцінити якість і роботу системи керування а також знайти точки рівноваги мережі. Для різних типів атак знайдені стратегії протидії, проведене імітаційне моделювання основних типів протоколів та їх вразливості до атак на відмову.

1. *Kelly F.P.* Charging and rate control for elastic traffic // European Trans. on Telecommunications. – 1997. – **8**. – P. 33 – 37.
2. *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.
3. *Srikant R.* The Mathematics of Internet Congestion Control. – Springer Verlag, 2004. – 137 p.
4. *Low S. Paganini F., Doyle J.* Internet congestion control // IEEE Control Syst. Mag. – 2002. – **22** (1). – P. 28 – 43.
5. *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.
6. *Андон П.І., Ігнатенко О.П.* Атаки на відмову в мережі Інтернет: опис проблеми та підходів щодо її вирішення / Ін-т програмних систем. – Препр. – К.; 2008. – 50 с.
7. *Welzl M.* Network Congestion Control: Managing Internet Traffic. Wiley. – 2005. – 263 p.
8. *Meyn S.* Control Techniques for Complex Networks. – Cambridge University Press. – 2007. – 582 p.
9. *Филиппов А.Ф.* Дифференциальные уравнения с разрывной правой частью. – М.: Наука, 1985. – 224 с.
10. *Ігнатенко О.П.* Моделювання процесів керування комп'ютерними мережами за умов конфлікту // Проблеми програмування. – 2009. – № 2-3. – С. 125 – 136.
11. *Ігнатенко О.П.* Потокові моделі динамічного планування черг в компю'терних мережах за умов конфлікту // Проблеми програмування. – 2010. – № 1. – С. 35 – 43.
12. *Ігнатенко О.П., Кордубан Д.О.* Розподіл ресурсів у багатопроцесорних середовищах за умов конфлікту і невизначеності // Зб. праць міжнар. конф. CSE-2010. – 2010. – С. 210 – 211.
13. *Ігнатенко О.П.* Моделі керування потоками даних мережі Інтернет за умов нестабільної поведінки // Проблеми програмування. – № 3, 2011. – С. 38 – 51.
14. *Ігнатенко О.П., Кордубан Д.О.* Потокові керовані моделі мереж з втратами за умов атаки на відмову // Матеріали V Міжнародної наукової конференції «Сучасні комп'ютерні системи та мережі: розробка та використання» (29 вересня – 01 жовтня 2011, Україна, Львів), 2011. – С. 72-73.
15. *Ігнатенко О.П.* Ігрові потокові моделі керування мережами // Матеріали Міжнародної наукової конференції CSE (24 вересня – 26 листопада 2011, Україна, Львів), 2011.
16. *Ігнатенко О.П.* Потокові керовані моделі мереж за умов атаки на відмову // Матеріали VI Міжнародної наукової конференції «Застосування інформаційних та комп'ютерних технологій» (12 – 14 жовтня 2011, Азербайджан, Баку), 2011. – С. 271 – 276.