

УДК 004.7(045)

**Є. В. Позняк асп.;****М. І. Гученко д. т. н., доц.**

## **ЗАСОБИ ПЕРЕДАЧІ ПОТОКОВОГО TCP-ТРАФІКУ НА БАЗІ NDIS ТА MATLAB ДЛЯ УМОВ ВИСОКОГО ДЖИТТЕРУ**

*Подано результати тестування засобів керування мультимедійним TCP-трафіком за допомогою NDIS-драйверу, що забезпечує роботу TCP-подібного протоколу на рівні доступу до каналу зв'язку та здатен імітувати динаміку Інтернет при передачі трафіку в локальній мережі. Розглянуто питання поліпшення TCP-протоколу для передачі трафіку потокового мультимедіа в Інтернет. Реалізацію TCP-подібного протоколу на базі NDIS доповнено засобами для пріоритетної передачі відеотрафіку та віртуальним буфером, що імітує RTT-затримки Інтернет. Для передачі мультимедійного TCP-трафіку в драйвері впроваджено алгоритм керування, що балансує швидкість прийому-передачі TCP-пакетів на основі статистики RTT-затримок. Відправка мультимедійного трафіку, регулювання роботи TCP-протоколу та отримання експериментальних результатів проводилось в середовищі Simulink.*

### **Вступ**

Більшість мультимедійного трафіку в Інтернет передається за допомогою транспортного протоколу TCP, а на рівні додатків найчастіше використовуються протоколи на основі гіпертекстових повідомлень (так майже половину HTTP-трафіку в Північній Америці в 2006 році склала передача потокової аудіо- та відеоінформації [1]). Існує практична проблема забезпечення якості обслуговування (QoS) мультимедійного трафіку в мережах Інтернет. Крім того, застосування протоколу TCP для трафіку реального часу є недостатньо ефективним, поряд із можливостями звичайних IP та UDP-протоколів, оскільки стандартні алгоритми керування потоками даних у TCP-протоколі мало пов'язані з протоколами верхніх рівнів, неефективно розподіляють смугу пропускання для потокового мультимедіа, різко знижують інтенсивність трафіку у разі втрати пакетів та неякісно працюють в умовах сильної флуктуації затримок передачі. А отже, існує необхідність розробки модифікацій протоколу для поліпшення його роботи в нових типах мереж (перш за все — для бездротових або протяжних каналів) та для поліпшення обслуговування нових видів трафіку.

### **Аналіз публікацій**

Аналіз публікацій показує, що оптимізація TCP-протоколу проводиться за такими напрямками:

1. Модифікація схеми керування потоками даних для зменшення впливу втрат пакетів та більш ефективного й справедливого розподілення каналу зв'язку [2];
2. Підтримка роботи TCP з використанням допоміжних протоколів (наприклад в ATCP [2]);
3. Підтримка TCP за рахунок регулювання керувальної інформації з боку каналного рівня на основі моделі AIMD TCP-протоколу [3];
4. Реалізація спеціальних схем керування потоками для поліпшення обслуговування пріоритетних видів трафіку [4, 5].

В роботі розглядається останній варіант. На сьогоднішній день розроблена значна кількість алгоритмів для оптимізації передачі мультимедійного трафіку в умовах протяжних каналів зв'язку із високим джиттером та втратами пакетів. Здебільшого ці алгоритми спираються на протокол UDP або на транспортні служби мереж ATM. Впровадження нестандартних алгоритмів керування потоковим мультимедіа у протоколі TCP є проблематичною задачею у зв'язку з необхідністю модифікації вбудованої в ядрі операційної системи реалізації протоколу. В той же час реалізація TCP-протоколів можлива на базі технології TCP Chimney Offloading або з допомогою проміжних драйверів. Використання NDIS-драйверу [6] дало можливість розвантажити мережний стек, поглибити міжпротокольную взаємодію, збільшити швидкість обміну даних. Крім того, ця технологія дає мож-

ливість швидкої розробки TCP-подібних протоколів призначених для оптимального обслуговування пріоритетного TCP-трафіку в Інтернет.

Для забезпечення найефективнішої передачі потокового трафіку в мережі необхідно підтримувати чітко визначені швидкості надходження даних до приймача, але, наприклад, для об'єднаних мереж та бездротових каналів це практично неможливо. Для балансування швидкостей та підтримки порядку надходження поточкових даних необхідно виконувати шейпінг трафіку за алгоритмом *token bucket* за допомогою буфера затримки в приймачі. Тут виникає оптимізаційна задача, щодо максимальної утилізації буфера приймача при мінімізації швидкості відправки даних TCP-протоколом.

Флуктуація затримок для мереж Інтернет (особливо для супутникових каналів) може значно перевищувати швидкість зчитування потокового мультимедіа приймачем трафіку. Для поліпшення обслуговування мультимедійного трафіку в умовах високої флуктуації (джиттеру) затримок в роботі [5] описано алгоритм, що дозволяє балансувати швидкість прийому-передачі TCP-пакетів на основі статистики RTT-затримок. Алгоритм базується на основі моделі з роботи [4] прийому-передачі трафіку в умовах високої флуктуації затримок

$$P_k = b_{1,k}R_k + b_{2,k}R_{k-1} + \dots + b_{i,k}R_{k-i-1}, \quad (1)$$

де  $R$  і  $P$  — відповідно, швидкості відправлення і надходження даних до приймача, дискретизованих проміжками часу зчитування даних з буфера приймача,  $k$  — номер інтервалу зчитування,  $b_{i,k}$  — доля пакетів переданих на  $k-i$  інтервалі зчитування.

На основі даної моделі оптимізаційний критерій ( $J_k$ ) швидкості передачі й завантаження буфера мультимедіа розраховується з формули

$$J_k = (\omega_p Q_k - \omega_q Q_r)^2 + (\omega_r R_k)^2, \quad (2)$$

де  $Q_r$ ,  $Q_k$  — максимальний й поточний рівні завантаження буфера приймача;  $w_p$ ,  $w_q$ ,  $w_r$  — вагові коефіцієнти (обираються від 1 до  $n$ ).

Керування трафіком мультимедіа відбувається шляхом регулювання міжпакетних інтервалів відправки пакетів даних, що розраховуються на основі оцінок розподілу RTT-затримок з дискретизацією зчитування даних приймачем мультимедіа, та швидкістю підтвердження прийому даних на основі ACK-пакетів. Але в роботі [5] тестування даного алгоритму було проведено лише на моделі мережі. В наступній роботі описано засоби проведення натурних експериментів з модифікаціями TCP-протоколу, розроблені на базі середовищ NDIS та Matlab.

### Мета дослідження

*Метою роботи є експериментальна перевірка ефективності призначеного для умов Інтернет алгоритму керування мультимедійним TCP-трафіком впровадженого на базі NDIS-драйвера, який забезпечує роботу TCP-подібного протоколу на рівні доступу до каналу зв'язку та здатен імітувати динаміку Інтернет при передачі трафіку в локальній мережі. Для цього необхідно розв'язати такі задачі: в NDIS-драйвері реалізувати віртуальний буфер для моделювання динаміки затримок в Інтернет; доповнити TCP-подібний протокол на базі NDIS-драйвера засобами керування для пріоритетної передачі відеотрафіку; з використанням засобів середовища Simulink забезпечити відправку трафіку мультимедіа, регулювання роботи NDIS-драйвера та отримання результатів роботи TCP-протоколу; перевірити якість роботи розробленого алгоритму керування трафіком мультимедіа.*

### Основний матеріал дослідження

Для моделювання RTT-затримок Інтернет в NDIS-драйвері реалізовано буфер затримки пакетів, що може затримувати кожний пакет на заданий час або підтримувати певний рівень завантаження буфера (за алгоритмом Drop Tail). Збір динаміки RTT-затримок Інтернет відбувався з допомогою програми подібної до утиліти *ping*. Для симуляції передачі мультимедійного трафіку в Інтернет був обраний набір даних з мінімальним RTT — 100 мс, середнім  $\approx$  200 мс та середньоквадратичним відхиленням  $\approx$  100 мс. Максимальна затримка передачі — 500 мс, після чого пакет вважався втраченим.

Середовище Matlab+Simulink має достатньо розвинену обчислювальну бібліотеку та засоби мо-

делювання. В цій роботі крім того пропонується використовувати Matlab та Simulink в якості бази для проведення натурних експериментів з мережним трафіком в реальному часі. Функціональні блоки Simulink таким чином представлятимуть в розгорнутому вигляді роботу транспортного протоколу та надаватимуть сервіси рівня сокетів. Структуру мережного стеку на базі NDIS та Matlab в Windows зображено на рис. 1. Використання Simulink та NDIS-драйверів в перспективі може також бути кросплатформним (із застосуванням Ndis-Wrapper в Linux). Експериментальна перевірка схем керування трафіком в Simulink є важливою для їх практичного впровадження, застосовуючи Layered Sockets Provider (LSP) в ОС Windows (або можливо й в Linux із застосуванням TCP-hook драйвера).

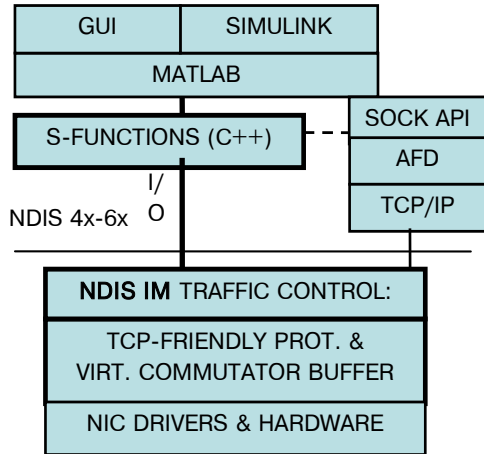


Рис. 1. Мережний стек NDIS+Matlab

Для тестування роботи розробленого алгоритму керування TCP-трафіком потокового мультимедіа [5] запропонована модель мережі в середовищі Simulink, що включає кероване та некероване джерела трафіку, проміжний мережний пристрій (комутатор), що встановлює затримки передачі даних, та приймач трафіку з буфером мультимедіа. Для проведення натурального експерименту Simulink використано в якості керованого джерела мультимедійного трафіку та віртуального комутатора, що моделює роботу Інтернет. Засоби керування трафіком в Simulink подано на рис. 2. Приймачем трафіку є звичайний TCP-клієнт, що працює на іншому хості в локальній мережі. В якості інформації про буфер приймача трафіку використовувалися дані про розмір TCP-вікна приймача та порядкові номери пакетів підтвердження.

В роботі в Simulink тестувався алгоритм контролю й підтримки завантаження буфера приймача розроблений на

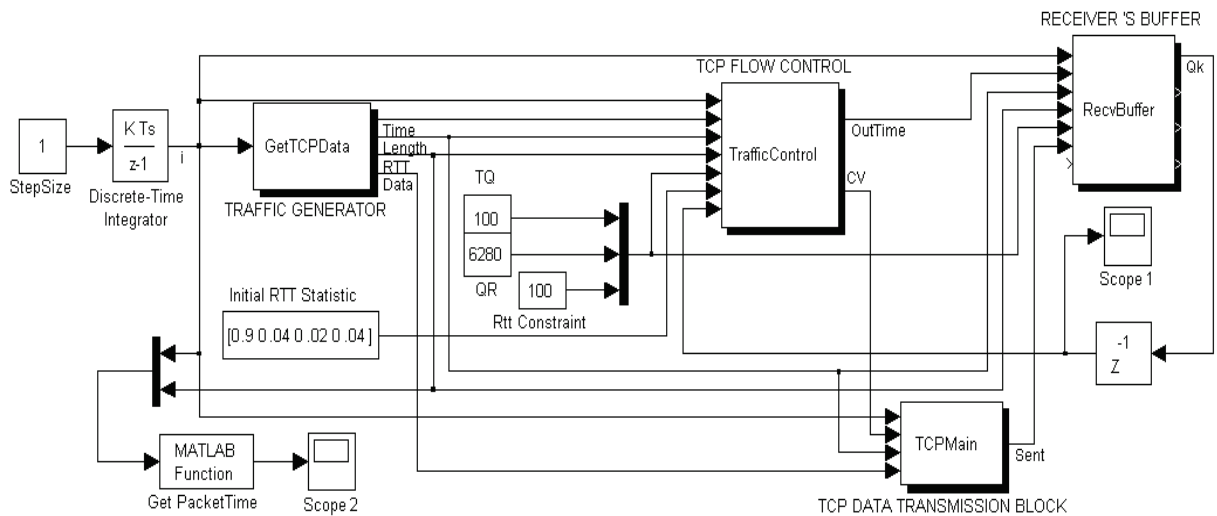


Рис. 2. Візуальні засоби керування потоковим TCP трафіком на базі NDIS в середовищі Simulink

базі робіт [4, 7] для використання в протоколі TCP. Відповідно до оптимізаційного критерію (2) задача алгоритму з роботи [4] — підтримувати максимально можливий рівень буфера, при одночасній мінімізації швидкості передачі. В роботі [7] керування буфером комутатора та боротьби з джиттером відбувається шляхом регулювання (на основі ICMP-повідомлень) міжкадрових (IDT) інтервалів відправки пакетів даних. При проведенні експерименту в Simulink підтримка рівня завантаження буфера приймача в схемі керування трафіком TCP-протоколу також відбувається регулюванням IDT-інтервалів відправки пакетів з використанням статистики (1) на базі оцінок поточних RTT-затримок та їх прогнозів. Статистика  $\{b_{i,k}\}$  розподілу затримок оцінюється на основі полів TCP-заголовку в пакетах підтвердження

$$\hat{b}_{i,k} = \frac{dT - TpAck_n}{dT} \hat{b}_{i,k} + \frac{LenAck_n}{R_{RttAck} - MinRtt}, \quad (3)$$

де  $LenAck_n$  — обсяг підтверджених даним пакетом даних;  $RttAck$  — RTT-затримка пакету з дискретизацією  $TQ$ ;  $MinRtt$  — мінімальна RTT-затримка;  $\hat{b}_{i,k}$  — поточні оцінки розподілення затримок за інтервал  $TQ$ ;  $TrAck_n$  — час між надходженням двох останніх АСК-пакетів;  $R_{RttAck-MinRtt}$  — швидкість передачі на час відправлення даних, прийом яких підтверджено цим АСК-пакетом.

Міжпакетний інтервал часу відправки пакетів регулюється згідно з формулою

$$Tp_n = w * Tp_n * \left( \frac{\Delta R_n * TQ}{b_{Rtt-MinRtt} * P_k * dT_n} \right), \quad (4)$$

де  $Tr_n$  — міжпакетний інтервал надходження двох останніх пакетів;  $b_{Rtt-MinRtt}$  — статистичний показник надходження пакетів для даного рівня  $\left( \frac{\Delta}{Rtt} \right)$  затримок;  $\Delta R_n$  — кількість вже відправлених пакетів на даному інтервалі  $TQ$ ;  $P_k$  — необхідна швидкість передачі даних мультимедіа;  $dT_n$  — час, що пройшов на даному інтервалі  $dT$ ;  $w$  — ваговий коефіцієнт ( $>1$  при наближенні до перевантаження, інакше  $<$  або рівний 1-ці).

Відправник трафіку та віртуальний комутатор в Simulink реалізовано у вигляді  $S$ -функцій (*GetTCPData.cpp*, *TrafficControl.cpp*, *TCPMain.cpp* та *RecvBuffer.cpp*), що працюють на базі NDIS-драйвера та забезпечують графічний інтерфейс. Регулятор мультимедійного трафіку запропонований в роботі [5] реалізовано в  $S$ -функції *TrafficControl.cpp* (блок «TCP FLOW CONTROL» на рис. 2). Дані відсилались, як і в модельному експерименті [5], у вигляді окремих пакетів, відправка кожного TCP-пакету означає одну ітерацію схеми керування трафіком. Під час проведення експерименту в локальній мережі відбувалась відсилка 65000 пакетів даних, для яких моделювались умови передачі даних в Інтернет. Середня швидкість передачі трафіка складала  $\approx 566$  кбіт/с без керування. На рис. 3 порівняні коливання завантаження буфера в умовах регулювання інтенсивності відправки пакетів на основі статистики RTT-затримок (4) та при звичайній роботі протоколу. На графіках видно, що регулювання трафіком, дозволяє дещо стабілізувати завантаження буфера, в той час, як рівень буфера при звичайній передачі TCP-протоколом в середньому вищий, має більшу амплітуду та частіше спадає до нуля. Загальні оцінки числа перевантажень та розвантажень TCP-буфера приймача наведено в таблиці. За рахунок зниження кількості перевантажень буфера, трохи знизилась і загальна інтенсивність передачі даних.

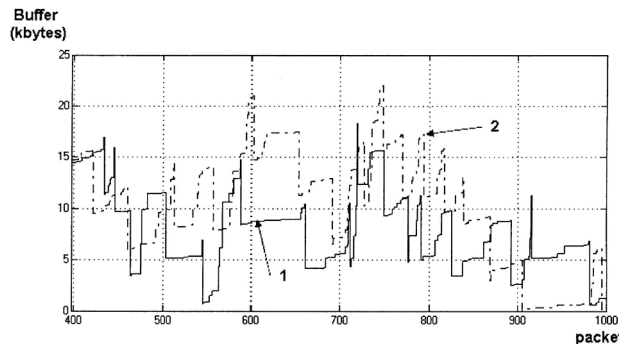


Рис. 3. Рівень завантаження буфера приймача (кб):  
— з використанням регулятора трафіку

#### Стабілізація завантаження буфера

Число перевантажень		Число розвантажень	
без керув.	з керув.	без керув.	з керув.
19	0	5	1

#### Висновки і перспективи

В роботі описано засоби керування потокового TCP трафіку на базі NDIS-драйвера, які призначені для балансування завантаження віддаленого TCP-буфера в умовах високої флуктуації RTT-затримок. В якості бази для проведення й подання даних натурного експерименту обрано середовище візуального моделювання — Simulink. Серед достоїнств Simulink: використання можливостей Matlab, візуальне подання процесів керування, можливість роботи системи керування (мережним трафіком) в реальному часі, швидкодія, наявність засобів обробки відео в реальному часі (Video Processing Blockset) й також є можливість перенесення коду в Linux. Серед

недоліків підходу можна назвати: необхідність програмування в значних обсягах, знання тонкощів роботи Simulink й Matlab. Використання візуального середовища тестування NDIS-драйверів є важливим для проведення досліджень та практичного впровадження засобів керування мережним трафіком, що працюють на нижніх рівнях моделі OSI. Для передачі звичайного відеотрафіку за допомогою розробленого NDIS-драйвера в ОС Windows пропонується використовувати спеціальний Layered Sockets Provider (LSP), який для окремих сокетів надаватиме пріоритетне обслуговування з використанням запропонованого TCP-подібного протоколу. Тестування NDIS роботи розробленого в [5] алгоритму дало позитивні результати, але для подальшого вдосконалення роботи алгоритму в умовах нестаціонарності затримок Інтернет необхідна розробка спеціального алгоритму прогнозування зміни статистики RTT-затримок.

## СПИСОК ЛІТЕРАТУРИ

1. «Ellacoya Data Shows Web Traffic Overtakes Peer-to-Peer (P2P) as Largest Percentage of Bandwidth on the Network», NXCcomm Show, CHICAGO — June 18, 2007 — Ellacoya Networks™, Inc. — (Офіційний документ від Ellacoya Networks). — Режим доступу до публікації : <http://www.ellacoya.com/news/pdf/2007/NXCcommEllacoyaMediaAlert.pdf>.
2. Ka-Cheong Leung, Victor O. K. Li, «Transmission Control Protocol (TCP) in wireless Networks: Issues, Approaches, and Challenges», IEEE Communications Surveys & Tutorials, Vol. 8, No. 4, 4th Quarter 2006. — Режим доступу до журн.: <http://www.comsoc.org/livepubs/surveys/public/2006/oct/index.html>.
3. Ki Baek Kim. Design of Feedback Controls Supporting TCP based on Modern Control Theory / Ki Baek Kim // INRIA Rocquencourt, Report No. 5014, Nov. 2003. — 37 p. — Режим доступу : <http://hal.archives-ouvertes.fr/docs/00/07/15/70/PDF/RR-5014.pdf>
4. Shu-Ching Chen. An Adaptive Rate-Control Streaming Mechanism with Optimal Buffer Utilization / Shu-Ching Chen, Mei-Ling Shyu, Irina Gray, Hongli Luo // Journal of Systems and Software, Special Issue on Adaptive Multimedia Computing. — March 2005. — Vol. 75, Issue 3. P. 271—282. — Режим доступу до журн.: <http://www.cs.fiu.edu/~chens/PDF/JSS05.pdf>.
5. Позняк Є. В. Поліпшення обслуговування мультимедійного TCP-трафіку засобами каналного рівня / Є. В. Позняк, М. І. Гученко, М. М. Іванова // Матеріали електронної техніки та сучасні інформаційні технології: міжнар. наук.-техн. конф., 21—23 травня 2008 р. : тези допов. — Кременчук, 2008. — С. 235—236.
6. Гученко М. І. Експериментальна реалізація TCP-протоколу на рівні доступу до каналу зв'язку/ М. І. Гученко, Є. В. Позняк, М. М. Іванова // Вісник Кременчуцького державного політехнічного університету імені Михайла Остроградського. — Кременчук, 2008(48) — Вип. 1., Ч. 1. — С. 35—38.
7. Макаренко А. В. Синтез адаптивної системи управління потоком кадрів в сетях Gigabit Ethernet [Електронний ресурс] / А. В. Макаренко // Журнал Радиоэлектроники. — 2002. — № 2. — Режим доступу до журн. : <http://jre.cplire.ru/jre/feb02/2/text.html>.

Рекомендована кафедрою обчислювальної техніки

Надійшла до редакції 8.09.08  
Рекомендована до друку 20.10.08

**Позняк Євген Володимирович** — аспірант, **Гученко Микола Іванович** — професор.

Кафедра комп'ютерних та інформаційних систем Кременчуцького державного політехнічного університету ім. М. В. Остроградського