

Н.О. СОЛОДКА, к.т.н., доцент

О.А. ЛЯШЕНКО, к.т.н., доцент

Державний вищий навчальний заклад «Український державний хіміко-технологічний університет», м. Дніпро

## Проектування та розробка клієнт-серверного додатку на основі однорангових мереж

Спроектовано та розроблено клієнт-серверний додаток для передачі даних в однорангових мережах. Запропоновано використання регулярних виразів в удосконаленому алгоритмі розбору рядків, який забезпечує значне зменшення часу передачі клієнтської інформації з одночасним додатковим її захистом.

### Постановка проблеми

Необхідність прискорення обміну великим обсягом даних та різного роду інформацією між користувачами, які знаходяться на відстані один від одного, викликало появу та бурхливий розвиток мережевих технологій.

На сьогодні однією з технологій передачі інформації є технологія, зміст якої точно передається англійським словосполученням peer-to-peer (P2P) networking або в українському еквіваленті — пірингові мережі [1]. Peer — це клієнт (приватна особа, людина), який, поєднуючись з іншими клієнтами, створює разом з ними, на перший погляд, тимчасову і хитку, однак, завдяки можливості підключення до неї десятків мільйонів людей, насправді потужну і непорушну мережу.

Комп'ютерні мережі поділяються на однорангові мережі та мережі з виділеним сервером [1]. Ця класифікація має принципове значення, тому що тип мережі характеризує її функціональні можливості.

Однорангові мережі — це комп'ютерні мережі, в яких не передбачено виділення спеціальних комп'ютерів, контролюючих адміністрування мережі. Кожен мережевий вузол виконує функції як клієнта, так і сервера. При вході в мережу кожен користувач самостійно надає в ній які-небудь ресурси у колективне користування і підключається до ресурсів, наданих в мережу іншими користувачами.

В останнє десятиріччя однорангові мережі набули виняткової популярності, тому що в порівнянні з класичною клієнт-серверної архітектурою, принципи побудови пірингових мереж забезпечують їм ряд переваг. До таких належать: масштабованість (в окремих точках мережі не створюється так званого «вузлого горлечка», оскільки обмін інформацією може відбуватися безпосередньо між кінцевими вузлами); стійкість (працездатність мережі зберігається при відключенні від мережі майже будь-якої кількості вузлів); приватність (дані користувачів можуть зберігатися, а обчислення проводитися безпосередньо на персональних комп'ютерах, без залучення третьої довіреної особи). Крім того, однорангові мережі досить прості у встановленні і налагодженні та набагато дешевші, ніж мережі з виділеним сервером [2].

Ці переваги обумовлюють основні області застосування однорангових мереж, серед яких — децентралізований файлообмін. Зокрема масштабованість, яку вимагає обмін великими файлами, досягається за рахунок прямого обміну даними.

Але, крім описаних вище переваг, пірингові мережі мають ряд недоліків, що потребують вирішення. Серед них: необхідність забезпечення захисту персональних даних у відповідності з чинним законодавством, складність адміністрування кожного комп'ютера окре-

мо, погіршення продуктивності при спільному використанні ресурсів. В одноранговій мережі, де немає централізованих сервісів, кожен користувач — потенційний порушник. В таких умовах забезпечення захисту будь-якого роду є досить складним завданням, яке постає при спробі досягнення стандартних цілей безпеки.

Актуальність теми роботи визначається необхідністю підвищення якості обслуговування в однорангових мережах передачі потокових даних. Для вирішення цього завдання в роботі пропонується застосування удосконаленого алгоритму розбору (так званого парсингу) рядків між клієнтським і серверним додатком на основі регулярних виразів. Використання такого алгоритму забезпечить прискорення передачі даних користувача та їх додаткову безпеку.

Отже, об'єктом дослідження є однорангова мережа для передачі потокових даних в реальному режимі часу.

Предмет дослідження — теоретичні і практичні завдання підвищення якості функціонування однорангових мереж передачі потокових даних.

### Аналіз останніх досліджень та публікацій

Істотний внесок у дослідження та моделювання передачі потокових даних в однорангових мережах внесли J.F. Buford, H. Yu, Eng Keong Lua [3]. Васильєв Д. С. досліджував можливості поліпшення якості передачі відео в деревоподібній P2P мережі з алгоритмом ARQ прикладного рівня [4]. Також йому належить авторство в розробці алгоритмів передачі потокових даних на прикладному рівні в пірингових мережах [5].

### Формулювання мети дослідження

Метою роботи є підвищення якості обслуговування в однорангових мережах передачі потокових даних з зірковою топологією за рахунок застосування алгоритму парсинга рядків між клієнтським і серверним додатком на основі регулярних виразів, а також забезпечення безпеки даних користувача.

Для досягнення цієї мети в роботі виконані наступні етапи дослідження:

1. Розробка алгоритму передачі даних в однорангових мережах з зірковою топологією, який дозволяє підвищити якість передачі потокових даних та їх безпеку.
2. Порівняльний аналіз ефективності запропонованого та існуючих алгоритмів керування передачею потокових даних в однорангових мережах.
3. Розробка експериментального клієнт-серверного додатку на основі однорангових мереж, як для обмеженого числа користувачів, так і для відкритого доступу.

### Виклад основного матеріалу

В роботі пропонується розробка клієнт-серверного додатку, який використовує технологію пірингових мереж з можливістю рестрації  $n$ -ої кількос-

ті користувачів і серверу, котрий може бути встановлений на комп'ютері у одного з користувачів. Напрямок застосування програми полягає в передачі даних між клієнтами за технологією peer-to-peer.

Проект складається з двох додатків: клієнтського та серверного; обидва працюють по протоколу управління передачею даних — Transmission Control Protocol/Internet Protocol (TCP/IP). Даний протокол забезпечує встановлення з'єднання між двома пристроями і надійну передачу даних. При цьому протокол TCP/IP контролює оптимальний розмір переданого пакета даних, здійснюючи нову передачу даних при збої здійснення операції. IP-протокол або адресний протокол є основою всієї архітектури передачі даних та забезпечує доставку сітьового пакету даних за потрібною адресою.

В даний час очевидним є той факт, що успішне та ефективне функціонування інформаційних систем і технологій визначає якість їх проектування. Щоб зробити процес проектування більш ефективним та зручним на ринку програмного забезпечення налічується більше 300 різних CASE-засобів. Їх застосування істотно спрощує процес розробки програмного забезпечення, безпосередньо проектування інформаційних систем та підвищує їх якість.

Структурне проектування клієнт-серверного додатку, який пропонується в роботі, здійснювалось за допомогою CASE-засобу верхнього рівня BPwin, який підтримує методології: IDEF0, IDEF3 і DFD. Детальний опис методологій, етапів побудови діаграм та рекомендації щодо їх використання наведені в роботах [6—8].

Процес проектування системи починається зі створення контекстної діаграми — діаграми найбільш абстрактного рівня опису системи в цілому, яка містить визначення суб'єкта моделювання.

Суб'єктом моделювання в даному випадку є клієнт-серверний додаток. Сервер виступає як сполучна ланка між клієнтами. Його задача полягає в тому, щоб забезпечити користувачів списками доступних файлів один одного та мати змогу обмінюватися ними.

При завантаженні або роздачі файлу, сервер виконує операцію з'єднання двох користувачів безпосередньо за технологією P2P. Для підключення до сервера потрібно мати вихід в Інтернет або бути підключеним до локальної/WiFi мережі, а також знати IP-адрес серверного додатку.

Контекстна діаграма роботи клієнт-серверного додатку, яка представлена на рис. 1, описує глобальну функцію системи, що моделюється.

Кожна з чотирьох сторін функціонального блоку «Робота клієнт-серверного додатку» має своє певне значення (роль), при цьому:

- верхня сторона має значення «Управління» (Control);
- ліва сторона має значення «Вхід» (Input);
- права сторона має значення «Вихід» (Output);
- нижня сторона має значення «Механізм» (Mechanism).

Аналізуючи створену діаграму, можна сказати, що вхідними даними для здійснення роботи додатку є з'єднання клієнта з сервером.

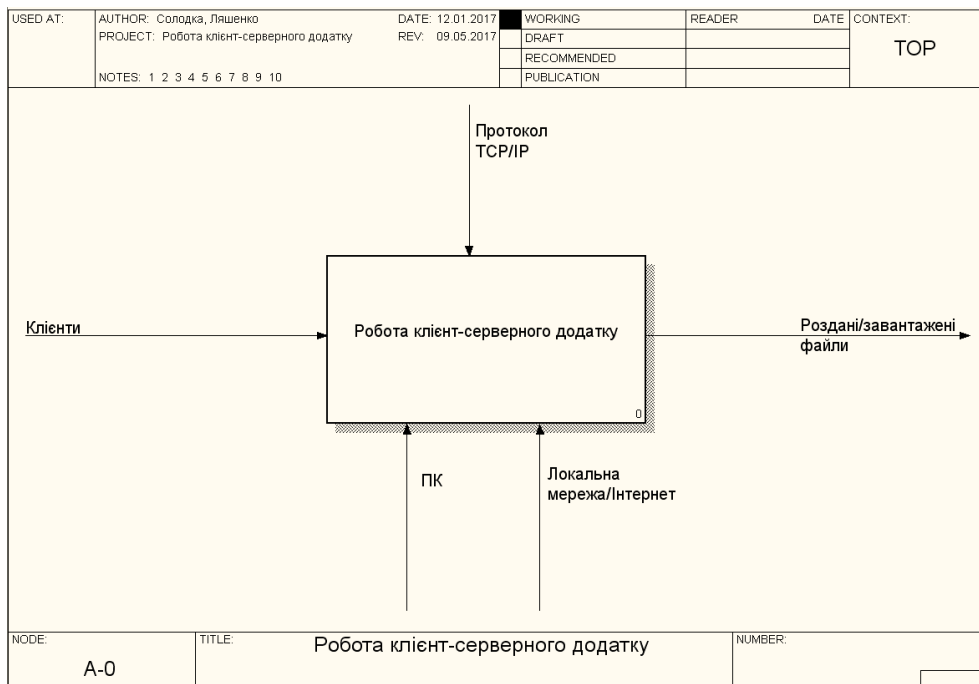


Рис. 1. Контекстна діаграма «Робота клієнт-серверного додатку»

Результатом діяльності є роздані або завантажені файли.

В якості механізмів, завдяки яким здійснюється процес роботи додатку, виступають ПК, наявність локальної мережі та можливості виходу в Інтернет для з'єднання з сервером або між клієнтами.

Управління системою, а саме обмін даними сервера з клієнтами, відбувається за допомогою протоколу передачі даних TCP/IP.

Для більш детального огляду роботи клієнт-серверного додатку контекстна діаграма була декомпонована за допомогою методології IDEF3 (рис. 2), яка

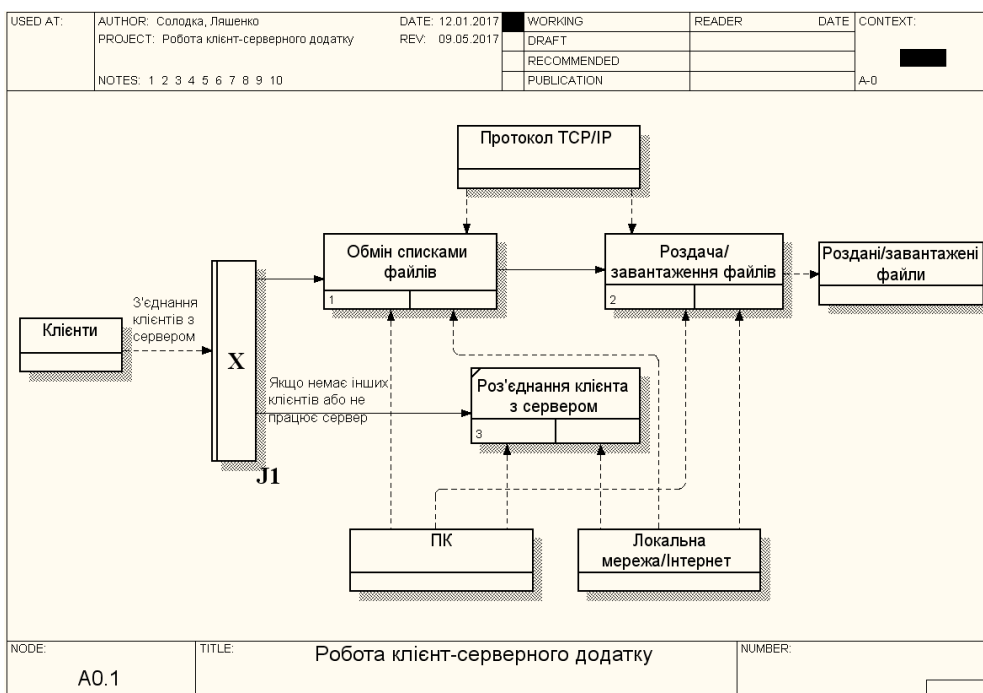


Рис. 2. Діаграма декомпозиції «Робота клієнт-серверного додатку»

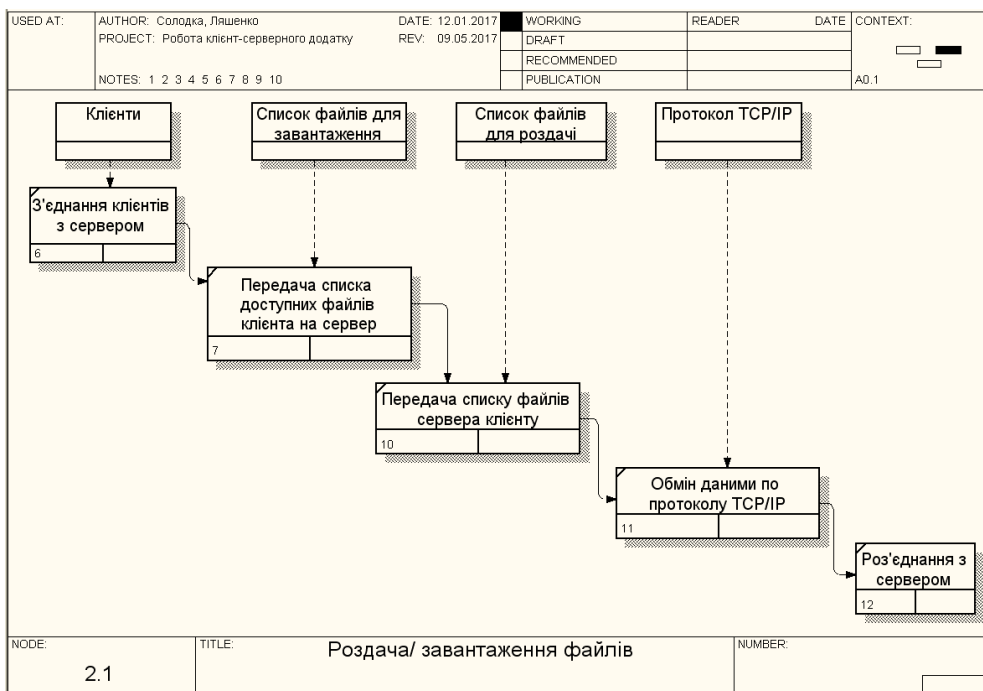


Рис.3. Діаграма декомпозиції блоку «Роздача/завантаження файлів»

дозволяє описати ситуацію, коли процеси виконуються в певній послідовності, а також описати об'єкти, що беруть участь спільно в одному процесі [7].

Методологія IDEF3 — використовується для опису логіки взаємодії інформаційних потоків.

Діаграма декомпозиції, яка зображена на рисунку 2, має моделювання наступного характеру: для з'єднання клієнтів та сервера необхідно ПК, локальна мережа або Інтернет; у разі відсутності інших клієнтів

або непрацюючого серверу, відбувається роз'єднання клієнта з сервером; в іншому випадку відбувається обмін списками файлів між клієнтами та сервером і роздача або завантаження файлів по протоколу TCP/IP.

На рисунку 3 представлено графічний опис завантаження та роздачі даних у вигляді діаграми декомпозиції блоку «Роздача/завантаження файлів» з використанням методології IDEF3.

Після того, як клієнт з'єднався з сервером, від-

бувається передача списку доступних файлів клієнта для роздачі на сервер. Наступним етапом виконується передача списку файлів сервера клієнту для завантаження. Після чого відбувається завантаження або роздача файлів між клієнтами по протоколу TCP/IP. Останнім етапом є процес роз'єднання з сервером.

На рисунку 4 зображена діаграма декомпозиції блоку «Обмін списками файлів». Як видно з рисунку,

процес обміну списками файлів починається з реєстрації клієнта. Потім виконується з'єднання клієнтів за технологією P2P, обмін даними за протоколом TCP/IP, обмін доступних файлів між клієнтом і сервером для завантаження/роздачі та роз'єднання з сервером. При цьому для виконання операції обміну даними виконується передача списку файлів для завантаження та списку файлів для роздачі.

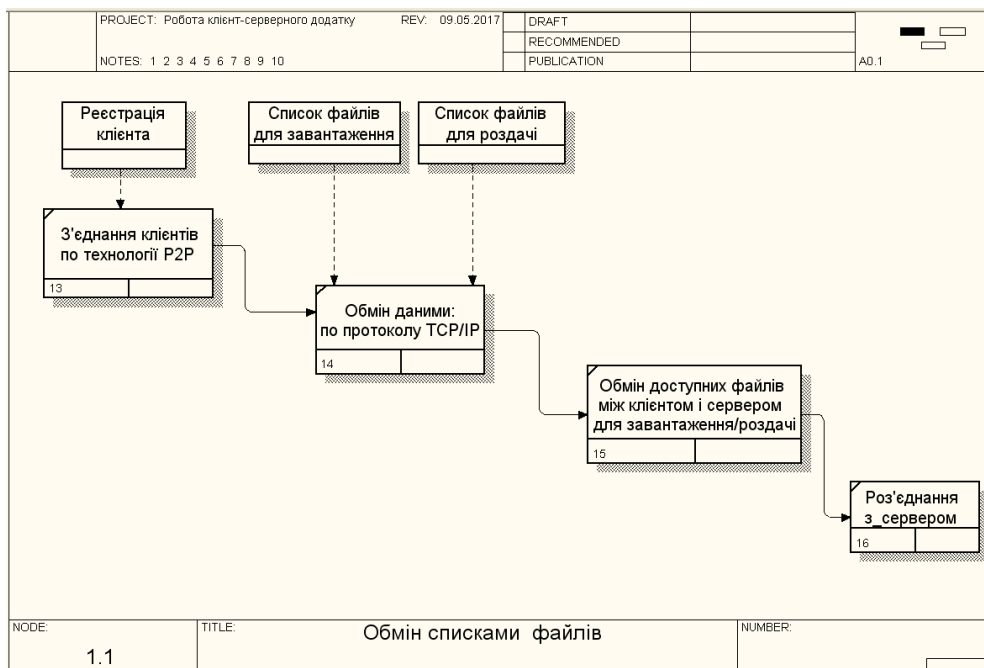


Рис. 4. Діаграма декомпозиції блоку «Обмін списками файлів»

Таким чином, отримані результати проектування із застосуванням засобів структурного аналізу є основою для розробки програмного забезпечення.

В якості засобів розробки для реалізації клієнт-серверного додатку використані мова програмування C++, середовище розробки – Microsoft Visual Studio 2013.

Розроблений додаток має дві частини — клієнтську («P2P Client») та серверну («Server P2P»). Для можливості підключення клієнтам необхідно мати вихід в Інтернет або бути підключеними до локальної/WiFi мережі. Після запуску серверної частини додатку клієнтські програми можуть до неї підключитися. В якості інтерфейсу користувача серверного додатка використовується консоль. Вона містить тільки інформаційну складову, яка виводиться та зберігається на протязі всієї роботи сервера. При цьому на консоль (рис. 5) виводиться інформація про статус користувача, а саме: який клієнт підключився, який відключився, скільки клієнтів в даний час online, а також серверні повідомлення.

Для підключення до сервера потрібно мати вихід в Інтернет або бути підключеним до локальної/WiFi мережі, а також знати IP-адресу серверного додатка. На рисунку 6 зображено інтерфейс цього додатка, кожен елемент якого описано згідно нумерації:

1. Кнопка Menu. При натисканні кнопки з'являється контекстне меню з пунктами: Connect (для введення IP-адреси та номера порту сервера), Settings

(для вибору директорій з файлами для роздачі та завантаження), Exit (для виходу з додатку).

2. Кнопка About. При натисканні кнопки з'являється вікно, що містить коротку інформацію про додаток.

3. Текстове поле для введення імені файлу.

4. Колонка, яка містить імена доступних файлів.

5. Колонка, яка містить формат доступних файлів.

6. Колонка, яка містить розмір доступних файлів.

7. Кнопка Find First. При натисканні кнопки здійснюється пошук файлу за його назвою, яка була введена в текстове поле.

8. Кнопка Next. При кожному натисканні кнопки здійснюється перехід на наступні файли, які відповідають критеріям пошуку.

9. Робоча область виведення списку доступних файлів.

10. Рядок стану, в якому виводиться статус з'єднання з сервером у вигляді: «Connection: YES» та «Connection: NO».

11. Кнопка Connect with виконує з'єднання або роз'єднання клієнта з сервером.

12. Кнопка Update List. При натисканні кнопки оновлюється список доступних файлів для завантаження.

13. Колонка, яка містить інформацію про обсяг файлу, що завантажується, у відсотках: 100% означає, що файл був повністю завантажений.

```

C:\Windows\system32\cmd.exe
SERVER
UUU - Users status info.
+++ - New user connection.
--- - User disconnected.
!!! - Server messege.

***Waiting for connection...***
+++ - Gena [192.168.1.41] new connect!
UUU - 1 user(s) online
!!! - List of files received successfully!
+++ - Гена-ПК [127.0.0.1] new connect!
UUU - 2 user(s) online
!!! - List of files received successfully!
--- Гена-ПК [127.0.0.1] disconnect
UUU - 1 user(s) online
--- Гена-ПК [192.168.1.41] disconnect
UUU - No Users online
+++ - Гена-ПК [127.0.0.1] new connect!
UUU - 1 user(s) online
!!! - List of files received successfully!
+++ - Gena [192.168.1.41] new connect!
UUU - 2 user(s) online
!!! - List of files received successfully!

```

Рис.5. Консоль серверної частини додатку

14. Колонка, що містить значення розміру файлів, які завантажуються.

15. Колонка, що містить імена файлів, які завантажуються.

16. Робоча область виведення списку файлів, які завантажуються.

Для завантаження файлу необхідно подвійне натискання мишею на його імені. Стан процесу скачування файлу відображається в правій частині клієнтського додатка (рис. 6) з інформацією про статус завантаження.

Серверна сторона в якості порту використовує номер 666, клієнтська — 667 відповідно. Різні номери портів потрібні для того, щоб на одному комп'ютері мали змогу працювати одночасно як клієнтський додаток, так і серверний. Дані порти є вільними у всіх операційних системах Windows. Також необхідно зазначити, що можна змінювати номери портів, як клієнта, так і на серверному додатку. У разі зміни номера порту сервера або його IP-адреси необхідно повідомити клієнтів про ці зміни.

З метою забезпечення додаткового захисту інформації у клієнтському додатку була застосована спеціальна логіка при роздачі файлів, яка полягає в наступному. При отриманні клієнтом запиту на роздачу файлу, він зчитує з сервера всі дані про файл, який треба відправити: ідентифікатор, ім'я, формат і розмір файлу. Зазначена інформація надходить до клієнта у так званому «сирому» вигляді рядка з сепараторами. В розробленому клієнт-серверному додатку розбір рядків (парсинг) виконується із застосуванням регулярних виразів – потужного, гнучкого і ефективного засобу обробки текстів [9]. При цьому кожен елемент рядка програма порівнює зі своїми масками, які зберігаються в пам'яті. Спочатку відшукується ідентифікатор файлу. У разі його знаходження, починається порівняння назви файлу, потім формату і останнім кроком — перевірка розмірів файлів. Якщо хоча б один елемент інформації не співпадає, відбувається відправлення повідомлення FileNotFound на сервер, який у свою чергу повідомляє про це клієнта-завантажувача. В разі знаходження файлу програма зчитує з пам'яті повний шлях до цього файлу, а не підс-

тавляє надісланий іншим клієнтом або шукає його знову, і починає з'єднуватися безпосередньо з клієнтом за технологією P2P для відправки файлу. Таким чином, дана логіка процесу відправки файлу забезпечує додатковий захист від проникнення і витоку даних у користувача.

Запропонований алгоритм розбору «сирих» рядків, які надходять з сервера, із застосуванням регулярних виразів реалізовано у функції ChangeList, яка належить до класу AllClients. Клас AllClients був створений для клієнтського додатка та призначений для зберігання і обробки інформації про всіх клієнтів, які підключені до сервера.

Запропоноване та використане в роботі удосконалення алгоритму розбору рядків більш детально розглянуто у [10].

Було проведено порівняльний аналіз ефективності запропонованого та існуючого алгоритму керування передачею потокових даних в однорангових мережах. Для цього створена функція SimpleChangeList, яка реалізує традиційний алгоритм розбору рядка без застосування регулярних виразів.

Порівняння відбувалося на основі показнику швидкості парсинга (табл. 1). Тестування швидкості парсингу рядка проводилося на ПК з наступними характеристиками: процесор Intel Core i7-2600K CPU 3.40 GHz; ОЗП 8Гб; відеокарта NVIDIA GeForce GTX 570; ОС Windows 7 64-розрядна.

Проаналізувавши дані таблиці 1, можна зробити висновок, що швидкість розбору рядків за допомогою функції ChangeList приблизно на 100% більша, ніж при застосуванні функції SimpleChangeList.

Для налагодження та перевірки працездатності клієнт-серверного додатку використовувався режим передачі користувачьких файлів різного розміру через мережу з шести персональних комп'ютерів (табл. 2), з'єднаних між собою як за допомогою витой пари, так і по бездротовому з'єднанню за допомогою WiFi. Чотири комп'ютери були підключені до локальної мережі, два — по WiFi-мережі.

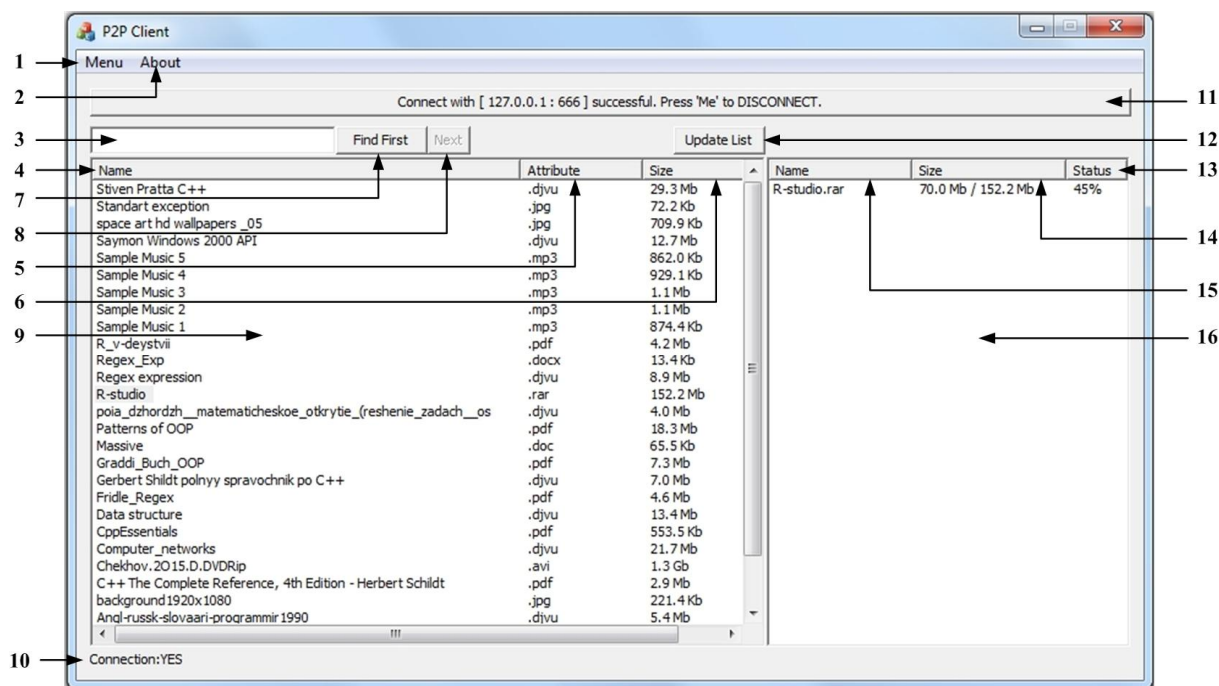


Рис. 6. Інтерфейс клієнтської частини додатку

Таблиця 1. Порівняльна швидкість парсинга «сирого» рядка

Кількість символів	Швидкість парсинга, сек	
	ChangeList	SimpleChangeList
800	0,001	0,049
1200	0,001	0,072
1600	0,002	0,107
2000	0,003	0,132
3000	0,004	0,315
4000	0,005	0,325
8000	0,006	0,493
12000	0,007	0,702

Таблиця 2. Технічні характеристики комп'ютерів

№ ПК	ОС	Процесор	Відеокарта	ОЗП
1	Windows 8	Intel Core i5 6200U CPU 2.40 GHz	AMD Radeon R7 M340	8 Гб
2	Windows 10	Intel Core i5 3230M CPU 2.60 GHz	AMD Radeon HD 8750M	4 Гб
3	Windows 7	Intel Core i7 2600K CPU 3.40 GHz	NVIDIA GeForce GTX 570	8 Гб
4	Windows 7	Intel Core i7 2600K CPU 3.40 GHz	NVIDIA GeForce GTX 570	8 Гб
5	Windows 7	Intel Core i7 2600K CPU 3.40 GHz	NVIDIA GeForce GTX 570	8 Гб
6	Windows 7	Intel Core i7 2600K CPU 3.40 GHz	NVIDIA GeForce GTX	8 Гб

	GHz	570
--	-----	-----

В процесі тестування передавалися файли різного розміру від 1 Кб до 3,2 Гб. Середня швидкість передачі файлу становила 5,87 Мб в секунду. У ході тестування були виправлені всі помилки роботи клієнтського і серверного додатків.

**Висновки та перспективи подальших досліджень**

У запропонованому клієнт-серверному додатку передача даних реалізована на основі однорангових мереж із зменшенням навантаження на серверну частину з одночасним збільшенням швидкості передачі даних за рахунок використання регулярних виразів. Реалізовано додаткову безпеку даних клієнтів та принцип спілкування командами між клієнтським і серверним додатком.

В перспективі подальших досліджень є розробка графічного інтерфейсу для сервера, створення рангу градацій для користувачів, використання декількох TCP-портів для завантаження та роздачі даних. Розроблені при створенні клієнт-серверного додатку алгоритми парсинга рядків та підвищення безпеки витоку даних можуть використовуватися програмістами у створенні власних програмних продуктів. Здобуті результати можна використовувати на підприємствах, на яких існує необхідність обміну даними в мережі.

**ЛІТЕРАТУРА**

1. Таненбаум Э. Компьютерные сети. 5-е изд. / Э. Таненбаум, Д. Уэзеролл. – СПб.: Питер, 2012. – 960 с.
2. Минко В. Обратная сторона пиринговых сетей – сложность обеспечения безопасности [Электронный ресурс] / В. Минко // Журнал сетевых решений/LAN. – 2015. – № 4 – Режим доступа до журн.: <https://www.osp.ru/lan/2015/04/13045700/>.

3. Buford J.F. P2P Networking and Applications: підручник / J.F. Buford, H. Yu, Eng Keong Lua; Elsevier Inc.– USA: 2009. – 396 p.
4. Васильев Д.С. Экспериментальное исследование качества передачи видео в древовидной P2P сети с алгоритмом ARQ прикладного уровня / Д.С. Васильев, А.В. Чунаев, А.В. Абилов // Т-Comm Телекоммуникации и транспорт, Том 8, № 1. – 2014. – С. 10-15.
5. Васильев Д.С. Разработка алгоритмов передачи потоковых данных на прикладном уровне в сетях беспилотных летательных аппаратов.: автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.12.13 „Системы, сети и устройства телекоммуникаций” / Васильев Д.С. – Ижевск, 2013. – 18 с.
6. Маклаков С.В. ВРWin и ERWin. CASE-средства разработки информационных систем / С.В. Маклаков // М.: ДИАЛОГ-МИФИ, 1999. – 256 с.
7. Моделирование и анализ IDEF-технологии: практикум / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М. Финансы и статистика, 2002. – 192 с.
8. Аксенов К.А. Работа с CASE-средствами ВРwin, ERwin: учебник / К.А. Аксенов, Б.И. Клебанов – Е.: ГОУ ВПО Уральский государственный технический университет УГТУ-УПИ, 2004. – 50 с.
9. Фридл Дж. Регулярные выражения, 3-е издание / Дж. Фридл. – СПб.: Символ, 2008. – 598 с.
10. Солодка Н.А. Применение алгоритма разбора строк в клиент-серверном приложении на основе регулярных выражений / Н.А. Солодка, Г.Г. Книжник // Наукова Україна: Збірник статей II Всеукраїнської наукової конференції (з міжнародною участю) 23-24 травня 2016 року. – Дніпропетровськ: Акцент ПП, 2016. – С.208-212.

пост. 29.06.2017