

<https://doi.org/10.31891/2219-9365-2026-85-6>

УДК 004.7.052:004.414.2

ТОРОШАНКО Андрій

Державний університет «Київський авіаційний інститут»

<https://orcid.org/0000-0002-0816-657X>

atoroshanko@gmail.com

АЛГОРИТМИ КЕРУВАННЯ ПОТОКАМИ В МЕРЕЖАХ З ВІРТУАЛЬНИМИ КАНАЛАМИ

У роботі представлений модифікований підхід до керування потоками у мережах із віртуальними з'єднаннями. Побудовано математичну модель керування потоками даних в комп'ютерній мережі з віртуальними з'єднаннями, яка враховує наявність індивідуальних конфліктних цілей: максимізація пропускну здатності та мінімізація середньої затримки передачі даних через пучок віртуальних каналів. Проаналізовані умови і способи досягнення стану рівноваги між інтенсивністю запитів і наявним ресурсом мережі. Сформульовано основні критерії справедливості та оптимальності розподілу ресурсів між віртуальними каналами. Пропонуються різні метрики справедливості, однією з яких є Парето-оптимальність.

Ключові слова: комп'ютерна мережа, пучок віртуальних каналів, оптимальність розподілу ресурсів, стан рівноваги, метрики справедливості, Парето-оптимальність

TOROSHANKO Andrii

State university «Kyiv aviation institute»

FLOW CONTROL ALGORITHMS IN NETWORKS WITH VIRTUAL CHANNELS

The paper presents a modified approach to flow control in computer networks that support virtual connections by introducing a resource-aware coordination mechanism for bundles of virtual channels. A bundle of virtual channels is defined as a set of co-directed physical links over which multiple software applications establish logical communication paths. Such an abstraction enables concurrent transmission of heterogeneous data streams while sharing a common physical infrastructure. The study analyzes two alternative transmission architectures. In the first scenario, each data stream is mapped to an independent logical TCP connection, which simplifies congestion isolation but increases protocol overhead and resource fragmentation. In the second scenario, a single physical TCP connection encapsulates multiple virtual channels, reducing control overhead and improving link utilization at the cost of more complex internal flow coordination.

The key control variables in a virtual channel bundle are the arrival intensity of data transfer requests and the limited network communication resources, including bandwidth, buffer capacity, and scheduling time. The paper formulates the problem of conflict-free coordination between these parameters as a multi-objective optimization task. A mathematical model of flow control is developed that captures the stochastic nature of request generation, the shared resource constraints, and the presence of competing performance goals—maximization of aggregate throughput and minimization of the average packet delay across all virtual channels. The model represents the network as a distributed system of interacting queues with adaptive service rates and introduces cost functions that reflect the resource expenditure per transmitted data unit.

Based on this model, decentralized algorithms are proposed for request shaping, rate adaptation, and dynamic allocation of communication resources among virtual channels. The algorithms operate using local state information and limited signaling, which ensures scalability and robustness in large-scale distributed environments. A mechanism for balancing the request intensity with the available network capacity is derived, allowing the system to converge to an equilibrium operating point that prevents persistent congestion while maintaining high utilization. Special attention is given to the minimization of transmission cost per virtual channel, achieved through proportional resource distribution and adaptive throttling of aggressive flows.

To evaluate the quality of resource sharing, several fairness criteria are formulated, including max-min fairness, proportional fairness, and Pareto optimality. The Pareto-efficient allocation is shown to provide a balanced trade-off between efficiency and equity, ensuring that no virtual channel can improve its performance without degrading another. The proposed approach enables efficient multiplexing of multiple data streams over shared TCP transport, reduces protocol overhead, and improves delay-throughput characteristics. The results demonstrate that decentralized flow control with virtual channel bundling provides a flexible and scalable solution for modern software-defined and overlay networks where heterogeneous traffic demands must be coordinated over constrained communication resources.

Keywords: computer network, bundle of virtual channels, optimality of resource allocation, equilibrium state, fairness metrics, Pareto-optimality

Стаття надійшла до редакції / Received 02.01.2026

Прийнята до друку / Accepted 15.02.2026

Опубліковано / Published 05.03.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Торошанко Андрій

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Інформаційно-комунікаційну мережу можна розглядати як велику систему [1-5], елементами якої є віртуальні канали та мережні вузли. Поняття та способи формування віртуальних каналів досить широко описані в науково-технічній літературі, зокрема в [4-8]. За визначенням [9] віртуальний канал – це фіксований маршрут, який складається з послідовності тимчасово створених каналів та пов'язаних з ними номерів портів комутаторів, через які проходять всі пакети при даному сеансі зв'язку від одного користувача до іншого.

Використання віртуальних каналів насамперед призначене для збільшення продуктивності роботи підсистеми передачі. Збільшення продуктивності роботи досягається за рахунок зменшення обсягів службового трафіку, що передається через мережу.

За визначенням [10], пучок віртуальних каналів – це сукупність однаково орієнтованих фізичних каналів, у яких сформовані деякі програмні застосунки. Ці застосунки можна використовувати для додавання функціональних розширень у мережні служби, напр., для керування навантаженнями певної групи джерел та раціонального розподілу загального навантаження в інформаційно-комунікаційних мережах (ІКМ).

Розглянемо ІКМ як велику систему [1-3]. У такій системі працюють два типи стаціонарних агентів [11] – віртуальні канали (*virtual channels*) та вузли мережі. Кожен віртуальний канал забезпечує маршрут передачі даних через сукупність реальних каналів зв'язку, що входять у даний віртуальний канал, вимоги користувача до затримки (*delay, latency*) і пропускної спроможності (*throughput*) віртуального з'єднання і величину наявних віртуальних запитів.

Ключовими параметрами пучка віртуальних каналів є запити та комунікаційні ресурси, при цьому основною системною проблемою є намагання узгодити ці параметри між собою без конфліктів, або хоча б мінімізувати число конфліктів. Кожен віртуальний канал наділяється деяким допустимим числом запитів на комунікаційні ресурси, наприклад, пропускної спроможності реальних каналів зв'язку. Вузол мережі як стаціонарний агент встановлює запити на комунікаційні ресурси і наділяється своєю метою – обчислити так званий "рівноважний запит", при якому запит на ресурси узгоджений з їхнім об'ємом.

Кожному новому віртуальному каналу також задається кінцева мета стосовно досягнення прийнятних пропускної здатності та затримки, та формуються апріорні дані про інтенсивність запитів для визначення можливості досягнення цієї мети. Якщо така можливість відсутня, можна відкласти формування віртуального каналу до того часу, поки інтенсивності запитів не знизяться, або змінити мета. Таким чином, запити комп'ютерної мережі грають роль керування доступом (*admission control*). Інтенсивність запитів є показником інформаційної цінності ресурсів. Тому використання інформаційних механізмів у комп'ютерних та телекомунікаційних мережах сприятиме підвищенню ефективності використання їх ресурсів.

Варіанти передачі потоків даних між програмними застосунками

Проаналізуємо два можливі варіанти передачі кількох потоків даних між програмними застосунками. У першому випадку для кожного потоку даних створюється свій логічний канал передачі за протоколом ТСП. Схематично цей випадок подано на рис. 1.

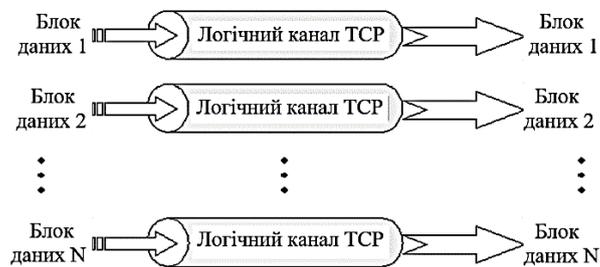


Рис. 1. Блоки даних передаються через окремі логічні канали

У наведеному прикладі має місце надмірність службових даних, що передаються, наприклад, надмірність пакетів, що беруть участь у побудові з'єднання. Припустимо, одне із віртуальних з'єднань буде закрито після передачі декількох пакетів даних, а зайнятість логічного каналу своїм віртуальним каналом, тим не менш, буде продовжена. Очевидно, ефективність даної схеми буде зменшена.

У другому випадку для всіх потоків даних створюється один реальний канал передачі за протоколом ТСП, який інкапсулює кілька віртуальних каналів. Схематично цей випадок подано на рис. 2.

Дані транспортного рівня передаються у загальний логічний ТСП-канал. У цьому випадку кількість службової інформації може бути зведена до мінімуму. Цей механізм є особливо актуальним для систем, де дані передаються невеликими порціями. При одночасному надходженні запитів на передачу даних від кількох блоків пропонується об'єднувати дані в один буфер і надсилати їх одночасно. На прийомі необхідно виконувати зворотну операцію – розбиття буфера з подальшим розподілом інформації по відповідних блоках. Подібна схема допомагає уникнути витрат, що виникають при установці/розриві додаткових мережних з'єднань.

Моделлю комп'ютерної мережі може бути спрямований граф $G = \langle V, E \rangle$, де $V = \{1, 2, \dots$ – множина вузлів, $E = \{1, 2, \dots$ – множина ребер графу.

Кожне ребро з множини E характеризується двома величинами:

- теоретична пікова пропускна спроможність C_{peak} (вимірюється у кількості даних, переданих в одиницю часу);
- реальна пропускна спроможність C_{commit} , причому $C_{commit} < C_{peak}$;
- величина $e_i = \psi(C_{peak}, C_{commit})$ представляє собою функціонал пропускних спроможностей ребра e_i

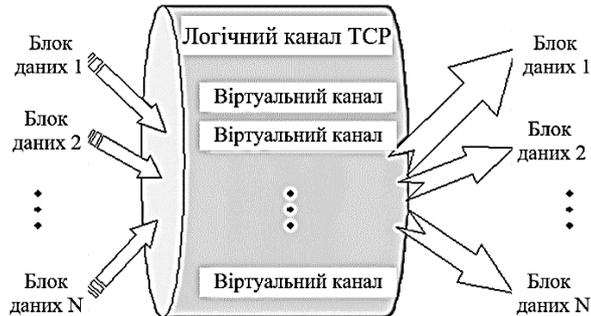


Рис. 2. Перехресне застосування віртуальних і реальних каналів

Математична модель мережі з віртуальними каналами

Величина C_{commit} є верхньою межею ступеня використання i -го каналу зв'язку. Вона введена для обмеження затримки. При цьому ступінь узагальненості ніяк не зменшується, оскільки C_{commit} можна взяти як завгодно близькою до C_{peak} . Мета керування потоками полягає в оптимальному розподілі пропускної спроможності між віртуальними каналами.

Нехай $A = \{1, 2, \dots, N\}$ означає сукупність віртуальних каналів. Припустимо, що величина A залишається незмінною. Кожен віртуальний канал визначається сукупністю реальних каналів, які входять до його складу, тобто, сукупністю ребер графа G : $L_a = \{i_1, i_2, \dots\}$, де $i \in E$.

Віртуальні канали, видаючи запити на комунікаційні ресурси, по суті, беруть участь у їх розподілі. Множина ресурсів, що дісталися в результаті розподілу даному віртуальному каналу, описується вектором $\mathbf{X} = \{x_1, x_2, \dots, x_M\}$, $x_i \geq 0$, $i = \overline{1, M}$. Кожен реальний канал керується своїм планувальником (*scheduler*), який формує віртуальні канали. Множина запитів задається вектором:

$$\mathbf{P} = \{p_1, p_2, \dots, p_M\}, \text{ де } p_i \geq e_i, e_i \in E, p_i \in \mathbf{P}, i = \overline{1, M}.$$

Припущення, що $p_i > 0$, не призводить до будь-якої втрати узагальненості розгляду, бо може бути як завгодно близьким до нуля.

Для кожного віртуального каналу введемо відношення переваги \succ на множині розподілів. Нехай вектор \mathbf{X} більш придатний для віртуального каналу a_{virt} , ніж \mathbf{Y} . Позначимо це припущення таким чином: $\mathbf{X} \succ_{a_{virt}} \mathbf{Y}$. Якщо $\mathbf{X} \succ_{a_{virt}} \mathbf{Y}$ і $\mathbf{Y} \succ_{a_{virt}} \mathbf{Z}$, то для віртуального каналу вектори \mathbf{X} та \mathbf{Z} рівноцінні.

Якщо ж $\mathbf{X} \succ_{a_{virt}} \mathbf{Y}$, а \mathbf{X} та \mathbf{Y} для каналу a_{virt} не рівноцінні, тоді вектор \mathbf{X} строго придатніший ніж \mathbf{Y} . Позначимо це твердження як $\mathbf{X} \succ_{a_{virt}}^C \mathbf{Y}$.

Нехай $T_{vc}(R_x)$ – пропускна спроможність віртуального каналу a_{virt} , який отримав у результаті розподілу ресурси \mathbf{X} , де $\mathbf{X} = \{x_1, x_2, \dots, x_M\}$ – множина ресурсів пучка віртуальних каналів; а функція

середньої затримки доставки по каналу a_{virt} визначена як $\tau_{virt} \left(\sqrt{\frac{1}{M} \sum_{i=1}^M |x_i|^2} \right)$. Аргументом цієї функції є

середньоквадратичне відхилення $\sqrt{\frac{1}{M} \sum_{i=1}^M |x_i|^2}$ затримок по всіх реальних каналах x_i .

Віртуальний канал a_{virt} намагається максимізувати $T_{vc}(R_x)$, поки не досягне мети g_a , (g_a – кількісний параметр мети, заданої користувачем віртуального каналу). Потім віртуальний канал a_{virt}

намагається мінімізувати значення функції середньої затримки $\tau_{virt} \left(\sqrt{\frac{1}{M} \sum_{i=1}^M |x_i|^2} \right)$.

Відношення переваги \succ визначимо так:

якщо $T_{vc}(R_x) \leq g_a$, $T_{vc}(\mathbf{Y}) \leq g_a$ і $T_{vc}(\mathbf{Y}) \geq T_{vc}(R_x)$, то $\mathbf{Y} \succ_{virt}$;

якщо $T_{vc}(R_x) < g_a$ і $T_{vc}(\mathbf{Y}) \geq g_a$, то $\mathbf{Y} \succ_{virt}$;

якщо $T_{vc}(R_x) = g_a$, $T_{vc}(\mathbf{Y}) \geq g_a$ і $\tau_{virt} \left(\sqrt{\frac{1}{M} \sum_{i=1}^M |x_i|^2} \right) \geq \tau_{virt} \left(\sqrt{\frac{1}{M} \sum_{i=1}^M |y_i|^2} \right)$, то $\mathbf{Y} \succ_{virt}$.

Введене таким чином співвідношення не охоплює всі випадки, однак є достатнім для поставлених у роботі цілей.

Якщо програмний застосунок є критичним по пропускній спроможності (наприклад, передача файлів), число віртуальних каналів, які відповідають меті g_a , має бути досить великим. Якщо ж застосунок є критичним по затримці (як, наприклад, у випадку, коли віртуальний канал з'єднує робочу станцію з базою даних), допускається порівняно мале g_a (наприклад, достатньо взяти середній розмір пакета в транзакції, поділений на час роздумів (*think time*) користувача).

Оскільки пропускна спроможність віртуального каналу визначається швидкістю найповільнішого реального каналу, що входить до його складу, то $T_{vc}(R_x) = \min_{i \in L_a} \{x_i\}$, $i \in T_{vc}(a_{ic})$.

Позначимо затримку доставки через віртуальний канал як τ_{virt} . Тоді розглядаючи реальний i -й канал зв'язку як систему масового обслуговування типу M/M/1, середню затримку на каналі в найгіршому випадку (тобто розглядається верхня межа для середньої затримки) можна визначити наступним чином [12, 13]:

$$\tau_{virt \max} = \frac{1}{(C_{peak} - C_{commit}) + (x_i - g_a)} \quad (1)$$

Середня затримка на всьому віртуальному каналі:

$$\tau_{virt \text{total}} = \sum_{i \in L_a} \frac{1}{(C_{peak} - C_{commit}) + (x_i - g_a)} \quad (2)$$

Кожен віртуальний канал a_{virt} наділяється деяким початковим ресурсом $Rs_{a_{virt}} > 0$, що відповідає пріоритетності a_{virt} . Таким чином ресурсну систему каналу a_{virt} із врахуванням (1) і (2) можна записати у вигляді: $Rs_{a_{virt}}(\mathbf{P}) = \{ \mathbf{X} \mid P^* \mathbf{X} \leq Rs_{a_{virt}} \}$.

Запит віртуального каналу на комунікаційні ресурси

Функціонал запиту з боку a_{virt} на комунікаційні ресурси представляється наступним чином [11, 13]:

$$\Phi_{a_{virt}}(\mathbf{P}) = \{ \mathbf{X}_{|\mathbf{X}| \in Rs_{a_{virt}}(\mathbf{P})}, \mathbf{X} \succ_{virt} \}, \text{ для всіх } |\mathbf{Y}| \in Rs_{a_{virt}}(\mathbf{P}).$$

Позначимо $\mathbf{Q} = (q_1, q_2, \dots)$ як M -вимірний вектор запитів (*queries*) від пучка віртуальних каналів.

Витрати $q_{ai \text{real}}$ на передачу одиниці даних по i -му реальному каналу дорівнюють p_i .

Отже, витрати на передачу одиниці даних по всьому віртуальному каналу визначаються як

$$a_{virt}: \quad \mathbf{Q}_{a_{virt}} = \sum_{i \in L_a} p_i.$$

Оскільки віртуальний канал a_{virt} прагне передавати дані зі швидкістю $Rate_{g_a}$, загальні витрати становитимуть: $Rate_{g_a} \times \mathbf{Q}_{a_{virt}}$, причому має виконуватися умова: $\mathbf{Q}_{a_{virt}} \times Rate_{g_a} = Rs_{a_{virt}}(\mathbf{P})$.

Якщо віртуальний канал може досягти мети g_a , він на кожному реальному каналі, що входить в L_a , отримує за запитами P пропускну здатність g_a . Різниця $Rs_{a_{virt}} - (\mathbf{Q}_a \times g_a)$, що залишилася, використовується для мінімізації середньої затримки.

Введемо позначення: $u_i = x_i - g_a$, $K_i = C_{peak} - C_{commit}$, $U = (u_1, u_2, \dots)$. Завдання мінімізації середньої затримки τ_{aver} через пучок віртуальних каналів сформулюємо в такий спосіб:

$$\tau_{aver} = \sum_{i \in L_a} \frac{1}{K_i + u_i} \rightarrow \min_{\substack{u_i \geq 0 \\ \mathbf{P} \times U \leq Rs_{a_{virt}} - \mathbf{Q}_a \times g_{a_{virt}}}} \quad (3)$$

за умов: $u_i \geq 0$, $\mathbf{P} \times U \leq Rs_{a_{virt}} - \mathbf{Q}_a \times g_{a_{virt}}$.

Область допустимих значень U опукла і компактна, а функція $F(U)$ безперервна і строго опукла у цій області [14]. Це означає, що існує єдине рішення завдання.

Формулюючи спрощену умову Куна-Такера [15, 16] для поставленого завдання, отримаємо для всіх i :

$$u_i = 0, \text{ або } \frac{1}{p_i^*(K_i + u_i)^2} = v, \quad (4)$$

де v – множник Лагранжа [2].

Перша з цих умов означає, що у оптимумі $x_i \geq g_{avirt}$. Друга умова говорить, що гранична затримка, що припадає на одиницю нормалізованого ресурсу, має бути однаковою у всіх реальних i -х каналів і для яких $x_i > g_a$. Зауважимо, що з (3) і (4) випливає така рівність (для всіх $u_i > 0$): $u_i = \frac{1}{\sqrt{p_i \cdot v}} - K_i$.

Кожен віртуальний канал використовує для обчислення множника Лагранжа алгоритм бінарного пошуку [18], поданий нижче:

1. $v^d = 0$ (нижня межа для v).

2. $v^u = \max \left\{ \frac{1}{p_i^*(K_i)^2}, i \in L_a \right\}$ (верхня межа для v , випливає з (4)).

$$3. v = \frac{(v^u - v^d)}{2}.$$

4. Для всіх $i \in L_a$:

а) $u_i = \frac{1}{\sqrt{p_i - v}} - K_i$; б) якщо $u_i < 0$, то $u_i = 0$.

5. Якщо $\mathbf{P}^*U = W_a - \mathbf{Q}_a^*g_a$, то U – оптимально; рішення знайдено, перехід до кроку 8. Інакше перехід до кроку 6.

6. Якщо $\mathbf{P}^*U > W_a - \mathbf{Q}_a^*g_a$, то $v^d = v$ і перехід до кроку 3 (обчислення нового значення v). Інакше перехід до кроку 7.

7. Якщо $\mathbf{P}^*U < W_a - \mathbf{Q}_a^*g_a$, то $v^u = v$ і перехід до кроку 3 (обчислення нового значення v). Інакше перехід до кроку 8 (закінчення обчислень).

8. Вихід.

Після обчислення U запит віртуального каналу a на комунікаційні ресурси визначається однозначно:

$$\tau_{i virt}^a = g_a + u_i, \text{ якщо } i \in L_a; \quad \tau_{i virt}^a = 0, \text{ якщо } i \notin L_a.$$

Загальний запит на i -й ресурс за запитами P дорівнює сумі індивідуальних запитів кожного з віртуальних каналів: $\tau_{i virt total} = \sum_{a \in A} \tau_{i virt}^a(\mathbf{P})$.

Різницю між запитом на ресурс і наявним ресурсом назовемо функцією надлишку (або просто надлишком) [19]. Позначимо її як $Z(\mathbf{P})$ – вектор з i компонентами $z_i(\mathbf{P}) = \tau_{i virt}^a(\mathbf{P}) - C_{commit}$.

Якщо $\tau_{i virt}^a(\mathbf{P}^*) = 0$ по всім i , то при рівноважних запитах \mathbf{P}^* система перебуває у стані конфліктної рівноваги [20]. Зазвичай, рівновага визначається як стан, у якому:

– немає невикористаних ресурсів;

– запит кожного стаціонарного агента (віртуального каналу) повністю задоволений.

До інтенсивності запитів комп'ютерної мережі це визначення непридатне, оскільки можлива ситуація, коли C_{commit} приймає значення строго більші, ніж $\tau_{i virt}^a(\mathbf{P})$, як би низько не падала інтенсивність запитів.

Нехай є два віртуальні канали, $Rs_{avirt1} = Rs_{avirt2} = 1$, $g_1 = g_2 = \infty$ та $C_{commit i} = 1$ (по всім i). В силу симетрії, віртуальні канали поділять третій реальний канал навпіл. Віртуальний канал 1 отримає (1/2, 0, 1/2), віртуальний канал 2 отримає (0, 1/2, 1/2). Надлишок складе: $Z(\mathbf{P}) = (-1/2, -1/2, 0)$.

Щоб виключити подібні ситуації, введемо нове визначення рівноваги. Якщо по всім i або $Rs_{avirt}(\mathbf{P}^*) = 0$, або $Rs_{avirt}(\mathbf{P}^*) \leq 0$ і $p_i = e$, то інтенсивність запитів і наявний поточний ресурс мережі при запитах \mathbf{P}^* перебувають у стані рівноваги.

Як впливає з цього визначення, надлишок (у стані рівноваги) допускається за мінімального обсягу запитів. Вузли мережі, будучи джерелами даних, встановлюють запити на основі співвідношення запиту і наявних ресурсів.

Як початкові запити можна брати $p_i = e$ для всіх i . На другому етапі всі віртуальні канали обчислюють запит при поточних запитах паралельно. Для виконання цих обчислень кожному віртуальному каналу, а потрібна інформація про запити на реальні канали, що входять до L_a . Параметр e отримується експериментально і відображає важливість оновлення запитів. На останньому кроці планувальнику реального каналу і необхідно знати ідентифікатори всіх віртуальних каналів, до складу яких входить даний реальний канал.

Справедливість та оптимальність алгоритмів керування потоками

Сформулюємо чотири критерії справедливості та оптимальності розподілу ресурсів [17] між віртуальними каналами.

(Критерій F1) Розподіл $\{X^1, X^2, \dots\}$ справедливий, якщо для всіх a з A виконується нерівність $T_{vc}(R_x) > 0$. Цей критерій стверджує, що не повинно бути віртуальних каналів, яким би дісталася внаслідок розподілу нульова пропускна спроможність. Будемо називати реальний канал i з L_a заторним каналом (від слова затор, вузьке місце) віртуального каналу a , якщо $T_{vc}(R_x) = x_i^a$. Таким чином, заторний канал визначає пропускну спроможність віртуального каналу.

(Критерій F2) Нехай i -й канал є пробним каналом віртуального каналу a . Розподіл $\{X^1, X^2, \dots\}$ справедливий, якщо $x_i^a \geq x_i^b$ для всіх b з A . Цю умову іноді називають максимінною, оскільки вона визначається виходячи з максимізації мінімуму пропускної спроможності. Цей критерій передбачає, що це віртуальні канали прагнуть максимізувати пропускну здатність, але з враховують затримку.

(Критерій F3) Розподіл $\{X^1, X^2, \dots\}$ справедливий, якщо для всіх віртуальних каналів a з A $X_a \succ_{R_{S_{a,virt}}(P)}$ для всіх Y з $R_{S_{a,virt}}(P)$. Ця умова побудована на основі поняття ігрової стабільності по Нешу [17]. На відміну від F1 і F2, критерій F3 враховує і пропускну здатність, і затримку (через те, що він ґрунтується на перевазі).

Перш ніж сформулювати четвертий критерій, запровадимо кілька допоміжних понять. Слідуючи [19, 20], назвемо коаліцією S_c деяку підсистему A . Коаліція S_c може покращити розподіл $\{X^1, X^2, \dots\}$, якщо знайде інший розподіл $\{Y^1, Y^2, \dots\}$, такий, що:

- $Y_a = X_a$, де a не належить S_c (формування коаліції не впливає на тих, хто до неї не входить);
- $Y_a \succ$ для a з S_c (для всіх членів коаліції новий розподіл не гірший за старий);
- знайдеться таке a з S_c , для якого $Y^a \succ$ (хоча б для одного члена коаліції новий розподіл строго кращий за старий).

Тепер визначимо четвертий критерій.

(Критерій F4) Розподіл $\{X^1, X^2, \dots\}$ справедливий, якщо немає таких коаліцій, які могли б покращити розподіл.

Умова F4 еквівалентна визначенню оптимальності за Парето. Очевидно, що справедливі твердження:

- якщо справедливий критерій F2, то справедливий і критерій F1;
- якщо справедливий критерій F4, то справедливий і критерій F3.

Зауважимо: чим більш великомасштабною є система, тим менш придатні загальносистемні характеристики керування нею. Більш підходящими у таких випадках будуть децентралізовані алгоритми (тобто алгоритми, що не базуються на загальносистемних характеристиках). Запропонована у роботі алгоритмічна структура є децентралізованою.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У роботі представлений модифікований підхід до управління потоками у мережах із віртуальними з'єднаннями. Побудовано математичну модель комп'ютерної мережі, яка враховує наявність індивідуальних конфліктних цілей: максимізація пропускної спроможності та мінімізація затримки. У багатьох роботах з управління потоками в мережах передбачається, що віртуальні канали мають однакові цілі щодо пропускної здатності і затримки, або робляться спроби оптимізувати загальносистемні (загальномережні) характеристики. Оскільки мережні технології постійно розвиваються, збільшуються можливості підтримки (врахування особливостей) дедалі більше диверсифікованих додатків і користувачів. Це робить припущення

про однакові цілі мало реалістичним. Далі, дуже важко визначити загальносистемні характеристики, які адекватно відображали б гетерогенність користувачів і додатків, їхню різну пріоритетність. Очевидно, що більш великомасштабною є система, тим менш придатні загальносистемні характеристики керувати нею. Цілком можливо, що більш підходящими у таких випадках будуть децентралізовані алгоритми (тобто алгоритми, що не базуються на загальносистемних характеристиках). Запропонована у роботі алгоритмічна структура є децентралізованою. У будь-яких завданнях розподілу ресурсів важливе питання справедливості (розподілу). У зв'язку з цим пропонуються різні метрики справедливості, однією з яких є Парето-оптимальність. Доведено, що запропоновані алгоритми управління потоками в комп'ютерних мережах Парето-оптимальні.

Література

1. Suh N. P. Design and Operation of Large Systems. *Journal of Manufacturing Systems*, Vol. 14, No. 3, 1995. – PP. 203-213. [https://doi.org/10.1016/0278-6125\(95\)98887-C](https://doi.org/10.1016/0278-6125(95)98887-C)
2. Lasdon, L.S., & Tabak, D. (1970). Optimization Theory of Large Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1, 300-301.
3. Waddell S., Waddock S., Cornell S., Dentoni D., McLachlan M., and Meszoely G. Large Systems Change: An Emerging Field of Transformation and Transitions. Taylor & Francis, Ltd. *The Journal of Corporate Citizenship*, No. 58, 2015. – PP. 5-30. <https://doi.org/10.4324/9781003579380>
4. Tanenbaum A., Feamster N., and Wetherall D. *Computer Networks*, Global Edition Pearson, 6th Edition, 2021. – 944 p.
5. Стеклов В. К., Кільчицький Є. В. Основи управління мережами та послугами телекомунікацій. Київ: Техніка, 2002. – 438 с.
6. Stallings W. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Pearson Education, Inc., Old Tappan, New Jersey, 2016. – 544 p.
7. Bonaventure O. *Computer Networking: Principles, Protocols and Practices*. Release. Сnp3book, 2018. – 272 p.
8. Kurose J. F., Keith W., and Ross W. K. *Computer Networking: A Top-Down Approach: 7th ed*. Pearson Education, Inc., 2017. – 864 p.
9. Закон України "Про електронні комунікації". Офіційний вісник України від 26.01.2021 р., № 6, стор. 10, стаття 306. Редакція від 29.07.2023.
10. Ложковський О. Г. Статистичне моделювання повнодоступного пучка з втратами. *Наукові праці ОНАЗ ім. О.С. Попова*, 2003, № 1. – С. 75-82.
11. Chen H., Huang L., Kumar S., and Kuo C. C. J. *Radio Resource Management for Multimedia QoS Support in Wireless Networks*. Springer, 2004. – 271 p.
12. Seung Hyong Rhee, Takis Konstantopoulos, Hyukjoon Lee, and Kwangsue Chung. Competitive Routing and Flow Control in Communication Networks of Parallel Links. *Computer Networks*, 42, 2003. – PP. 135-140.
13. Rosberg Z., and Gopal I. S. Optimal Hop-by-Hop Flow Control in Computer Networks. *IEEE Trans. on Automatic Control*, vol. AC-31, No. 9, 1986. – PP. 813-822.
14. Pallaschke D. E., and Urbanski R. *Pairs of Compact Convex Sets: Fractional Arithmetic with Convex Sets (Mathematics and its Applications)*. Springer, 2010. – 308 p.
15. Dimitri P., and Bertsekas D. P. *Nonlinear Programming 2nd Edition*, Athena Scientific, 1999. – 780 pp.
16. Bazaraa M. S., Sherali H. D, and Shetty C. M. *Nonlinear Programming: Theory and Algorithms: 3rd edition*. Wiley-Interscience; 2006. – 880 p.
17. Nash J. C. *Compact numerical methods for computers: linear algebra and function minimisation*, 2nd ed. Adam Hilger, Bristol and New York, 1990. – 278 p.
18. Biswas S. K., Pranab K. M., Uttam K., and Roy U. K. Binary Search-Based Fast Scheduling Algorithms for Reliability-Aware Energy-Efficient Task Graph Scheduling with Fault Tolerance. *IEEE Transactions on Sustainable Computing*, V. 9, Issue 3, 2023. – PP. 433-451.
19. Karwan M., Lotfi V., Telgen J., and Zionts S. *Redundancy in Mathematical Programming: A State-of-the-Art Survey*. Publisher: Springer Berlin Heidelberg, 1983. – 300 p.
20. Saaty T. L. *Mathematical Models of Arms Control & Disarmament: Application of Mathematical Structures to Politics*. Operations Research Society of America Publications, No. 14. John Wiley and Sons, 1968. – 190 p.

References

1. Suh, N. P. (1995). Design and operation of large systems. *Journal of Manufacturing Systems*, 14(3), 203–213. [https://doi.org/10.1016/0278-6125\(95\)98887-C](https://doi.org/10.1016/0278-6125(95)98887-C)
2. Lasdon, L. S., & Tabak, D. (1970). Optimization theory of large systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1, 300–301.
3. Waddell, S., Waddock, S., Cornell, S., Dentoni, D., McLachlan, M., & Meszoely, G. (2015). Large systems change: An emerging field of transformation and transitions. *The Journal of Corporate Citizenship*, 58, 5–30. <https://doi.org/10.4324/9781003579380>
4. Tanenbaum, A., Feamster, N., & Wetherall, D. (2021). *Computer Networks* (6th ed.). Pearson.

5. Steklov, V. K., & Kilchytskyi, Ye. V. (2002). *Fundamentals of management of telecommunication networks and services*. Kyiv: Tekhnika.
6. Stallings, W. (2016). *Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud*. Pearson Education.
7. Bonaventure, O. (2018). *Computer networking: Principles, protocols and practices*. CNP3book.
8. Kurose, J. F., & Ross, K. W. (2017). *Computer networking: A top-down approach* (7th ed.). Pearson Education.
9. Law of Ukraine. (2021). *On electronic communications*. Official Bulletin of Ukraine, No. 6, Article 306 (as amended July 29, 2023).
10. Lozhkovskiy, O. H. (2003). Statistical modeling of a full-access loss system. *Scientific Works of O.S. Popov Odesa National Academy of Telecommunications*, 1, 75–82.
11. Chen, H., Huang, L., Kumar, S., & Kuo, C. C. J. (2004). *Radio resource management for multimedia QoS support in wireless networks*. Springer.
12. Rhee, S. H., Konstantopoulos, T., Lee, H., & Chung, K. (2003). Competitive routing and flow control in communication networks of parallel links. *Computer Networks*, 42, 135–140.
13. Rosberg, Z., & Gopal, I. S. (1986). Optimal hop-by-hop flow control in computer networks. *IEEE Transactions on Automatic Control*, 31(9), 813–822.
14. Pallaschke, D. E., & Urbanski, R. (2010). *Pairs of compact convex sets: Fractional arithmetic with convex sets*. Springer.
15. Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific.
16. Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear programming: Theory and algorithms* (3rd ed.). Wiley-Interscience.
17. Nash, J. C. (1990). *Compact numerical methods for computers: Linear algebra and function minimisation* (2nd ed.). Adam Hilger.
18. Biswas, S. K., Pranab, K. M., Uttam, K., & Roy, U. K. (2023). Binary search-based fast scheduling algorithms for reliability-aware energy-efficient task graph scheduling with fault tolerance. *IEEE Transactions on Sustainable Computing*, 9(3), 433–451.
19. Karwan, M., Lotfi, V., Telgen, J., & Zionts, S. (1983). *Redundancy in mathematical programming: A state-of-the-art survey*. Springer.
20. Saaty, T. L. (1968). *Mathematical models of arms control and disarmament: Application of mathematical structures to politics*. John Wiley & Sons.