

УДК 004.2:004.5

Коваль Ю.В., асистент

Iu. V. Koval, Assistant

**Протокол багатоточкового потоку:  
означення та застосування в створенні  
механізмів розпаралелювача та  
аналізатора подій в системі жестового  
інтерфейсу**

**Multipoint stream protocol: definition and  
application for parallelizer and events  
analyzer mechanisms in gestures interface  
system creation**

Київський національний університет імені  
Тараса Шевченка, 83000, м. Київ, пр-т.  
Глушкова 4д,  
e-mail: kafedraTK@unicyb.kiev.ua

Taras Shevchenko National University of  
Kyiv, 83000, Kyiv, Glushkov ave., 4d,

e-mail: kafedraTK@unicyb.kiev.ua

*В статті запропоновано протокол багатоточкового потоку. Визначено вимоги до такого потоку та запропоновано схему функціонування протоколу. В якості прикладу застосування багатоточкового протоколу розглянуто реалізацію механізмів розпаралелювача та аналізатора подій в системі інтерактивного жестового інтерфейсу комунікації людина-комп'ютер.*

*Ключові слова: протокол багатоточкового потоку, інтерфейс жестової мови.*

*Requirements for protocol to develop parallelizer and events analyzer mechanisms in gestures interface system are discussed in the article. A multipoint stream protocol is proposed. The complete set and semantics of protocol methods were proposed. Security key usage and strategy for next key selection proposed. Also additional techniques for security strengthening proposed such as using of key shift field and fields shift field. The next security improvement is the usage of cover fields. The complete scheme of state of one point that used the multipoint protocol presented. A different ways of using multipoint stream protocol were demonstrated on examples of realising a parallelizer mechanism and events analyzer mechanism of sign language human-computer interface system. A problem for commands set of parallelizer and events analyzer mechanisms was discussed and the solution proposed. The advantage of multistreaming of multipoint stream protocol for command and result stream was also demonstrated.*

*Key Words multipoint stream protocol, sign language interface.*

Статтю представив д.ф.-м.н., проф. Анісімов А.В.

**1. Вступ та постановка задачі**

Схема інтерфейсу комунікації людина-комп'ютер на базі жестової мови та жестів, запропонована в (1), містить елемент розпаралелювач, призначенням якого є забезпечення одночасної передачі значень на обробку декільком процесам. Ці процеси є складовими віртуального процесу відповідно до (2) та керуються аналізатором подій, що також є складовим процесом. Серед сучасних технічних систем можливість одночасної передачі значень мають комп'ютерні мережі, побудовані за технологією broadcast. На сьогодні найбільш уживаною мережевою технологією є TCP/IP (3). Проте серед сучасних протоколів, реалізованих для мереж

у межах цієї технології, немає протоколу, для якого виконується такий набір вимог:

1. одночасна передача на декілька (можливо, сусідніх) комп'ютерів;
2. значення передаються потоком;
3. у потоці передаються окремі повідомлення;
4. є можливість приєднатися до потоку;
5. відсутність контролю отримування;
6. наявність системи захисту.

Вимоги 1, 2 та 3 відповідають задачі обробки жестової інформації.

Вимога 4 викликана потребою створення потоку на декілька отримувачів.

Вимога 5, традиційна для протоколу UDP, вважається недоречною для передачі потоку.

Причина полягає в неможливості передати потік значень без втрат та/або помилок навіть для невеликих за розміром потоків. Розбиття потоку на окремі повідомлення дозволяє передати їх безпомилково в більшості ситуацій, а у випадках помилок зберігати структуру потоку й правильно продовжувати його передачу. У випадку втрати отримувачем потоку відновлення його роботи відбувається завдяки виконанню вимоги 4. Такий підхід дозволяє відмовитися від технології TCP, що дуже доречно за умов, коли відновлений потік вже не буде актуальним із міркувань часу.

Вимога 6 необхідна для реалізації можливості роботи з незахищеними віддаленими мережами.

З вищесказаного випливає така постановка задачі: в межах технології TCP/IP розробити мережевий протокол, що відповідає вказаним вимогам. На базі запропонованого протоколу розробити механізми розпаралелювача та аналізатора подій в системі жестового інтерфейсу. В цій статті пропонується необхідний протокол та вказуються методи його реалізації.

## 2. Протокол багатоточкового потоку

Під потоком у цій статті будемо розуміти структуровану послідовність значень. Це означає: значення, що утворюють потік, об'єднані в блоки, що мають внутрішню структуру й об'єднані в послідовності. Передбачається, що блоки можуть бути різних розмірів, тому будемо називати їх повідомленнями. Послідовність повідомлень може утворювати складовий підпотік. В потоці завжди існує головний підпотік, в якому передаються команди на створення та знищення підпотоків. Головний підпотік також може використовуватися для передачі повідомлень. Обмін інформацією для програм організується з використанням вказаних структурних одиниць.

Одне повідомлення потоку при передачі в мережі розділяється на пакети, що передаються на нижчому мережевому рівні. Кожен такий пакет містить традиційні для протоколів сімейства TCP/IP поля – IP-адреси відправника та отримувача, числовий номер протоколу та розміру пакету. В якості

адреси отримувача можливе використання broadcast-адреси IP-підмережі. Це забезпечує одночасне отримання пакета кількома комп'ютерами. Використання адреси конкретного комп'ютера означає, що процеси-отримувачі виконуються в межах цього комп'ютера або на комп'ютерах після нього в розумінні топології мережі та маршруту передачі.

На відміну від протоколів TCP та UDP, в яких вказуються порти відправника та отримувача (3), у протоколі багатоточкового потоку вказується номер порту потоку. Використання одного й того самого номеру порту для всіх точок, під'єднаних до потоку, необхідне для забезпечення одночасної багатоцільової передачі значень.

Для забезпечення захисту інформації в кожен пакет додаються поля упаковки, нового ключа, тасування ключа та тасування полів.

Поле з контрольною сумою пакета присутнє та використовуються як зазвичай.

Для узгодженості з блочною структурою інформації додаються поля з номерами та розмірами повідомлення та пакету в ньому.

Поле з числовим ідентифікатором підпотіку вказує відповідний ідентифікатор. Для запобігання колізії на створення підпотоків вважається, що у випадку створення існуючого підпотіку точка приєднується до нього.

Повна структура пакета така:

1. IP-адреса відправника;
2. IP-адреса отримувача;
3. числовий ідентифікатор протоколу;
4. розмір пакету;
5. порт потоку.

Подальші поля закриваються ключем:

6. упаковка;
7. тасування ключа;
8. тасування полів;
9. новий ключ;
10. контрольна сума;
11. числовий ідентифікатор підпотіку;
12. номер повідомлення;
13. розмір повідомлення;
14. номер пакету повідомлення;
15. розмір пакету повідомлення;

16. пакет повідомлення;
17. упаковка.

Під з'єднанням в описуваному протоколі розуміється сукупність точок, під'єднаних до потоку інформації, що формується процесами, які назвемо учасниками потоку. Кожен учасник потоку може передавати та приймати інформацію, використовуючи потік. Захист від одночасної передачі забезпечується технологією опорної мережі. Наприклад, для мережі Ethernet це є детектування колізії.

Кожен учасник потоку передає новий ключ, що може бути використаний у потоці. Пропонуються такі стратегії вибору ключа:

1. ключ від точки з найменшою IP-адресою;
2. ключ від точки з найбільшою IP-адресою;
3. ключ від точки з найбільшим об'ємом переданої інформації.

Вибір стратегії покладається на програміста при вказуванні параметрів з'єднання. Під'єднання точки до потоку можливе лише за умови правильно вказаної стратегії вибору наступного ключа.

Як вже вказувалося, рішення щодо відсутності механізму відновлення цілісності потоку прийнято з огляду на жорсткі часові вимоги. Незручності цього підходу мають компенсуватися як зовнішніми чинниками на кшталт надійності мережі в розумінні мінімальних втрат пакетів, що передаються, так і внутрішніми, як наприклад, обробка лише частини повідомлень для отримання результату. У випадку втрати з'єднання точка або заново під'єднується до потоку, або переходить до виконання інших дій.

Передбачено такі інтерфейсні функції для реалізації протоколу:

1. create – створення точки;
2. listen – очікувати на утворення потоку;
3. connect – об'єднання точок в потік;
4. join – приєднати точку до потоку;
5. close – закрити потік;
6. disconnect – від'єднати точку від потоку;
7. read – читати повідомлення;
8. write – надіслати повідомлення;

9. keepalive – підтримка потоку;
10. new – створити новий підпотік;
11. delete – закрити підпотік.

Схему станів роботи точки потоку зображено на рис. 1.

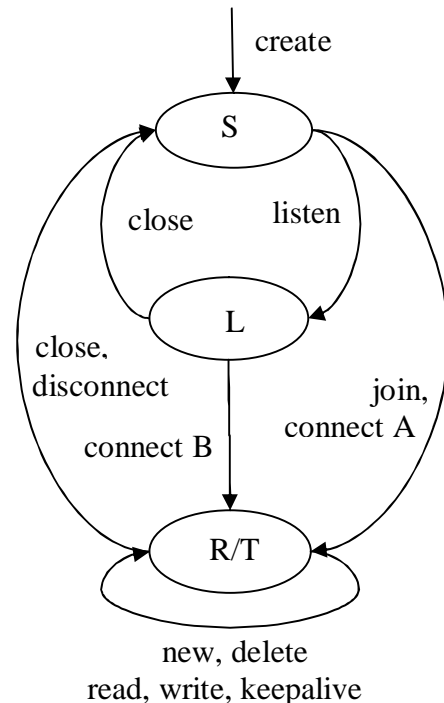


Рис. 1. Схема станів точки потоку  
В наведеній схемі використані такі позначення:

S – початковий стан,

L – стан очікування з'єднання,

R/T – робочий стан потоку,

create, listen, close, join, new, delete, read, write, keepalive – відповідні дії точки, описані вище,

connect A – активне формування спільного потоку з окремих точок,

connect B – пасивне приєднання до потоку зі стану очікування.

### 3. Механізми розпаралелювача та аналізатора подій

Механізм розпаралелювача має з'єднувати вхідний потік, що передає інформацію, та вузли модулів, що отримують інформацію(1). Оскільки вхідні потоки можуть бути різними й наперед зреалізованими та з визначеними інтерфейсами, то модуль розпаралелювача з'єднується з вхідним потоком за традиційними протоколами. Зазвичай це TCP-сесія. Модулі, що здійснюють обробку, та сам модуль розпаралелювача утворюють

багатоточковий потік. Для формування потоку модулі обробки утворюють точки та очікують доти, доки розпаралелювач не з'єднає ці точки в єдиний потік.

Аналізатор подій виконує роль менеджера віртуального процесу для модулів функціональної обробки (2). На підставі рішень аналізатора подій запускаються або зупиняються окремі модулі. Для реалізації такого керування множиною підпроцесів запропоновано утворити багатоточковий потік, що створюється аналізатором подій, після чого до нього під'єднуються підпроцеси окремих модулів. В утвореному потоці передаються команди керування модулями та результати роботи модулів. Для розмежування цих інформаційних потоків в межах багатоточкового потоку створюються підпотоки.

Описана реалізація накладає певні вимоги на синтаксис команд, що використовуються для керування модулями. Синтаксично однакові команди повинні виконуватися в модулях одночасно. Прикладом такої команди є команда «перейти до обробки

наступного повідомлення». Що стосується різних команд, то вони мають бути синтаксично різними. Модулі, що не мають таких команд, повинні сприймати ці команди як такі, що не викликають ніяких дій. Прикладами таких команд є команди встановлення параметрів окремих модулів.

#### 4. Висновки та подальші напрямки роботи

В результаті проведених досліджень було запропоновано протокол багатоточкового потоку, за допомогою якого реалізуються механізми розпаралелювача та аналізатора подій в системі інтерфейсу людина-комп'ютер на базі жестової мови та жестів.

Подальші дослідження будуть спрямовані на те, щоб реалізувати запропонований протокол, реалізувати на його основі модулі розпаралелювача та аналізатора подій, узгодити зовнішні функціональні модулі, створити загальну систему розпізнавання жестів та жестової мови та дослідити її роботу.

#### Список використаних джерел

1. *Крак Ю.В., Коваль Ю.В., Тернов А.С.* До розробки інтерактивного інтерфейсу моделювання та розпізнавання жестової інформації / Ю.В. Крак, Ю.В. Коваль, А.С. Тернов // Вісник Київського національного університету імені Тараса Шевченка Серія фізико-математичні науки. – 2014. – №4. - С.175-178.
2. *Крак Ю.В., Коваль Ю.В., Ставровський А.Б.* Віртуальний процес: означення та застосування в створенні системи жестового інтерфейсу / Ю.В. Крак, Ю.В. Коваль, А.Б. Ставровський // Вісник Київського національного університету імені Тараса Шевченка Серія фізико-математичні науки. – 2015. – №1. - С. 141-144.
3. *Marsic I.* Computer networks: Performance and quality of service/ I. Marsic // Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, New Jersey - 2013.

#### References

1. *KRAK IU.V., KOVAL IU.V., TERNOV A.S.* (2014) *To creation of interactive interface for sign information modeling and recognition.* Bulletin of Taras Shevchenko National University of Kyiv Series Physics & Mathematics. Vol. 4. p.175-178.
2. *KRAK IU.V., KOVAL IU.V., STAVROVSKIY A.B.* (2015) *Virtual process: definition and application for gestures interface system creation.* Bulletin of Taras Shevchenko National University of Kyiv Series Physics & Mathematics. Vol. 1. p.141-144.
3. *Marsic I.* (2013) *Computer networks: Performance and quality of service* Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, New Jersey.

Надійшла до редколегії 15.05.2015