

**ПРОТОКОЛ ОРГАНІЗАЦІЇ З'ЄДНАННЯ МІЖ ВУЗЛАМИ
СИТУАТИВНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ****ПРОТОКОЛ ОРГАНИЗАЦИИ СОЕДИНЕНИЯ МЕЖДУ УЗЛАМИ
СИТУАТИВНОЙ КОМПЬЮТЕРНОЙ СЕТИ****PROTOCOL OF CONNECTION ORGANIZATION BETWEEN
SITUATIONAL NETWORK HOSTS**

Анотація. У статті надано детальний опис логічних і процедурних характеристик протоколу організації з'єднання між вузлами ситуативної комп'ютерної мережі. Проведено оцінку ефективності використання протоколу, розглянуто його програмну реалізацію. Зроблено висновки щодо доцільності використання запропонованого механізму.

Аннотация. В статье представлено детальное описание логических и процедурных характеристик протокола организации соединения между узлами ситуативной компьютерной сети. Проведена оценка эффективности использования протокола, рассмотрена его программная реализация. Сделаны выводы относительно целесообразности использования предложенного механизма.

Summary. This work presents a detailed description of the logical and procedural characteristics of the situational networks protocol, discussed its software implementation. Conclusions regarding the advisability of the proposed mechanism are made.

Різке збільшення доступності різних типів обчислювальної техніки для звичайних користувачів зробило повсякденним процес оброблення текстової, мультимедійної та інших типів інформації, спростило процес спілкування і обміну інформацією. Відзначимо, що будь-який обмін інформацією між абонентськими терміналами практично унеможливується за відсутності телекомунікаційної інфраструктури навколо них. При цьому кожний абонентський термінал (наприклад, ноутбук) являє собою досить потужний обчислювальний пристрій з декількома мережними інтерфейсами (Wi-Fi, Ethernet тощо). Теоретично кожен з таких терміналів можна використати для відправлення, отримання і трансляції інформаційних повідомлень, формуючи таким чином мережу без зовнішньої централізованої інфраструктури у ситуаціях, коли є необхідність в інформаційному обміні між користувачами [1].

Раніше був запропонований механізм ситуативних мереж, що вирішує проблему формування мережі з розподілених у просторі абонентських терміналів [2, 3]. Також була розроблена математична модель механізму [4], що дозволило визначити такі критичні характеристики, як таймери, а також оцінити залежність часових та ймовірнісних характеристик ситуативної мережі від її розміру.

Проте невід'ємною частиною реалізації будь-якого механізму обміну інформацією є специфікація алгоритмів його роботи та характеристик протоколів зв'язку. Така специфікація дозволяє провести перевірку працездатності механізму, уточнення його основних параметрів, а також перевірку відповідності математичної моделі діючому протоколу.

Таким чином, **метою** статті є розробка телекомунікаційного протоколу, що реалізує в собі діючу модель механізму ситуативних мереж.

Для успішної реалізації протоколу обміну інформацією [5] є доцільним створення наступних характеристик протоколу механізму ситуативних мереж у такому порядку: логічні характеристики протоколу, процедурні характеристики протоколу, програмна реалізація протоколу.

1. Логічні характеристики протоколу. На даному етапі розглядаються логічні характеристики протоколу ситуативних мереж, основою яких є формат повідомлень, що передаються між абонентськими терміналами на різних етапах інформаційного обміну. Відзначимо, що семантика полів кадрів каналного рівня залишається незмінною, за тією лише різницею, що наповнення полів «MAC (Media Access Control) – адреси» відправника та отримувача визначається способом адресації та відрізняється для різних типів повідомлень. На рис. 1 зображено спосіб інкапсуляції ситуативного повідомлення в кадрі протоколу каналного рівня, а також приклади інкапсуляції в кадри розповсюджених технологій каналного рівня.



Рисунок 1 – Інкапсуляція ситуативного повідомлення в кадр каналного рівня:

- 1) – загальний принцип інкапсуляції ситуативного повідомлення;
- 2) – приклад інкапсуляції ситуативного повідомлення в кадр протоколу Ethernet;
- 3) – приклад інкапсуляції ситуативного повідомлення в кадр протоколу Wi-Fi.

Формат ситуативного повідомлення складається із заголовка, що містить в собі керуючі інформаційні поля та поля даних, зображення якого показано на рис. 2.

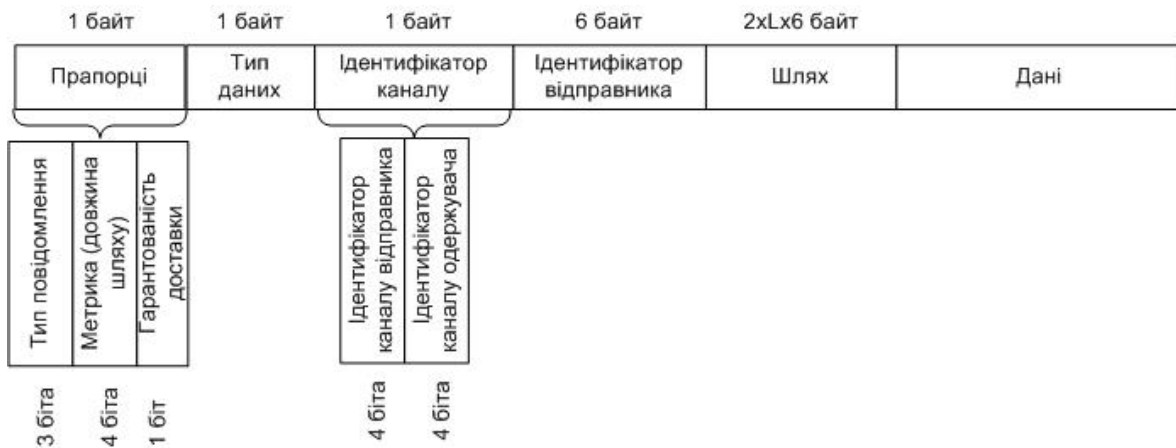


Рисунок 2 – Формат ситуативного повідомлення

Тип повідомлення (3 біта). Поле, яке визначає тип ситуативного повідомлення, що залежить від етапу обміну інформацією між абонентами. Основні типи повідомлень, що передбачено протоколом механізму ситуативних мереж надано в табл. 1.

Таблиця 1 – Типи ситуативних повідомлень

Тип повідомлення	Опис
0	Початковий запит
1	Відповідь на початковий запит
2	Запит на деталізацію необхідної інформації
3	Відповідь на запит про деталізацію необхідної інформації
4	Запит на створення віртуального каналу для передавання даних
5	Відповідь про підтвердження створення віртуального каналу
6	Передавання даних
7	Завершення з'єднання

Метрика або довжина шляху (4 біта). За допомогою цього поля визначається оптимальний шлях між відправником та отримувачем інформації. В даному випадку використовується кількість переходів між відправником та отримувачем. Слід зазначити, що вказане поле також відіграє роль обмежувача довжини заголовка ситуативного повідомлення, визначає кількість пар MAC-адрес, що записані в полі «Шлях».

Гарантованість доставки (1 біт). Поле, за допомогою якого визначається наявність необхідності підтверджувати доставку даних до отримувача. Реалізація самого механізму

гарантованості доставки всередині ситуативної мережі протоколом не передбачена та покладається на протоколи більш високих рівнів.

Тип даних (1 байт). Поле, що відповідає типу інформації (або типу сервісу), що передається ситуативною мережею. Являє собою поле «прапорців» щодо наявності різних типів ресурсів, кожний біт якого відповідає за наявність того чи іншого типу ресурсів. Можливі значення цього поля надані в табл. 2.

Таблиця 2 – Призначення прапорців поля «Тип даних» у заголовку ситуативного повідомлення

Прапорець	Опис
00000001	Обмін файлами (використання протоколу ftp, або подібного до нього)
00000010	Надання доступу до переферійних пристроїв (принтер, сканер тощо)
00000100	Отримання доступу до зовнішніх мереж, зокрема до мережі Інтернет через віддалений вузол мережі
00001000–10000000	Зарезервовані для майбутнього використання

Ідентифікатор каналу (1 байт). Складається з ідентифікатора каналу відправника і ідентифікатора каналу отримувача (по 4 біта кожний), що визначають унікальні номери процесів взаємодіючих усередині кожного абонентського терміналу. Зазначені поля можна порівняти з полями «порт відправника» та «порт отримувача» у протоколах UDP або TCP. Розмір ідентифікаторів каналів відправника та отримувача є цілком достатнім, враховуючи специфіку роботи ситуативних мереж.

Ідентифікатор відправника (6 байт). MAC-адреса одного з активних інтерфейсів (найменша за значенням), що представляє собою унікальний ідентифікатор для кожного абонентського терміналу. Це поле використовується для запобігання дублюванню записів у таблиці маршрутів абонентських терміналів.

Шлях. Поле змінної довжини. Довжина поля визначається кількістю абонентських терміналів на маршруті між відправником та отримувачем та складається з пар MAC-адрес. Кожна пара MAC-адрес являє собою один перехід між сусідніми абонентськими терміналами.

2. Процедурні характеристики протоколу. Під процедурними характеристиками розуміють правила виконання дій, передбачених протоколом взаємодії. Вказані правила можна умовно поділити на такі складові [7]:

- адресація;
- процедура оброблення повідомлення, що відноситься до типу протоколу на вузлі мережі;
- процедура забезпечення гарантованої якості обслуговування;
- набір операцій, що доступні сервісам або протоколам більш високих рівнів для взаємодії з даним протоколом. Набір основних операцій, що передбачені механізмом ситуативних мереж, наведено у табл. 3.
- процедура установалення з'єднання.

Таблиця 3 – Перелік основних операцій механізму ситуативних мереж

Назва операції	Опис
INIT (ініціалізація)	Ініціалізація ситуативних механізмів. Вибір мережних адаптерів для прийому та відправлення ситуативних повідомлень. Ініціалізація ресурсів, що можуть надаватися в користування іншим абонентам ситуативної мережі
REQUEST (запит)	Формування та відправка ситуативного повідомлення, що містить запит (початковий, на деталізацію або на створення віртуального каналу)
WAIT_REPLY (очікування відповіді)	Очікування відповіді/відповідей (що залежить від типу надісланого раніше запиту) на відправлений запит
WAIT_REQUEST (очікування запитів)	Очікування запитів (початкових, на деталізацію ресурсів або на створення віртуального каналу)
REPLY (відповідь)	Формування та відправка ситуативного повідомлення, що містить відповідь на конкретний тип запиту
DATA_SEND (відправка даних)	Формування та відправка ситуативного повідомлення, що містить дані
DATA_RECEIVE (прийом даних)	Очікування ситуативного повідомлення, яке містить вхідні дані
DISCONNECT (роз'єднання)	Завершення з'єднання

Розглянемо більш детально процедуру встановлення з'єднання та схему організації зв'язку між ініціатором запиту та будь-яким вузлом ситуативної мережі (рис. 3).

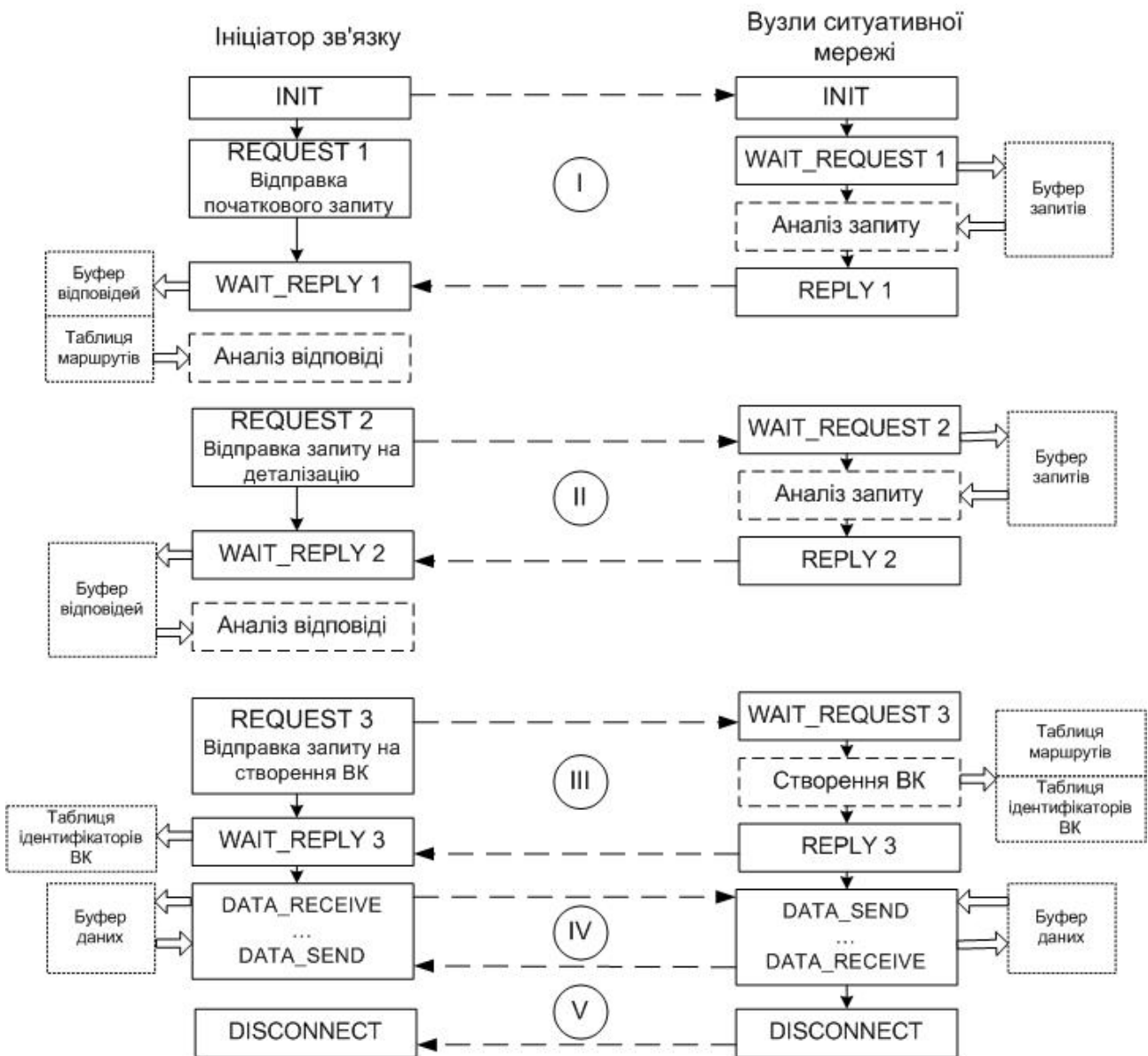


Рисунок 3 – Схема організації зв'язку між ініціатором запиту та вузлами ситуативної мережі

На рис. 3 римськими цифрами позначені етапи формування віртуального каналу (ВК): I – початковий запит, II – запит на деталізацію, III – установа віртуального з'єднання, IV – обмін даними та V – завершення з'єднання.

На схемі відображено порядок проведення операцій, а також взаємодію процедурних характеристик протоколу з інформаційними накопичувачами у вузлах ситуативної мережі. Операції, що здійснюються на рівні протоколу, показані на схемі у вигляді прямокутників. Процес обміну інформаційними повідомленнями між вузлами ситуативної мережі надано у вигляді пунктирних стрілок. Крім того на схемі подано інформаційні накопичувачі, за допомогою яких здійснюється обмін інформацією між різними процедурами протоколу, а також прикладним програмним забезпеченням.

Для коректної роботи процедурних характеристик як на джерелі, так і на інших вузлах ситуативної мережі на початковій стадії роботи протоколу необхідно визначити ряд параметрів. До таких параметрів відноситься інформація про мережні адаптери, що будуть задіяні для роботи протоколу ситуативних мереж усередині кожного вузла мережі, типи і перелік ресурсів, що

надаються користувачем для можливості доступу іншим користувачам, а також порядок доступу до відповідних ресурсів.

Після проведення процедури ініціалізації (операція INIT) джерело виконує операцію відправлення початкового запиту (REQUEST 1), при цьому формується та відправляється ситуативне повідомлення, з кодом відправлення 0, в якому розміщено інформацію про типи інформаційних ресурсів, до яких хоче отримати доступ прикладний процес джерела. Після відправлення запиту джерело переходить до режиму очікування відповіді на початковий запит (WAIT_REPLY 1). При отриманні відповіді на початковий запит (код повідомлення 1) джерело поміщує його в буфер відповідей, крім того для забезпечення можливості подальшого доступу до вузлів ситуативної мережі, що надіслали відповідь до таблиці маршрутів, додаються шляхи (маршрути) до кожного з вузлів. Аналіз відповідей, а також прийняття рішення про надсилання запиту на деталізацію здійснюється прикладними процесами на більш високому рівні, при цьому відповіді доставляються цим процесам через спеціальний буфер.

Вузли ситуативної мережі після проходження процесу ініціалізації переходять у стан очікування запиту (WAIT_REQUEST 1). Як тільки запит отримано, він розміщується у спеціальному буфері запитів. За необхідності прикладний процес може звернутися до цього буфера та отримати звітти запит для проведення аналізу та прийняття рішення про відправку відповіді про наявність відповідних типів інформаційних ресурсів. Якщо вузол сформує відповідь, він переходить до стану REPLY 1 (відправка відповіді на початковий запит).

На другому етапі, після отримання інформації щодо наявності ресурсів у ситуативній мережі джерело формує запит на деталізацію інформації (REQUEST 2), при цьому формується та відправляється ситуативне повідомлення з кодом 2, в якому розміщено інформацію про конкретний тип інформаційного ресурсу, а також шлях до конкретного вузла ситуативної мережі. Після відправлення запиту джерело переходить до режиму очікування відповіді на запит про деталізацію (WAIT_REPLY 2). При отриманні відповіді на запит про деталізацію джерело розміщує її в буфер відповідей. Аналіз відповідей, а також прийняття рішення про надсилання запиту на створення віртуального каналу (ВК) здійснюється прикладними процесами на більш високому рівні, при цьому відповіді доставляються цим процесам, як і у попередньому випадку, через буфер відповідей.

Дії, що проводять вузли ситуативної мережі на другому етапі, подібні до процесів, що відбуваються на першому етапі. Вузли ситуативної мережі переводяться до стану очікування запиту на деталізацію ресурсів (WAIT_REQUEST 2) і після отримання такого запиту та його аналізу прикладним процесом, вузол формує відповідь (REPLY 2) та відправляє його джерелу.

На третьому етапі джерело формує запит на створення ВК (REQUEST 3). Вказаний етап є схожим до попередніх етапів, проте в даному випадку всередині запиту вказується номер віртуального каналу джерела, а також конкретизується ресурс, до якого намагається отримати доступ джерело (наприклад, назва конкретного файлу), після чого джерело переходить до етапу очікування відповіді від вузла ситуативної мережі про створення віртуального каналу (WAIT_REPLY 3). При отриманні відповіді джерело розміщує його в буфер відповідей, а також формує відповідний запис в таблицю ідентифікаторів віртуальних каналів, що дозволяє проводити фільтрацію відповідних ситуативних повідомлень з усієї сукупності ситуативних повідомлень і подальшому розподілу даних, що відносяться до визначених ВК.

Вузли ситуативної мережі після відправлення відповіді на запит на деталізацію переходять у стадію очікування запиту на формування ВК (WAIT_REQUEST 3) та після отримання такого запиту та його аналізу прикладним процесом, вузол формує підтвердження (REPLY 3), в якому вказані номер ВК вузла ситуативної мережі, також вузол ситуативної мережі вносить інформацію про шлях до джерела запиту в таблицю маршрутів, а також запис про віртуальне з'єднання в таблицю ідентифікаторів віртуальних каналів.

На четвертому етапі джерело та вузол ситуативної мережі проводять обмін ситуативними повідомленнями, що містять корисні дані. При цьому отримувач ситуативних повідомлень, виконуючи операцію DATA_RECEIVE, запитує дані з буфера та передає їх процесу, що її викликав. У свою чергу відправник, проводить зворотну операцію DATA_SEND, приймає дані від процесів користувачів, формує ситуативне повідомлення та відправляє його отримувачу.

Відзначимо, що протокол ситуативних мереж не має вбудованого механізму контролю помилок, за винятком стандартних механізмів, що передбачені протоколами каналних рівнів. Також протокол ситуативних мереж не підтримує функціональності підтвердження про доставку у повному обсязі, протокол передбачає поле гарантованості доставки, яке може бути використане прикладними процесами.

Кінцевим етапом обміну інформацією між джерелом і вузлом ситуативної мережі є етап завершення з'єднання. Ініціатором цього етапу може бути будь-яка зі сторін, що бере участь в обміні інформацією. На вузлах ситуативної мережі виконується операція DISCONNECT, за якої формується та відправляється до мережі відповідне ситуативне повідомлення, а також розривається віртуальне з'єднання (видаляються відповідні записи з таблиці віртуальних каналів).

3. Оцінка ефективності протоколу. Ефективність механізму обміну інформацією, звичайно, визначається ефективністю протоколів, які він використовує, а також ефективністю його програмної реалізації.

Ефективність програмних модулів, що використовує механізм, можна оцінити лише шляхом експериментальних досліджень відповідної програмної реалізації. Для оцінки ефективності протоколу може бути використано показник ефективної пропускної здатності протоколу, який може бути визначено як середню швидкість передавання даних користувача або протоколу прикладного рівня [7]. Ефективна пропускна здатність протоколу напряму залежить від розміру службового навантаження, яке додає протокол до корисного навантаження користувача і, в свою чергу, не має сенсу в абсолютному вираженні, а набуває такого сенсу лише порівняно з іншими протоколами, з якими вказаний протокол порівнюється.

Протокол механізму ситуативних мереж у запропонованій реалізації логічних і процедурних характеристик представляє собою протокол, що забезпечує реалізацію третього та четвертого рівнів моделі взаємодії відкритих систем OSI/ISO. Таким чином, виявлення ефективної пропускної здатності протоколу ситуативних мереж доцільно порівняти з подібною ефективністю стека протоколів TCP/IP, що також реалізують вказані рівні моделі взаємодії відкритих систем.

Відзначимо, що розмір службового навантаження в протоколі ситуативних мереж є змінною величиною і залежить від кількості переходів між приймачем і передавачем інформації та визначається за формулою:

$$L_{\text{ситуат}} = L_{\text{незм}} + 2(N - 1)L_{\text{MAC}},$$

де $L_{\text{незм}}$ – незмінна частина заголовку ситуативного повідомлення, що дорівнює 9-ти байтам;

N – відстань (кількість переходів) між відправником і отримувачем інформації;

L_{MAC} – розмір уніфікованої MAC-адреси мережного адаптера, що дорівнює 6-ти байтам.

Стек протоколів TCP/IP такої залежності не має, при цьому розмір заголовка протоколу TCP складає 20 байт, і розмір заголовка IP також складає 20 байт. Таким чином, до кожного фрейма каналного рівня додається 40 байт службової інформації протоколів TCP/IP. Порівняння значень розмірів службових заголовків протоколу ситуативних мереж і стека протоколів TCP/IP у залежності від відстані (кількості переходів) між абонентами подано на рис. 4.

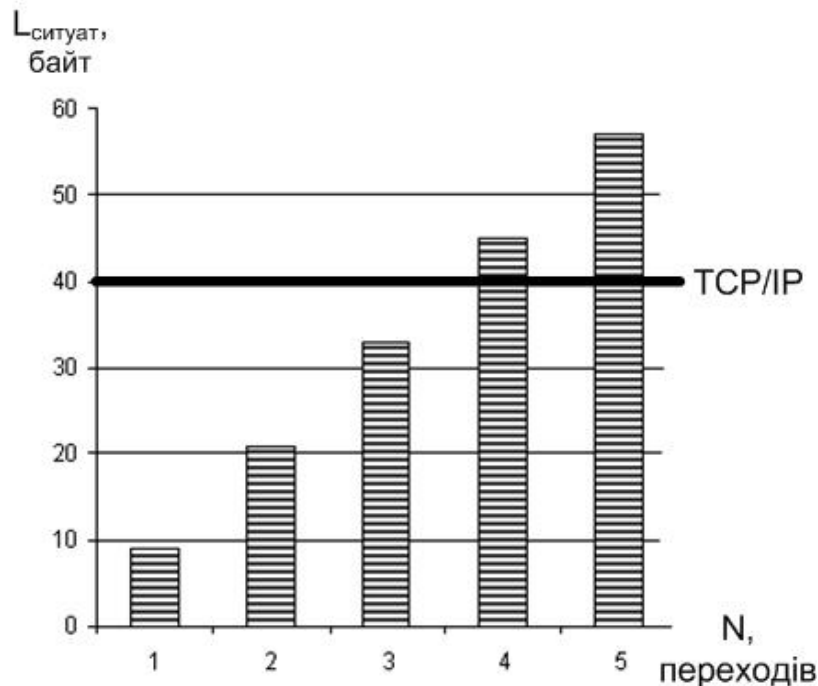


Рисунок 4 – Розміри службового заголовку протоколу ситуативних мереж у залежності від відстані між абонентами

Як видно з рис. 4, за умови, коли кількість переходів між відправником та отримувачем повідомлення є меншим або дорівнює трьом ($N \leq 3$), ефективність пропускну здатності протоколу ситуативних мереж є вищою за стек протоколів TCP/IP.

4. Програмна реалізація протоколу. На сьогоднішній день існують два основні шляхи програмної реалізації протоколу передавання інформації. Перший шлях являє собою програмний модуль, що забезпечує розширення функцій ядра операційної системи. Другий шлях являє собою реалізацію роботи механізму у вигляді бібліотеки функцій програмного інтерфейсу API (Application Programming Interface). Кожний зі шляхів має свої переваги та недоліки, проте основним недоліком створення модуля, що розширює функції ядра операційної системи, є те, що вказаний модуль буде значною мірою залежати від типу, а також версії операційної системи для якої створюється цей модуль. Таким чином, для забезпечення більшої уніфікації програмного забезпечення було обрано підхід, що передбачає створення бібліотеки програмного інтерфейсу. Також вказаний підхід дозволить спростити процес розповсюдження програмної реалізації механізму.

Загальні принципи взаємодії двох систем показано на рис. 5. Оброблення інформаційного потоку користувача являє собою низку дій, що відбуваються при реалізації різних програмних модулів. Порядок роботи з інформаційним потоком, джерелом якого є прикладний процес користувача, відбувається таким чином: прикладний процес при обробленні даних звертається до реалізації відповідної мережної служби, яка, в свою чергу, звертається до програмного модуля, що реалізує механізм ситуативних мереж. Програмний модуль, який реалізує механізм ситуативних мереж, являє собою основний потік програми, який забезпечує керування інформаційними накопичувачами (буферами), а також реалізує логіку дії протоколу ситуативних мереж. Після оброблення повідомлення основний потік програми формує кадр канального рівня і звертається до вихідного інтерфейсу, який передає його до мережі.

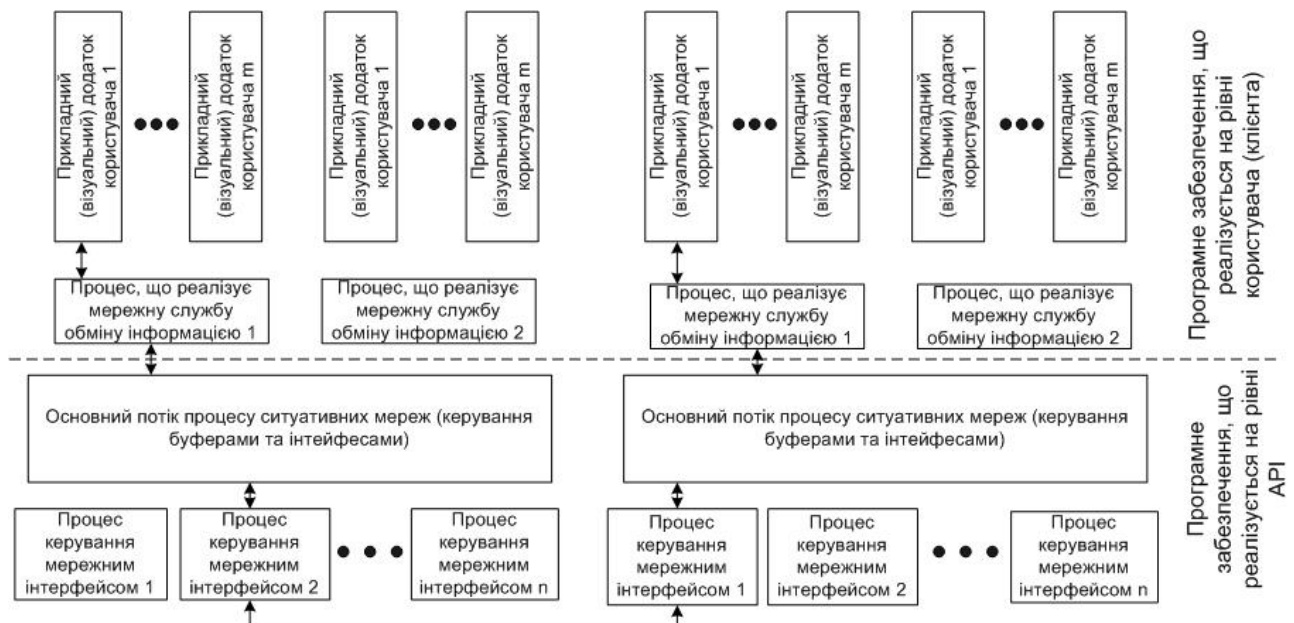


Рисунок 5 – Модель програмної взаємодії процесів ситуативних мереж

Процес керування мережним інтерфейсом являє собою відокремлений програмний потік, основним завданням якого є перехоплення вхідних фреймів канального рівня і якщо вказаний вхідний фрейм відповідає типу ситуативних повідомлень, то інформація, що входить до фрейма передається на оброблення основному потоку ситуативних мереж.

Взаємодія між основним потоком і процесами, що реалізують мережні служби, відбувається за допомогою інформаційних накопичувачів (буферів), інформація про які подана на рис. 3. При обміні інформацією між основним потоком і декількома процесами мережних служб відбувається за допомогою поля «Тип даних», що надано в заголовку ситуативного повідомлення. Вибір номера відповідного прикладного процесу користувача відбувається за допомогою поля «Ідентифікатор каналу», що також надане в заголовку ситуативного повідомлення.

Принцип взаємодії між прикладним додатком користувача і програмним процесом мережної служби визначається на етапі реалізації прикладного програмного забезпечення і не залежить від реалізації протоколу ситуативних мереж.

На закінчення слід відзначити таке:

– запропонована модель реалізації механізму ситуативних мереж надає можливість створення прикладного програмного забезпечення будь-якого рівня складності, який може передбачати реалізацію як однієї мережної служби (наприклад, тільки служби обміну файлами), так і універсальних програмних комплексів, що реалізують декілька мережних служб одночасно;

– зазначена модель дозволяє також розробити програмне забезпечення, що може використовуватись не лише для формування ситуативних мереж обміну інформацією в ситуаціях передбачених раніше [1], а також для формування інших типів мереж без централізованої інфраструктури (класичних безпроводових ad-hoc мереж, мереж кризових центрів тощо);

– в подальшому є цілком можливим реалізація механізму ситуативних мереж у вигляді модуля ядра будь-якої з існуючих операційних систем, що може допомогти забезпечити більшу конкурентну здатність відповідної операційної системи.

Література

1. Каптур В. А. Механізм керування розподіленими ресурсами в ситуативних мережах / В. А. Каптур, О. В. Степаненко // Наукові праці ОНАЗ ім. О.С. Попова. – 2010. – № 1. – С. 31–37.
2. Степаненко О. В. Алгоритми пошуку та передавання інформації в ситуативних мережах / О. В. Степаненко // Радіоелектронні і комп'ютерні системи. – 2010. – № 3. – С. 75–80.
3. Пат. 55941 Україна, МПК Н 04 L 12/28, Н 04 L 12/407, Н 04 L 29/02, Н 04 L 29/06. Спосіб встановлення з'єднання між вузлами ситуативної мережі / Каптур В. А., Степаненко О. В.; заявник і патентовласник Одеська національна академія зв'язку ім. О.С. Попова. – u201008903 ; заявл. 16.07.10 ; опубл. 27.12.10, Бюл. № 24.
4. Каптур В. А. Моделювання процесів передавання інформації в ситуативних мережах / В. А. Каптур, О. В. Степаненко // Наукові праці ОНАЗ ім. О.С. Попова. – 2011. – № 1. – С. 31–37.
5. Каптур В. А. Протокол організації віртуальних пірінгових каналів в локальних комп'ютерних мережах / В. А. Каптур // Наукові праці ОНАЗ ім. О.С. Попова. – 2006. – № 2. – С. 85–95.
6. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы. / В. Г. Олифер, Н. А. Олифер. – [4-е изд.]. – СПб.: Питер, 2010. – 943 с.
7. Воробийенко П. П. Принципы организации сетей с коммутацией пакетов: учеб. пособ. / Воробийенко П. П., Нечипорук О. Л., Струкало М. И. – Одесса: УГАС им. А.С. Попова, 2000. – 101 с.