

## ПІДХІД ДО СИНХРОНІЗАЦІЇ ДАНИХ КОРИСТУВАЧІВ, ЩО ВЗАЄМОДІЮТЬ У РЕАЛЬНОМУ ЧАСІ

Шемшур Я. С., Штогріна О.С.

*Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна*

*E-mail: yaroslav.shemshur@gmail.com*

### **An approach to synchronize customer data that interacts in real time**

The article provides a possibility to synchronize program objects between of real-time interaction users. A way to optimize the volume of traffic when using mobile devices to transmit a large number of small messages is described.

Більшість сучасних інформаційних технологій базуються на клієнт-серверній архітектурі. В її основу був покладений принцип повного відділення реалізації виконуваних функцій від інтерфейсу для доступу до них. Використання серверу з відповідною функціональністю, як сполучної ланки між користувачами, забезпечило швидкий та безпечний обмін інформацією [1]. Проте, на сьогоднішній день все більше виникає потреба обміну інформацією між всіма користувачами певного застосування в реальному часі. Також необхідно враховувати, що зазвичай використовуються мобільні застосування, що накладає обмеження пов'язані з передачею даних по мобільній мережі. Синхронізація даних користувачів в реальному часі затребувана в таких сферах як робототехніка, медицина, архітектура, та в багатьох застосуваннях, які несуть науковий або розважальний характер у тому числі на основі доповненої чи віртуальної реальності. Прикладом такої взаємодії можуть бути проекти в яких є необхідність одночасної взаємодії фахівців однієї предметної області: розробка архітектури присадибної ділянки, будинку, чи навіть великого торгового комплексу, керування технікою різного рівня, проведення навчання оперування чи взаємодії з медичинською технікою, використання графічного редактору, та ін.

Враховуючи те, що клієнти повинні постійно обмінюватися великою кількістю повідомлень з сервером, постає проблема мінімізації об'єму трафіку, особливо при використанні мобільних пристроїв. Також особливу увагу необхідно приділити безпеці передачі повідомлень користувачів, та забезпечити можливість використання необхідного методу шифрування. Як приклад в ігровій індустрії на сьогоднішній день існують програмні рішення, які вирішують проблеми синхронізації даних користувачів в реальному часі, але вони не повністю відповідають все більш зростаючим вимогам до швидкодії, об'ємів трафіку та можливості вибору альтернативних способів забезпечення безпеки. А саме:

«Photon Server» (<http://www.photonengine.com>), «Smart Fox Server» (<http://www.smartfoxserver.com>) та «Unity network» (<https://docs.unity3d.com/Manual/UNet.html>). Ці програмні рішення мають жорстко встановлені правила серіалізації/десеріалізації та використовують стандартні протоколи з великим об'ємом «службових» даних [2].

Отже актуальним завданням є розробка підходу до синхронізації даних користувачів, що взаємодіють в реальному часі, що дозволить підвищити швидкість обміну повідомленнями, зменшити об'єм трафіку та обирати відповідний спосіб захисту повідомлень, що передаються.

Для вирішення поставленого завдання запропоновано зменшити об'єм трафіку за рахунок мінімізації об'єму повідомлення та використання більш легкого протоколу передачі даних.

Для мінімізації об'єму повідомлення розроблено спеціальний метод серіалізації/десеріалізації об'єкта, що дозволяє серіалізувати/десеріалізувати тільки частину об'єкта, що змінилась, або зазначати тільки ідентифікатор вже заздалегідь описаного стану. Як приклад, створюється список анімацій об'єкта, з відповідними ідентифікаторами, самі ж анімації зберігаються на пристрої користувача. Для зміни анімацій об'єкта, передається тільки ідентифікатор анімації, яку необхідно відтворити.

Після порівняння та дослідження протоколів передачі даних UDP та TCP було виявлено, що протокол UDP дозволяє зменшити об'єм трафіку при передачі невеликих повідомлень до 4х разів. Але, на відміну від TCP, він функціонує без підтвердження та гарантії доставки повідомлень. Тому запропоновано передавати дані по протоколу UDP та використовувати розроблений додатковий програмний модуль на сервері, який би відповідав за підтвердження прийому повідомлень.

Протокол UDP функціонує на транспортному рівні моделі OSI, а це означає, що він не може відповідати за шифрування даних. Тому було запропоновано розробити програмний модуль, який буде відповідати за захист при передачі повідомлень. В даному випадку програмний модуль дозволяє використовувати необхідні алгоритми шифрування в залежності від вимог, висунутих до безпеки застосування. Також цей модуль відповідає за генерацію та пересилку користувачу динамічних ключів шифрування, які постійно оновлюються при взаємодії з сервером.

Отже при реалізації запропонованого підходу до синхронізації даних користувачів, що взаємодіють в реальному часі були розроблені програмні модулі прийому/відправки та шифрування/дешифрування повідомлень, серіалізації/десеріалізації об'єкта та обробки даних. На рис. 1 продемонстрований

обмін даними між користувачами у реальному часі при використанні запропонованого підходу.

Рішення запроповані в рамках підходу дозволяють мінімізувати об'єм кожного повідомлення, що в сукупності з передачею по протоколу UDP, суттєво зменшує загальний об'єм трафіку. Також зменшення об'єму повідомлення, підвищує швидкість обміну даними, за рахунок зменшення часу та обчислювальних ресурсів на шифрування/дешифрування повідомлення.

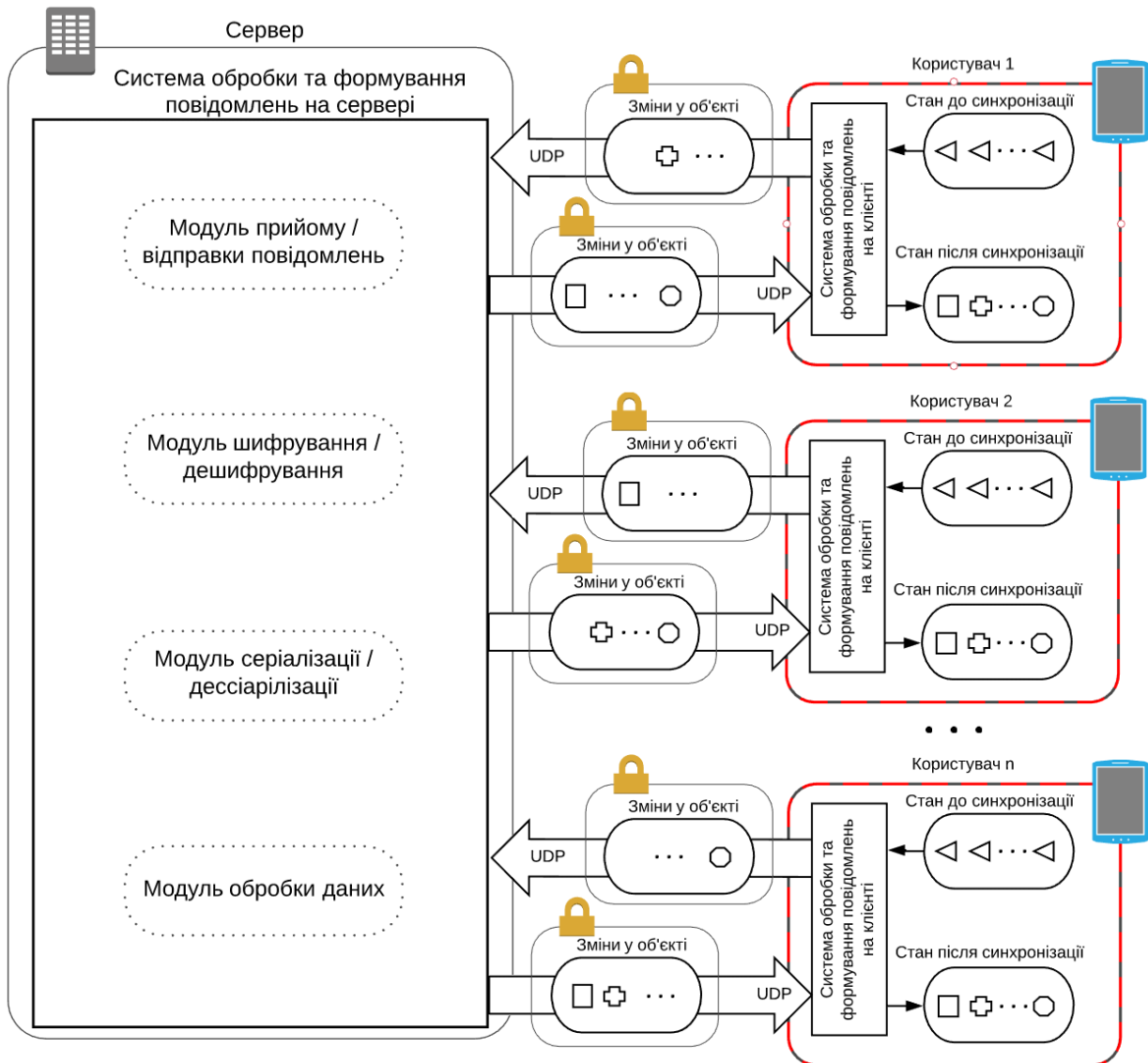


Рис. 1. Обмін даними між користувачами у реальному часі

### Література

1. Alex Berson Client/server architecture : New York - McGraw-Hill, 1996 — 20с.
2. Jason Edelman Network Programmability and Automation: Skills for the Next-Generation Network Engineer / Jason Edelman, Scott S. Lowe, Matt Oswalt Virginia Wilson — 359с.