

УДК 681.31

*Роман Анатолійович Ромащенко*

**Постановка проблеми та її зв'язок із важливими науковими і практичними завданнями. Формулювання мети статті**

Обслуговування на вузлах телекомунікаційної мережі будується по системі з обмеженою чергою та з пріоритетом. Як пріоритетні категорії виступають категорії якості обслуговування. Виходячи з цього, виникнення процесів масового обслуговування на входах та виходах вузлів – неминуче. Будь-яка система масового обслуговування характеризується показником імовірності відмови в обслуговуванні заявки [1], тобто події, при якій пакет, що прийшов для обслуговування знайде всі місця у чергах зайнятими. Збільшення імовірності відмови пакету в обслуговуванні призводить до зменшення живучості визначеного напрямку телекомунікаційної мережі. У TCP підтвердження відокремлені від дозволів на передачу даних, тобто управління вікном в TCP не прив'язане безпосередньо до підтверджень, як це зроблено в більшості протоколів передачі даних [2].

При нульовому розмірі вікна відправник не може посилати сегменти, за винятком двох випадків. По-перше, дозволяється посилати термінові дані, наприклад, щоб користувач міг знищити процес, що виконується на віддаленій машині. По-друге, відправник може послати 1-байтовий сегмент, просячи одержувача повторити інформацію про розмір вікна у очікуваному наступному байті. Стандарт TCP явно передбачає цю можливість для запобігання тупикових ситуацій у разі втрати оголошення про розмір вікна [3].

Відправники не зобов'язані передавати дані відразу, як тільки вони приходять від додатка. Також ніхто не вимагає від одержувачів посилати підтвердження якнайскоріше. Ця свобода дій може використовуватися для поліпшення продуктивності.

Розглянемо TELNET – з'єднання з інтерактивним редактором, що реагує на кожне натиснення клавіші. У гіршому разі, коли символ прибуває до TCP-сутності, що передає, то ця сутність створює 21-байтовий TCP-сегмент і передає його IP-рівню, який, у свою чергу, посилає 41-байтову IP-дейтаграму.

На приймаючій стороні TCP-сутність негайно відповідає 40-байтовим підтвердженням (20 байт TCP-заголовка і 20 байт IP-заголовка). Потім, коли

## МЕТОДИКА СТИСКУ ДАНИХ

редактор прочитає цей байт з буфера, TCP-сутність пошле оновлену інформацію про розмір буфера, пересунувши вікно на 1 байт вправо. Розмір цього пакету також складає 40 байт. Нарешті, коли редактор обробить цей символ, він відправляє назад ехо-камеру, що передається 41-байтовим пакетом. Разом для кожного введеного з клавіатури символу пересилається чотири пакети загальними розміром 162 байти. В умовах дефіциту пропускної спроможності ліній цей метод роботи небажаний.

Для поліпшення ситуації багато реалізацій TCP використовують затримку підтверджень і оновлень розміру вікна на 500 мс в надії отримати додаткові дані, разом з якими можна буде відправити підтвердження одним пакетом. Якщо редактор встигне видати ехо-камеру протягом 500 мс, віддаленому користувачеві треба буде вислати тільки один 41-байтовий пакет, таким чином, навантаження на мережу знизиться вдвічі.

Хоча такий метод затримки і знижує навантаження на мережу, проте, ефективність використання мережі відправником продовжує залишатися невисокою, оскільки кожен байт пересилається в окремому 41-байтовому пакеті. Метод, що дозволяє підвищити ефективність, відомий як алгоритм Нагля [4]. Пропозиція Нагля: якщо дані поступають відправникові по одному байту, відправник просто передає перший байт, а інші поміщає в буфер, поки не буде отримано підтвердження прийому першого байта. Після цього можна переслати усі накопичені в буфері символи у вигляді одного TCP-сегмента і знову почати буферизацію до отримання підтвердження відісланих символів. Якщо користувач вводить символи швидко, а мережа повільна, то в кожному сегменті передаватиметься значна кількість символів, таким чином, навантаження на мережу буде істотно знижено. Окрім того, цей алгоритм дозволяє посилати новий пакет, навіть якщо число символів в буфері перевищує половину розміру вікна або максимальний розмір сегменту.

Алгоритм Нагля широко застосовується різними реалізаціями протоколу TCP, проте іноді бувають ситуації, в яких його краще відключити. Наприклад, при роботі застосування X - Windows в Інтернеті інформація про переміщення миші пересилається на віддаленій комп'ютер. Якщо буферизувати ці дані для пакетної пересилки, курсор переміщатиметься ривками з великими

паузами, внаслідок чого користуватися програмою буде дуже складно, майже неможливо.

Ще одна проблема, що здатна значно понизити продуктивність протокола TCP, відома під ім'ям синдрому безглузлого вікна [5]. Сутність проблеми полягає в тому, що дані пересилаються TCP-сутністю великими блоками, але приймаюча сторона інтерактивного застосування зчитує їх посимвольно.

Девід Кларк (David Clark) запропонував заборонити приймаючій стороні відправляти інформацію про одnobайтовий розмір вікна. Замість цього одержувач повинен почекати, поки в буфері не накопичиться значна кількість вільного місця. Зокрема, одержувач не повинен відправляти відомості про нові розміри вікна до тих пір, поки він не зможе прийняти сегмент максимального розміру, який він оголошував при встановленні з'єднання, або його буфер не звільнився хоч би наполовину.

Окрім того, збільшенню ефективності відправки може сприяти сам відправник, відмовляючись від відправки занадто маленьких сегментів. Замість цього він повинен почекати, поки розмір вікна не стане досить великим для того, щоб можна було послати повний сегмент або, щонайменше, рівний половині розміру буфера одержувача. (Відправник може оцінити цей розмір по послідовності повідомлень про розмір вікна, отриманих ним раніше.)

У завданні позбавлення від синдрому безглузлого вікна алгоритм Нагля і рішення Кларка доповнюють один одного. Нагль намагався розв'язати проблему застосування, що посимвольно надає дані TCP-сутності. Кларк намагався вирішити проблему застосування, яке посимвольно отримує дані у TCP. Обидва рішення можуть працювати одночасно. Сутність їх полягає в тому, щоб не посилати і не просити передавати дані занадто малими порціями.

Приймаюча TCP-сутність може піти ще далі в справі підвищення продуктивності, просто оновлюючи інформацію про розмір вікна великими порціями. Як і відправляюча TCP-сутність, вона також може буферизувати дані і блокувати запит на читання READ, що поступає від застосування, до тих пір, поки у неї не накопичиться великий об'єм даних. Таким чином, зменшується кількість звернень до TCP-сутності і, отже, знижуються накладні витрати. Однак, такий підхід збільшує час очікування відповіді, що призводить до зменшення продуктивності мережі в цілому. Все це підтверджує актуальність роботи для галузі зв'язку.

Таким чином метою статті є розробка методики стиску даних, застосування якої на окінцевих вузлах дозволить уникнути синдрому безглузлого вікна.

### Виклад основного матеріалу

Сутність методики стиску даних полягає в одночасному використанні декількох систем числення для представлення однієї і тієї ж інформації, яка міститься в пакетах, що надходять

для обслуговування, з метою мінімізації заповнення вільної області пам'яті буферів накопичувачів.

Вихідні дані:

основи системи числення в залишкових класах  $P_b$

СМО вузла телекомунікаційної мережі, що має наступні характеристики:

час обслуговування заявок  $T_{обс}$  – випадкова величина, що має показовий розподіл з параметром  $\mu$ .

число каналів –  $n$ ;

число місць у чергах –  $m$ ;

ємність буфера накопичувача –  $ml$

потік пакетів – найпростіший (стаціонарний пуасонівський) із щільністю  $\lambda$ ;

Умови, допущення та обмеження:

потік, що розглядається – є потоком даних;

пакети мають однакову тривалість  $l$

інформація, що міститься в пакетах, які надходять для обслуговування, представлена в двійковій позиційній системі числення

значення основ системи числення в залишкових класах та їхня кількість вибираються виходячи з дотримання виконання умови:

$$z_i l \leq \prod_{i=1}^b p_b, \quad (1)$$

де  $z_i$  – кількість місць у черзі, що відводиться для запису пакетів,  $i$ -того пріоритету.

Реалізація методики стиску даних зводиться до послідовного проведення двох етапів:

стиску (кодування) вихідної інформації; декомпресії (декодування) стиснутої інформації та видачі її у канал обслуговування.

Етап 1. Стиск (кодування) вихідної інформації.

Стиск вихідної інформації полягає у послідовному виконанні ряду кроків, а саме:

Крок 1.1. Представимо інформацію, що міститься у  $k$  розрядах черги з мінімальним пріоритетом як деяке ціле число  $N$ , що записане у позиційній системі числення.

Отримане число  $N$  можливо представити у виді [6]:

$$N = a_{h-1}\pi_{h-1}\pi_{h-2}\dots\pi_2\pi_1 + a_{h-2}\pi_{h-2}\pi_{h-3}\dots\pi_2\pi_1 + a_2\pi_2\pi_1 + a_1\pi_1 + a_0, \quad (2)$$

де  $\pi_1, \pi_2, \dots, \pi_h$  – ряд цілих позитивних чисел, що в подальшому будуть називатися основами системи числення;

цифри  $a_{j-1}$  – сутність чисел  $0, 1, \dots, \pi_{j-1}$  ( $j = 1, 2, \dots, h$ ).

Обсяг діапазону чисел, що можуть бути представлені у даній системі числення, дорівнює [7].

$$\Psi = \pi_1\pi_2\dots\pi_h \quad (3)$$

Якщо ввести позначення  $p_i = \pi_1\pi_2\dots\pi_i$ , то вираз (2) може бути записаний у виді:

$$N = a_{h-1}p_{h-1} + a_{h-2}p_{h-2} + \dots + a_1p_1 + a_0. \quad (4)$$

Оскільки практично вся інформація передається по каналах зв'язку у виді

послідовності двійкових символів, то  $\pi_1 = \pi_2 = \dots = \pi_i = p = 2$ .

Тоді

$$N = a_{n-1}p^{h-1} + a_{n-2}p^{h-2} + \dots + a_1p + a_0, \quad (5)$$

звідки діапазон чисел, що можуть бути представлені у даній системі числення, дорівнює  $\psi = p^h = 2^h$ .

На рис. 1 наведений приклад запису числа  $N$  в двійковій системі числення у вигляді пакунку, що містить шістдесят розрядів.

Крок 1.2. Запишемо число  $N$  як множину, що має декілька елементів  $\alpha_b$ , розбивши  $h$  розрядів на  $b$  частин (рис. 2). Тобто  $N = \{\alpha_1, \alpha_2, \dots, \alpha_b\}$ .

Крок 1.3. Отримаємо число  $A = \{\alpha_1, \alpha_2, \dots, \alpha_b\}$ , що представлене в системі залишкових класів, умовно прийнявши значення елементів  $\alpha_1, \alpha_2, \dots, \alpha_b$  в якості сукупності залишків від ділення деякого цілого числа на послідовність заданих взаємопростих чисел (основ системи числення).

Крок 1.4. Переведемо число  $A$  із системи залишкових класів у позиційну систему числення згідно з алгоритмом, наведеним в [7].

Крок 1.5. Запишемо отримане число  $A$  у  $w$  розрядах ( $w$  – мінімальна необхідна кількість розрядів для запису числа  $A$ ).

Крок 1.6. Здійснюється перевірка виконання умови

$$\frac{h-w}{1} \geq 1. \quad (6)$$

У випадку виконання умови (6) у буфер накопичувача вузла телекомунікаційної мережі запишеться число  $A$ , у протилежному випадку – застосування методики стиску даних для пакетів, що надійшли на обслуговування є недоцільним. Оскільки будь-яке число  $N$  з діапазону  $[0, z_i]$  тільки єдиним чином може бути представлене у виді залишків за обраними взаємопростими основами, то число  $N$  та число  $A$  будуть містити одну і ту ж саму інформацію, що записана у різних системах числення [7], тобто  $N \equiv A$ , що надає змогу при зворотному перетворенні отримувати вихідну інформацію.

Крок 1.7. Повторимо виконання кроків 1.1 – 1.6, збільшивши при цьому пріоритет черги на обслуговування на одиницю. Така дія буде виконуватися доти, поки не буде перевірена доцільність застосування методики стиску даних до пакетів всіх пріоритетів.

Крок 1.8. У регістри, що вивільнилися після застосування методики стиску даних записуються пакети, що надходять на вузол телекомунікаційної мережі для обслуговування (рис. 3).

У випадках, якщо всі регістри черги визначеного пріоритету виявляться зайнятими пакет, що надійшов на обслуговування записується до черги нижчого пріоритету, а якщо це неможливо – у чергу вищого пріоритету. Його обслуговування розпочнеться після обслуговування всіх пакетів вищих пріоритетів та

декомпресії і подальшої обробки всіх стиснутих пакетів однакового з ним пріоритету.

Причому, для можливості здійснити декодування перетвореної методикою стиску даних інформації, пакет, що надійшов на обслуговування, переміститься у свою чергу тільки після вивільнення у цій черзі регістрів

Етап 2. Декомпресія (декодування) стиснутої інформації.

Декомпресія стиснутої інформації полягає у послідовному виконанні ряду кроків, а саме:

Крок 2.1. Здійснюється перевірка наявності у черзі найвищого пріоритету СМО вузла телекомунікаційної мережі стиснутої інформації. У випадку її відсутності пакети, що надійшли на обслуговування обробляються у відповідності до процедури застосування протоколу TCP. У протилежному випадку відбувається виконання кроку 2.2.

Крок 2.2. Здійснюється перевірка наявності у черзі разом зі стиснутою інформацією пакетів, що поступили на СМО вузла телекомунікаційної мережі для обслуговування після застосування методики стиску даних. У випадку, якщо у черзі такі пакети відсутні, здійснюється виконання кроку 2.3, у протилежному випадку – зазначені пакети записуються до черги нижчого пріоритету, а якщо це неможливо – обробляються СМО вузла телекомунікаційної мережі з метою вивільнення необхідного простору буфера накопичувача для декомпресії стиснутої інформації. Така процедура буде продовжуватися доти, поки у черзі разом зі стиснутою інформацією не залишиться пакетів, що надійшли на обслуговування після застосування методики стиску даних.

Крок 2.3. Здійснюється декомпресія стиснутої інформації шляхом переведу числа  $A$  із позиційної системи числення в систему числення залишкових класів застосувавши  $b$  раз формулу

$$\alpha_i = N - [N/p_i]p_i, \quad \text{для } i = 1, 2, \dots, b, \quad (7)$$

де  $p_1, p_2, \dots, p_b$  – послідовність заданих взаємопростих чисел (основ системи числення в залишкових класах).

Крок 2.4. Запишемо число  $N$  як множину, що має декілька елементів  $\alpha_b$ , (рис. 4). Тобто  $N = \{\alpha_1, \alpha_2, \dots, \alpha_b\}$ .

Отримані групи символів розглядаються СМО вузла телекомунікаційної мережі в якості пакетів, що надійшли на обслуговування зі стандартною структурою кадру.

Крок 2.5. Здійснюється обробка декомпресованих пакетів відповідно до процедури застосування протоколу TCP.

Крок 2.6. Здійснюється перезапис першого з переміщених у черги інших пріоритетів пакетів, що надійшли на обслуговування, після застосування методики стиску даних.

Крок 2.7. Здійснюється повторне виконання кроків 2.5, 2.6 зі збільшенням порядкового номеру пакету, що надійшов на обслуговування, після застосування методики стиску даних, на один. Така процедура буде продовжуватися доти, поки

СМО вузла телекомунікаційної мережі не обробить як всі декомпресовані пакети так і пакети, що надійшли на обслуговування після застосування методики стиску даних

Крок 2.8. Здійснюється повторне виконання кроків 2.1 – 2.7 зі зменшення значення пріоритету черги на обслуговування на одиницю. Однак в цьому випадку, при неможливості запису пакетів до черги нижчого пріоритету, вони записуються до черги вищого пріоритету. Окрім того, перед кожним повторним виконанням кроків 2.5 – 2.7 здійснюється перевірка наявності у чергах вищих пріоритетів пакетів, що надійшли на обслуговування. У випадках наявності таких пакетів – здійснюється їх першочергове обслуговування, починаючи з пакетів найбільшого пріоритету, у протилежному випадку виконання кроків 2.5 – 2.7 відбувається без змін. Така процедура буде продовжуватися доти, поки не буде здійснена декомпресія пакетів у чергах всіх пріоритетів.

Оцінка ефективності методики стиску даних

Максимальний ступінь стиску ( $k_c$ ), який може бути досягнутий при застосуванні методики стиску даних до пакетів, що знаходяться у черзі одного пріоритету дорівнює:

$$k_c = (z_i l / 1). \quad (8)$$

Ступінь стиску, розрахований згідно виразу (8) буде досягатися лише у випадках, коли значення числа  $A$  у черзі  $i$ -того пріоритету можна записати в одному розряді. При представленні числа  $A$  у двійковій системі числення ймовірність настання цієї події дорівнює

$$P_{\max} = \frac{1}{2^{z_i l}}. \quad (9)$$

Якість стиску ( $k_{cc}$ ) можна оцінити за допомогою математичного очікування кількості розрядів, що буде зайнято у буфері накопичувача  $i$ -того пріоритету після застосування методики стиску даних використовуючи вираз

$$k_{cc} = \frac{z_i l}{M(w)} = \frac{z_i l}{P_1 w_1 + P_2 w_2 + \dots + P_{z_i} w_{z_i}} = \frac{z_i l}{\sum_{j=1}^{z_i} P_j w_j} \quad (10)$$

$$N = 100100011111010010011111011001000001101001000110100010001001$$

Рис. 1. Приклад запису числа  $N$  у двійковій системі числення в вигляді пакунку, що містить шістдесят розрядів

$$N = 100100011111010010011111011001000001101001000110100010001001$$

Рис. 2. Приклад розбиття числа  $N$  на сім частин, кожна з яких має значення  $\alpha_j$

де  $P_j$  – ймовірність події, такої, що для запису числа  $A$  необхідно 1, 2, 3 і т. д. розрядів;

$w$  – кількість розрядів.

Аналізуючи вирази (9), (10) можна зробити висновок, що при використанні методики стиску даних ступінь стиску не залежить від типу потоку даних, що надає змогу застосовувати його до всіх пакетів, що стоять у черзі на обслуговування і відносяться до потоку даних.

Швидкість стиску (час, що витрачається на стиск інформації обсягом  $z_i l$  можна оцінити, скориставшись формулою:

$$T = x t_r, \quad (11)$$

де  $x$  – кількість машинних тактів, що є необхідною, для отримання числа  $A$ ;

$t_r$  – час, що є необхідним для здійснення одного такту.

Ймовірність відмови СМО вузла телекомунікаційної мережі спеціального призначення в обслуговуванні пакета, що розраховується згідно з виразом

$$P_{\text{від}} = P_{n+m} = \left(\frac{\alpha^n}{n!}\right) \left(\frac{\alpha}{n}\right)^m P_0 \quad (12)$$

де  $m = \sum_{i=1}^h z_i l$  – загальна кількість місць у чергах

СМО вузла телекомунікаційної мережі,

$h$  – кількість черг СМО вузла телекомунікаційної мережі при застосуванні методики стиску даних прийме вид

$$P_{\text{від}} = P_{n+m} = P_0 \left(\frac{\alpha^n}{n!}\right) \left(\frac{\alpha}{n}\right)^{\sum_{i=1}^h [z_i l + \frac{z_i l - M(w)}{1} \lceil 1 \rceil]}, \quad (13)$$

де  $\lceil Y \rceil$  – найближче менше ціле числа  $Y$ .

Залежність ймовірності відмови СМО вузла телекомунікаційної мережі в обслуговуванні пакета від щільності вхідного потоку пакетів  $\lambda$  наведений на рис. 5.

Аналізуючи графіки, наведені на рис. 5 можна стверджувати, що застосування методики стиску даних на вузлах телекомунікаційної мережі дозволить зменшити ймовірність відмови СМО в обслуговуванні пакетів.

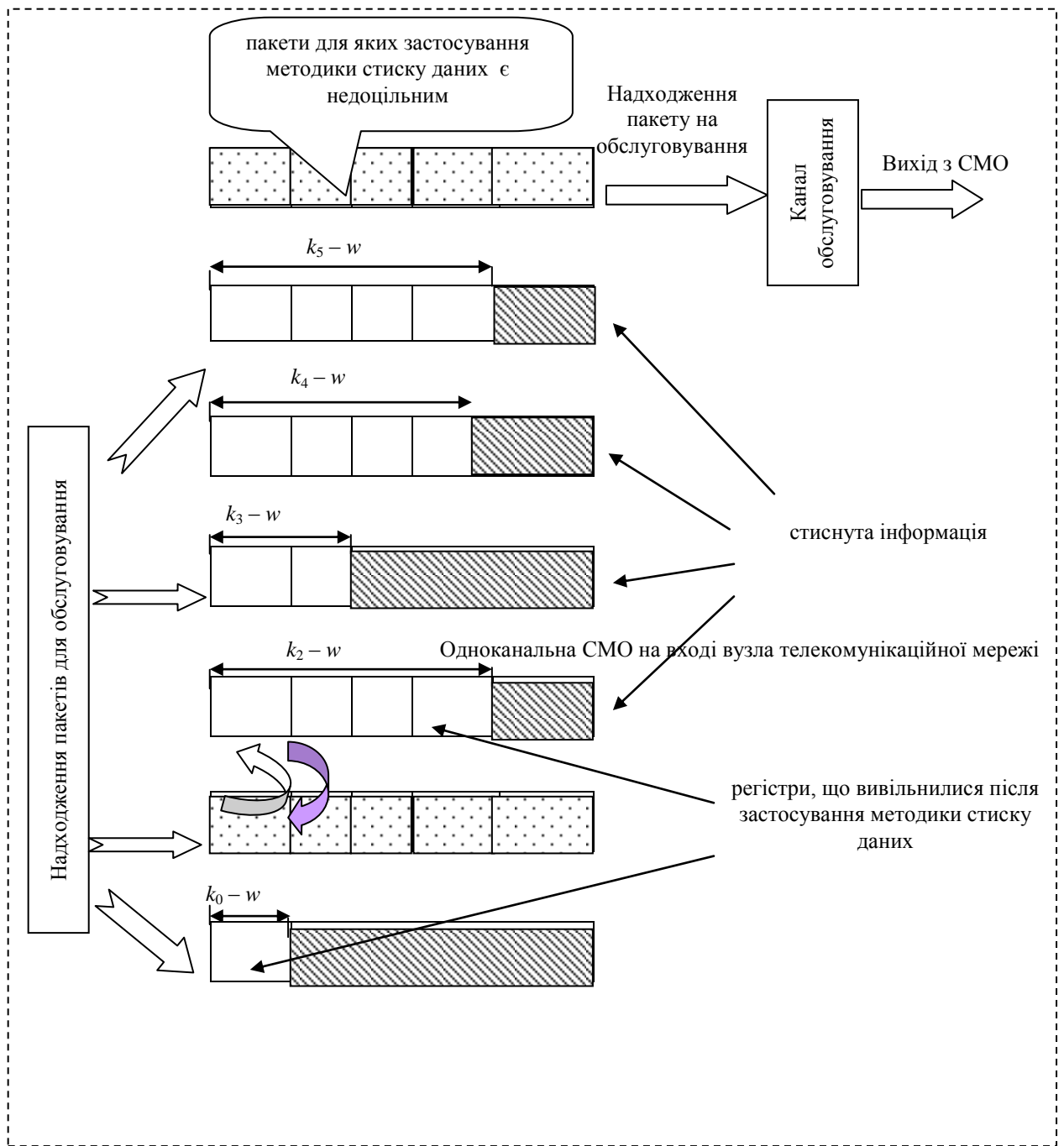


Рис. 3. Приклад застосування методики стиску даних для одноканальної СМО на вході вузла телекомунікаційної мережі

$$N = 00000000001111011001000001101001000110100010001001$$

$\underbrace{\hspace{1.5em}}_{\alpha_1} \quad \underbrace{\hspace{1.5em}}_{\alpha_2} \quad \underbrace{\hspace{1.5em}}_{\alpha_3} \quad \underbrace{\hspace{1.5em}}_{\alpha_4} \quad \underbrace{\hspace{1.5em}}_{\alpha_5} \quad \underbrace{\hspace{1.5em}}_{\alpha_6} \quad \underbrace{\hspace{1.5em}}_{\alpha_7}$

Рис. 4. Запис числа N, як множини, що має декілька елементів  $\alpha_b$

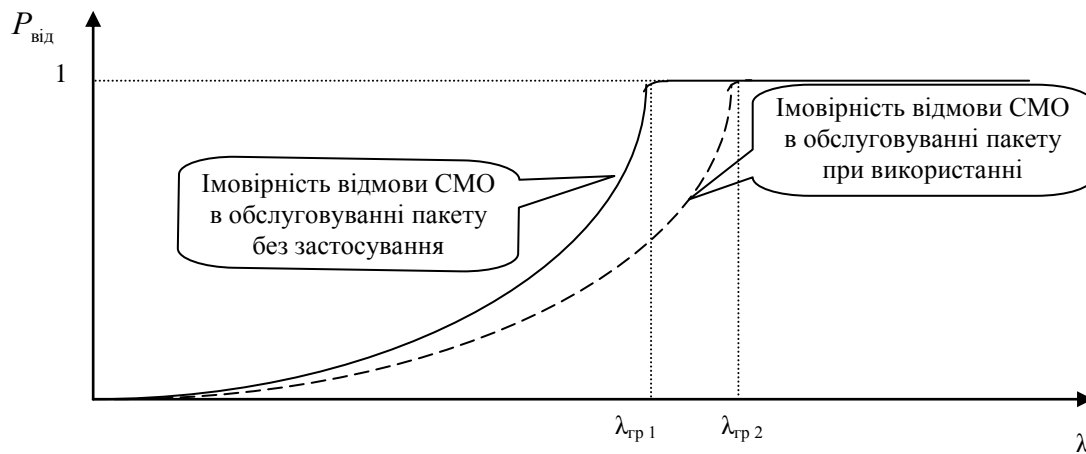


Рис. 5. Залежність імовірності відмови СМО вузла телекомунікаційної мережі в обслуговуванні пакета від щільності вхідного потоку пакетів  $\lambda$

### Висновки й перспективи подальших досліджень

В результаті проведених досліджень була розроблена методика стиску даних застосування якої спрямоване на зменшення імовірності відмови вузла в обслуговуванні пакетів.

Сутність зазначеної методики полягає в одночасному використанні декількох систем числення для представлення однієї і тієї ж інформації, яка міститься в пакетах, що надходять

для обслуговування, з метою мінімізації заповнення вільної області пам'яті буферів накопичувачів.

Зазначена методика, використовуючи властивості системи числення в залишкових класах дозволяє отримувати ступінь стиску, який не залежить від типу потоку даних, що надає змогу застосовувати його до всіх пакетів, що стоять у черзі на обслуговування і відносяться до потоку даних.

### Література

1. Канновская И. Ю. Военно-прикладная теория вероятностей: Часть II: Учебное пособие. – Киев: КВИРТУ ПВО, 1987. – 208 с. 2. Clark D., Jacobson V., Romkey J., and Sal Wen H.: «An Analysis of TCP Processing Overhead», IEEE Commun. Magazine, vol. 27, pp. 23-29, June 1989. 3. Ghani N., and Dixit S.: «TCP/IP Enhancements for Satellite Networks», IEEE Commun. Magazine, vol. 37, pp. 64-72, 1999. 4. Nagle J. «Congestion Control in TCP/IP Internetworks», Computer Commun.

Rev., vol. 14, pp. 11-17, Oct. 1984. 5. Clark D. D: «Window and Acknowledgement Strategy in TCP», RFC 813, July 1982. 6. Глушко О. В. Основы вычислительной техники и программирование: Конспект лекций / О. В. Глушко, В. С. Ланцев, М. А. Патланжоглу, В. Д. Руденко. – К.: КВИРТУ ПВО, 1986. – 216 с. 7. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. -М : Советское радио, 1968. - 438 с.

Представлены результаты разработки имитационной модели орбитального информационного комплекса космических систем дистанционного зондирования Земли. Результаты могут быть использованы при исследовании пространственно-временных и информационных возможностей рассмотренного класса систем.

Ключевые слова: имитационное моделирование, космическая система дистанционного зондирования Земли.

Presents the results of a simulation model of the orbital space complex information systems for remote sensing. The results can be used to study the space-time information capabilities and the classes of systems.

Key words: simulation, space system for Earth remote sensing.