

УДК 004

О.Чумак, магістр гр. КН-22М-2

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КОМУТАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ЦЕНТРАЛІЗОВАНОЇ СЛУЖБИ МИТТЄВОГО ОБМІНУ ПОВІДОМЛЕННЯМИ

У статті розроблено програмне забезпечення, яке призначено для системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Метою розробки є дослідження та програмна реалізація системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Об'єктом дослідження є процес комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Предметом дослідження є методи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. У сучасному цифровому середовищі, що швидко розвивається, обмін миттєвими повідомленнями став невід'ємною частиною нашого життя, об'єднуючи людей по всьому світу в реальному часі. Незалежно від того, чи це невеликий груповий чат між друзями чи масова групова розмова в межах спільноти, потреба в спілкуванні в реальному часі є всюди. Розробка програми чату в реальному часі вимагає ретельного розгляду кількох ключових вимог. Ці вимоги допоможуть переконатися, що програма є ефективною, масштабованою та здатною забезпечувати чудову взаємодію з користувачем. Ось основні вимоги:

1. Спілкування в реальному часі: основна вимога полягає в забезпеченні двостороннього зв'язку між користувачами в реальному часі. Це означає, що нові повідомлення мають доставлятися миттєво без необхідності ручного оновлення.

2. Постійність повідомлень: повідомлення слід зберігати та отримувати з бази даних для історичних цілей. Це дозволяє користувачам отримувати доступ до своїх старих повідомлень чату навіть після виходу з системи або зміни пристрою.

3. Масштабованість: створюйте програму для обробки зростаючої бази користувачів і обсягу повідомлень.

4. Підтримка групових чатів: реалізуйте функції як для невеликих групових чатів, так і для великих каналів спільноти з тисячами учасників.

5. Розширення підтримки: дизайн системи повинен залишатися відкритим для майбутніх розширень, дозволяючи повторне використання існуючих компонентів і додавання нових для впровадження нових функцій.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

- Дослідження системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

- Програмна реалізація системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

Об'єктом дослідження є процес комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

Предметом дослідження є методи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями.

Методи дослідження базуються на методах теорії телеграфіку, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Основні завдання для месенджера:

- текстовий чат користувач – користувач, зазвичай це лікар – лікар або лікар – медсестра;

- різні повідомлення користувачів, наприклад, лікаря про появу результатів аналізів або прибуття пацієнта до приймального відділення;

- текстовий чат у контексті пацієнта або випадку лікування, коли лікар може поставити якісь уточнюючі питання іншим лікарям з прив'язкою до лікування або пацієнту;

- передача файлів у повідомленнях;

- робота з історією повідомлень (пошук, перегляд, позначки про прочитання).

В перспективі:

- різні сценарії повідомлення, наприклад, повідомлення лікаря при зміні статусу талона на запис до поліклініки, розсилки по посадах, структурних підрозділах;

- «пейджер» – push-повідомлення лікаря на його особистий мобільний пристрій, якщо лікар не підключений до внутрішньої мережі;

- аудіо-відео конференції (консиліум);

- телемедичні консультації, у тому числі за участю пацієнта, авторизованого через Держпослуги.

Основні користувачі системи повідомлень на цьому етапі – лікарі та середній медперсонал. Пацієнти тут не задіяні, за винятком у перспективі їхньої участі в телемедичних консультаціях, авторизувавшись через Держпослуги.

Коли йдеться про якісь системи повідомлень, «олдфаги» згадують IRC та ICQ. Якщо треба «модно та молодіжно» – мова заходить про Discord та Slack. Шанувальники приватності беруть Matrix. Решта використовують Telegram. Начебто бери та користуйся. Але для наших цілей це все абсолютно не застосовується: медична інформація, з якою працюють лікарі, – це такий забористий коктейль із персональних даних та медичної таємниці, що потребує особливих підходів, зокрема:

- практично вся робота йде у захищеному контурі, куди немає доступу стороннім сервісам;

- персональні дані та інша чутлива інформація повинна зберігатися та оброблятися до;

- небажано використовувати якихось іноземних постачальників.

Загалом усе це різко обмежує набір можливих рішень.

А чого, власне, хочеться від месенджера:

- відкритість – Відсутність vendor lock in, відкритий код;

- контроль – можливість розгорнути self-hosted інсталяцію;

- наявність реалізацій (клієнтів) під основні мови (платформи), які у нас: Java, PHP, Node.js, Python;
- стабільність – технологія має пройти фазу «хайпу»;
- розвиток – технологія має бути «мертвою»;
- шифрування – добре, але не в першу чергу, вся робота йде у захищеному контурі;
- інтеграція – можливість вбудовування у існуючі системи, зокрема, авторизація та список користувачів;
- розширюваність – певний підхід до створення розширень у протоколі/ПЗ без «глобальних милиць».

Десь тут ми почали розуміти, що Телеграм, Дискорд та інший Слак нас не врятують. Потрібно переходити до області Open Source. Завдання ускладнювало те, що потрібно було інтегруватися з нашими існуючими системами та сервісами, а також була потрібна передача різної специфіки з нагоди лікування (лікар, терміновість, тип діагнозу тощо). Можна було б придумати свій формат повідомлень – обмінюватися json-ами та передавати всю необхідну специфіку в полях, але хотілося не втрачати можливості роботи з якимись стандартними клієнтами без наших доробок для полегшення інтеграції сторонніх модулів.

Безумовно, одним із можливих шляхів було залишити все як є та продовжувати розвивати власне рішення. Тим більше, що на той момент вже окрім чатів була в якомусь вигляді реалізована підтримка аудіоконференцій. Альтернативним напрямом пошуку став перехід від готових рішень до області протоколів.

Побіжний пошук показав активний розвиток різних децентралізованих протоколів, наприклад, Matrix, Signal. Але за низкою параметрів, зокрема, можливість роботи з історією це нам не підходило. Щось стало відвертою екзотикою (OSCAR). Або більше належало до категорії «месенджер» ніж протокол (Mattermost). І тут хтось згадав про XMPP. Насправді у нас вже був досвід використання XMPP, але як корпоративний месенджер. Якраз у той період, коли ICQ вже перестала бути популярною, а щось просунуте ще не набуло потрібної популярності. Згодом, вже в «наш час» ми вдруге намагалися його використати, але незважаючи на цікаві фішки, які там з'явилися, наші технічні фахівці не змогли (або не захотіли) до ладу все налаштувати і XMPP програв гонку якомусь платному рішенню.

JID

Jabber Identifier будується за тим же принципом, що адреса електропошти: ім'я@домен. Може бути записаний у короткій формі ім'я@домен (bare JID) або у повній (full JID) ім'я@домен/ресурс. Ресурс служить для того, щоб можна було розрізнити кількох клієнтів, підключених до одного облікового запису. У кожного клієнта ресурс має бути унікальним. Тоді ми можемо вибирати надіслати повідомлення лише одному клієнту або всім одразу. JID може бути не тільки у користувача, а й у чат-кімнати, передплати і т.д. Для ресурсу вводиться поняття «пріоритету» – якщо повідомлення буде відправлено на короткий JID, то воно буде доставлене тому клієнту, пріоритет якого вищий (або всім, якщо пріоритет у всіх однаковий).

Станза (строфа)

Закінчений елемент XML-потоків, який містить певну керуючу інформацію:

- інформація про присутність (Presence) – інформаційні пакети спеціального виду, які містять інформацію про те, чи підключений в даний момент певний JID до мережі Jabber, а також передає його статус, статусне повідомлення та пріоритет;
- IQ (Info/Query) – особливий вид стансів, що реалізує механізм типу "запит-відповідь". Інтерпретація IQ-стансів дозволяє «сутності» зробити запит та отримати відповідь від іншої «сутності». Тип даних, що передаються у запиті чи відповіді визначає простір імен (namespace) дочірнього елемента щодо IQ;
- Message – використовується для обміну повідомленнями між користувачами.

Ростер (список контактів)

Розбитий на групи список Jabber-адрес ваших співрозмовників (контактів). Зберігається на сервері та передається клієнту на запит. Сервер також обробляє запити на додавання, видалення контакту зі списку та зміни групи для конкретного контакту.

ХЕР (розширення)

XMPP Extension Protocol – розширення протоколу XMPP. Наприклад, XEP-0045 – розрахований на багато користувачів чат, XEP-0084 – підтримка аватарок користувачів, XEP-0107 – статус користувача (user mood). ХЕР описують як якісь базові речі (XMPP Core), і безліч просунутого і дуже цікавого функціоналу.

Взагалі, розширення – одна з найцікавіших особливостей XMPP, коли, використовуючи цеглини, описані вище (різні станси), ми описуємо необхідний нам функціонал. За бажання можна зробити і власне розширення. На даний момент налічується близько двох сотень діючих ХЕР.

Використання XMPP

Для всіх цих випадків можна виділити одну картину (особливо характерну для великих порталів): швидкий старт сервісів, використовуючи XMPP, а потім, коли вступає в гру комерційна складова і завдання прив'язати користувача до порталу, вже народжуються якісь власні рішення, можливо, що залишаються у своїй масі, заснованими на XMPP.

Список компаній та рішень вселяє, завдання «заробляти з користувача» перед нами не стояло, і ми вже були готові бігти і робити все на XMPP. Але тут з'ясувалась одна особливість: для XMPP необхідно розглядати не тільки протокол, але переважно сервер і клієнта, що його реалізують. А все тому, що набір реалізованих розширень (тих ХЕР) від сервера до сервера можуть відрізнятися.

Вибираємо сервер

Найчастіше згадуваними серверами XMPP є (у дужках – мова реалізації):

- Ejabberd (Erlang).
- Prosody (Lua).
- Openfire (Java).

Коли ви читаєте про десятки та сотні тисяч користувачів, яких тримає один XMPP-сервер, швидше за все йдеться про Ejabberd. Але ми одразу розуміли, що можливі доопрацювання, а фахівців з Erlang серед нас не було. Тому вибір ліг на Openfire від компанії Igniterealtime, до речі, автора одного з найпопулярніших XMPP-клієнтів для Android – Smack.

Деталі нашої реалізації

Для оптимізації роботи з нашим веб-додатком на PHP реалізували відправку повідомлень через плагін з REST API – інакше щоразу авторизуватися виходить накладно за часом та ресурсами. Додаткова фішка плагіна – підтримується надсилання повідомлень до json, включаючи наші додаткові поля:

Крім того, повідомлення в цьому плагіні складаються в чергу – додатковий плюс для масштабування.

Повідомлення ми зробили через кімнати (груповий чат) – бот відправляє повідомлення у потрібну кімнату і всі, хто до неї входить, отримують повідомлення. Зазвичай кімнати створюються за принципом "одна кімната – одне відділення". Це виявився найшвидший і найпростіший спосіб для реалізації.

Загальні враження щодо XMPP та Openfire

XML-природа протоколу нехай надмірна, але сувора і зручна.

ХЕР описують багато «смачних» речей, але треба уважно дивитися, що реалізовано для конкретних клієнтів та серверів. Список Openfire: <http://download.igniterealtime.org/openfire/docs/latest/documentation/protocol-support.html>. Для нас загалом цей список виявився достатнім.

Поняття «ресурсу» («пристрою») – може бути застосовано дуже широко, наприклад, у нас як «пристрій» може виступати бічна панель повідомлень для веб-програми, основне вікно чату в тому ж веб-додатку, мобільний пристрій, програма – "пейджер" і т.д.

До переваг Openfire можна віднести:

- активну розробку;
- багато готових плагінів;
- хороші можливості для кастомізації: за допомогою плагінів та розширень можна налаштувати авторизацію, обробку пакетів, маршрутизацію та багато іншого.

З недоліків:

– не дуже успішно реалізований механізм плагінів, що реалізують своє REST API. Очевидно, автори спочатку не особливо розраховували на таке застосування, тому вийшло те, що вийшло;

- відсутнє автоматичне чищення історії в кімнатах – видаляємо скриптом із БД;
- при старті Openfire підвантажується ВСЯ історія по ВСІМ кімнатах – призводить до різкого зростання споживання пам'яті, вирішилося обмеженням глибини історії;
- за замовчуванням кімнати, що не використовуються, видаляються. Довго шукали в чому причина, поки не знайшли, що це регулюється опцією «Disable MUC room unloading for this service» у властивостях служби групового чату. Тут же можна налаштувати після кількох днів кімната, що не використовується, буде видалена, а також завантажувати чи ні всі кімнати при старті.

Окрім цього, для нас певною проблемою стало розгортання сервісу на регіонах – початкові варіанти та особливості інфраструктури вимагали застосування ручних налаштувань, і, на жаль, людський фактор дався взнаки. Після налаштування і контейнери допрацювали, стало набагато простіше.

Якщо потрібно швидко підняти корпоративний централізований месенджер або інтегрувати його до існуючого продукту – XMPP та Openfire чудовий варіант для старту. Чат, груповий чат – все працює "з коробки".

Потрібно уважно дивитися які XEP реалізує сервер і клієнт.

Загалом XMPP – це ближче до фреймворку, коли сервер (і клієнт) реалізують багато всяких цікавих штук, але потрібно докласти певного зусилля для того, щоб це стало закінченим рішенням.

Перспективи (плани на наступний етап):

- перехід на використання PubSub замість групового чату для повідомлень;
- «Пейджер» та push-повідомлення для лікарів – з відправкою знеособлених даних незахищеними мережами;
- авторизація через Держпослуги;
- інтеграція Openfire з Jitsi для аудіо-відео конференцій;
- інтеграція Openfire з Minio/IPFS для зберігання великих файлів, зокрема записів конференцій.

Розробка структурної схеми

Структурна схема розробленого програмного забезпечення системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями зображена на рисунку 1.

Розроблене програмне забезпечення складається з наступних блоків:

Клієнтська частина:

- Блок виводу GUI користувача програми.
- Блок авторизації.
- Блок обміну мовною інформацією.
- Блок обміну текстовими повідомленнями.
- Блок обміну відеоінформацією.

- Блок історії повідомлень по контактам.
- Блок пошуку.
- Блок відображення контактів.
- Блок налаштувань.
- Блок встановлення статусів.

Серверна частина:

- База даних користувачів.
- База даних інформації про користувачів.
- База даних оффлайн повідомлень.

У основі програмного забезпечення лежить протокол OSCAR та Jabber.

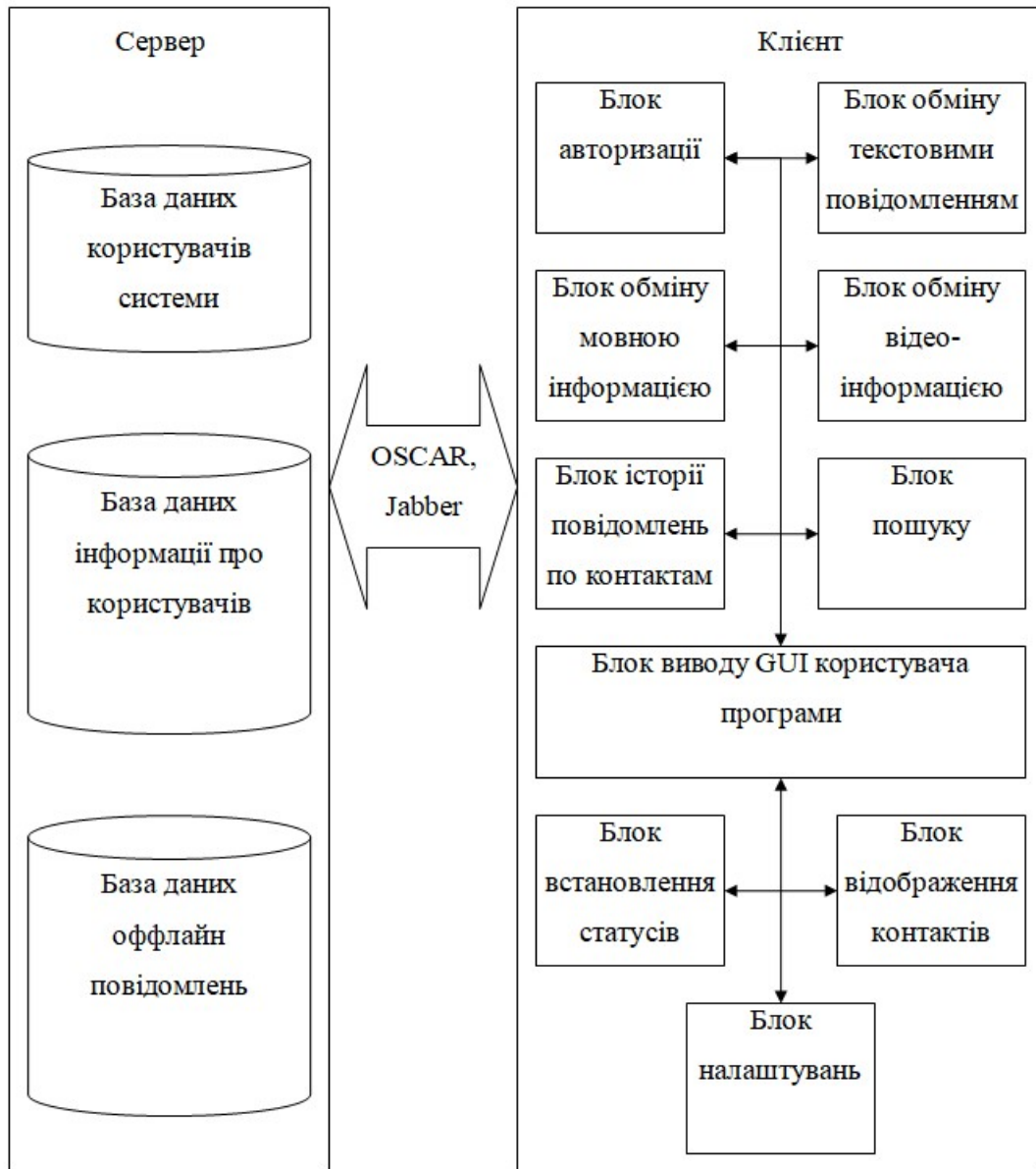


Рисунок 1 – Структурна схема системи

Месенджери змінили наше спілкування, це є факт. З одного боку ми завжди на зв'язку. Раніше ми могли забути про існування деяких знайомих та далеких родичів. Зараз WhatsApp і Viber регулярно нагадують про те, що та чи інша людина з'явилася в мережі.

Зворотний бік медалі у тому, що ми позбавляємо наші контакти природної енергії людського спілкування. Ми не докладасмо зусиль, щоб виявити любов та увагу до близьких. Завантажуючи картинку з поздоровленням, ми робимо це автоматично, без емоцій. Згадайте,

як ви вітали вчителя з професійним святом, колегу – з Новим роком, троюрідного брата – з днем народження. Чи не траплялося так, що ви копіювали вітання з інтернету, або завантажували стандартну картинку з побажаннями?»

Viber вивчив психологічні моделі поведінки людей у месенджерах, опитавши 2,8 тис. респондентів. Переважна більшість (88%) відзначили, що спілкуватися в листуванні легше, ніж телефоном чи реальному житті, оскільки співрозмовники не бачать друга, над відповіддю можна подумати, а висловлювання емоцій є стікери і емоди.

Також 45% опитаних вважають зручними спільні чати, стверджуючи, що так простіше залишатися на зв'язку з усіма близькими людьми, а 33% вважають за краще спілкуватися із співрозмовником у приватному діалозі.

Використання месенджерів – не добре і не погано. Вони заощаджують час, дозволяючи бути на зв'язку з тими, з ким ми раніше не могли розмовляти так часто. З іншого боку, вони позбавляють спілкування трепету, передчуття, інтимності. Під час самоізоляції месенджери дозволяли допомагати старшому поколінню, піклуватися, інформувати близьких про нововведення. Але ми ставали самотніми, годинами дивлячись у телефон.

З погляду психологічного здоров'я я наголошую на важливості живого спілкування, обіймів, прояву тілесності. Месенджери, швидше, повинні бути нам помічниками. Коли немає іншої можливості, ми робимо дзвінок у WhatsApp або Viber. Зідзвонюємося з тими, хто з тих чи інших причин далеко – колегами, закордонними родичами та друзями. Але якщо є можливість зустрітися, обійняти, торкнутися, поговорити до душі, краще зробити це».

Месенджери стали новим інструментом у бізнесі. За допомогою них зручно організовувати комунікацію з покупцями, які можуть поставити запитання про товар або послугу, та миттєво отримати відповідь від менеджера або чат-бота. Згідно з дослідженням, 62,5% компаній виставляють клієнтам рахунки через послуги обміну повідомленнями.

Месенджер-маркетинг допоможе бізнесу збільшити продажі. Наприклад, сервіс доставки їжі Elementaree з месенджерів отримує мільйонний виторг за пару годин.

У бізнесі через месенджер покупець зможе купувати товари, послуги та керувати замовленнями – для цього не потрібно буде переходити на окремий сайт. Зараз ця функція є в китайському WeChat, через який можна забронювати авіаквиток, оформити доставку їжі та не тільки.

Режим доповненої реальності в месенджерах схожий на маски в Instagram Stories і Snapchat

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями; Досліджена система комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями; На основі отриманих результатів досліджень створена програмна реалізація системи комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комутації користувачів на основі централізованої служби миттєвого обміну повідомленнями. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.
2. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D.,

- «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.
3. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.
 4. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
 5. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». Підводні технології, 2024, № 13, с. 28-35.
 6. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
 7. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макрофотографій». IV міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.
 8. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
 9. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
 10. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
 11. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп’ютерні науки та кібербезпека. № 4. С. 30-37. 2019.
 12. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп’ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.
 13. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISC’2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenco and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
 14. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139
 15. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.
 16. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.
 17. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.
 18. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп’ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.
 19. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67
 20. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції “Проблеми та перспективи розвитку IT-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНСУ. - 2014. - С. 240.
 21. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.
 22. Смірнов О.А., Дреєв О.М., Доренський О.П. «Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі. Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.
 23. Смірнов О.А., Доренський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.
 24. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.
 25. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп’ютерних мереж. – Кіровоград: КНТУ 2012. – 454 с