

DOI: <https://doi.org/10.15276/ict.03.2026.74>

УДК 004.8:004.7

Аналіз моделей розрахунку характеристик комп'ютерних мереж у задачах управління їх структурою протягом життєвого циклу

Науменко Роман Іванович¹⁾

ORCID: <https://orcid.org/0009-0003-7612-1773>; 9530675@stud.op.edu.ua

Нестеренко Сергій Анатолійович¹⁾

ORCID: <https://orcid.org/0000-0002-3757-6594>; san@op.edu.ua

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

У статті розглянуто проблему вибору моделей розрахунку характеристик комп'ютерних мереж у задачах управління їх структурою протягом життєвого циклу. Запропоновано розмежовувати стадію життєвого циклу системи та її структурно-конфігураційний стан, що дає змогу окремо описувати контекст розвитку мережі та конкретні структурні зміни. Дослідження має оглядово-методологічний характер; літературну базу сформовано на основі структурованого тематичного пошуку та аналізу фундаментальних і сучасних праць з моделювання характеристик мереж, аналітичних, нейромережевих, графових і генеративних підходів. Вибір моделі запропоновано здійснювати з урахуванням характеру структурно-конфігураційної зміни, доступності та структури даних, необхідного типу оцінки або гарантії та допустимого обчислювального бюджету. Проаналізовано можливості моделей теорії масового обслуговування, мережевого числення (Network Calculus), класичних нейромережевих, графових нейромережевих і дифузійних моделей. Запропоновано їх функціональний поділ на ролі генератора, оцінювача, верифікатора та оптимізатора, а також операційну процедуру переходу від параметрів задачі до вибору відповідної ролі та класу моделі. Показано, що придатність моделі до перевірки встановлених цільових показників рівня обслуговування (Service Level Objectives, SLO) визначається не лише точністю прогнозування, а й відповідністю типу її вихідного результату формі вимоги, що перевіряється. На ілюстративному прикладі модернізації перевантаженого каналу продемонстровано послідовність оцінювання кандидатних конфігурацій, перевірки SLO та вибору допустимого структурного рішення. Додатково проведено обмежену ретроспективну перевірку запропонованої процедури на опублікованих сценаріях застосування GNN, мережевого числення та дифузійних моделей; у розглянутих випадках визначені процедурою функціональні ролі узгоджувалися з фактичним способом використання моделей у відповідних дослідженнях. Обґрунтовано доцільність комбінування аналітичних і моделей на основі даних (data-driven models) у межах єдиного процесу підтримки структурних рішень. Запропонований підхід може використовуватися для вибору та поєднання моделей під час оцінювання альтернативних станів комп'ютерної мережі протягом її життєвого циклу.

Ключові слова: комп'ютерні мережі; життєвий цикл; структурно-конфігураційний стан; цільові показники рівня обслуговування (SLO); теорія масового обслуговування; мережеве числення (Network Calculus); графові нейронні мережі; дифузійні моделі; оптимізація мережі

Для цитування: Науменко Р. І., Нестеренко С. А. «Аналіз моделей розрахунку характеристик комп'ютерних мереж у задачах управління їх структурою протягом життєвого циклу». *Інформатика. Культура. Техніка.* 2026; Том 3, № 2 (4): 959–977. DOI: <https://doi.org/10.15276/ict.03.2026.74>

Актуальність. Комп'ютерні мережі є складними програмно-апаратними системами, структура та характеристики яких змінюються протягом життєвого циклу. Зростання кількості абонентів, зміна інтенсивності й структури трафіку, модернізація обладнання, зміна пропускної здатності каналів, маршрутів або топології призводять до переходу мережі між послідовними структурно-конфігураційними станами. Кожний такий перехід може змінювати затримку передавання, втрати пакетів, завантаження каналів, пропускну здатність та доступність мережевих сервісів. Тому прийняття рішень щодо розвитку мережі потребує попереднього оцінювання характеристик її нового стану та перевірки виконання встановлених вимог до рівня обслуговування.

При цьому доцільно розрізняти два рівні розвитку мережі. Перший визначається стадією життєвого циклу системи й характеризує загальний контекст її створення, використання, підтримки або виведення з експлуатації. Другий відповідає конкретному структурно-конфігураційному стану мережі, що визначається її апаратною структурою, параметрами каналів, маршрутизацією та структурою мережевого трафіку. У межах однієї стадії життєвого циклу мережа може багаторазово змінювати свою конфігурацію, тому зміна структурного стану не є тотожною переходу між стадіями життєвого циклу.

Для оцінювання наслідків таких змін використовуються різні класи моделей. До традиційних підходів належать аналітичні моделі, моделі теорії масового обслуговування, мережеве числення та імітаційне моделювання. Розвиток методів машинного навчання доповнив їх нейромережевими моделями, зокрема графовими нейронними мережами, здатними враховувати взаємозв'язок між топологією, маршрутизацією та трафіковим навантаженням. Окремий напрям становлять генеративні моделі, зокрема дифузійні, які можуть застосовуватися для формування трафікових або топологічних сценаріїв і як складова процедур оптимізаційного пошуку.

Різні класи моделей суттєво відрізняються вимогами до вихідних даних, обчислювальною вартістю, інтерпретованістю та характером отриманого результату. Аналітичні моделі можуть забезпечувати швидку оцінку або визначення граничних характеристик, тоді як навчені моделі дозволяють відтворювати складні нелінійні залежності за наявності достатнього обсягу даних. Графові моделі забезпечують явне врахування топологічного контексту, а генеративні – формування нових сценаріїв, проте сам факт генерації сценарію не означає підтвердження його допустимості.

Отже, у задачах управління структурою мережі проблема полягає не лише у виборі математично найточнішої моделі. Необхідно враховувати стадію життєвого циклу системи, характер передбачуваної зміни, доступність даних, тип необхідної оцінки або гарантії та допустимі обчислювальні витрати. Особливо важливим це є у випадках, коли результат моделювання використовується не лише для прогнозування характеристик, а для прийняття рішення щодо допустимості структурної зміни мережі.

Аналіз останніх досліджень і виділення невирішеної частини проблеми
Моделювання характеристик комп'ютерних мереж охоплює широкий спектр підходів, які відрізняються математичною основою, деталізацією та призначенням. У сучасному огляді моделей продуктивності мереж систематизовано імітаційні, аналітичні, машинні та гібридні підходи та показано, що їх застосування пов'язане з компромісом між точністю, обчислювальною вартістю, областю застосування й характером доступних даних [3]. Така систематизація достатньо повно характеризує технології моделювання характеристик мережі, але орієнтується передусім на властивості самих моделей і не розглядає їх функціональні ролі в процесі прийняття рішень щодо зміни структури мережі.

Інший напрям досліджень пов'язаний із застосуванням методів машинного навчання на різних стадіях життєвого циклу мережевого слайсування (network slicing). У роботах [4, 5] задачі й методи машинного навчання систематизовано відповідно до стадій життєвого циклу мережевого слайса. Ці результати підтверджують доцільність урахування контексту життєвого циклу при виборі методів. Однак об'єктом таких досліджень є логічно сформований мережевий слайс, ресурси якого розподіляються в межах наявної фізичної інфраструктури. У цій роботі розглядається інша постановка – зміна самої структурно-конфігураційної основи комп'ютерної мережі, яка може передбачати додавання або вилучення вузлів і каналів, заміну обладнання, зміну фізичної пропускної здатності та інші структурні рішення. Такі зміни пов'язані не лише з перепризначенням ресурсів, а й із дискретними витратами переходу між фізичними конфігураціями та, як правило, меншою оборотністю прийнятих рішень. Отже, характер помилки прогнозування та вимоги до підтвердження допустимості структурного рішення відрізняються від задач керування віртуальними мережевими слайсами.

Близькою до розглядуваної проблематики є концепція цифрового двійника мережі (Network Digital Twin, NDT). У референсній архітектурі NDT передбачається використання цифрового представлення мережі для аналізу її стану, виконання аналізу сценаріїв та підтримки управління на основі даних [6]. Такий підхід задає загальний замкнений контур взаємодії даних, моделей і системи управління, але сам по собі не визначає універсальних критеріїв вибору конкретного класу моделі залежно від типу структурної зміни, режиму доступності даних і характеру потрібної гарантії.

Окремі роботи демонструють доцільність комбінування моделей із різними властивостями. Зокрема, в DeepNC графова нейронна модель використовується для швидкої попередньої оцінки кандидатних конфігурацій, тоді як мережеве числення застосовується для подальшої формальної перевірки перспективних варіантів [7]. Такий підхід показує, що навчена й формальна моделі можуть не конкурувати між собою, а виконувати різні функції в одному процесі прийняття рішення. Разом із тим це рішення орієнтоване на конкретну задачу й не формує загальної рольової схеми для різних класів моделей і різних типів структурних змін комп'ютерної мережі.

Таким чином, наявні дослідження достатньо детально характеризують окремі класи моделей, застосування, орієнтовані на життєвий цикл машинного навчання в окремих мережевих доменах і приклади гібридного використання моделей. Проте недостатньо систематизованим залишається зв'язок між стадією життєвого циклу мережі, характером структурної зміни, режимом доступності даних, типом необхідної гарантії, допустимим обчислювальним бюджетом та функціональною роллю моделі. Це визначає необхідність формування узгодженого підходу до вибору й комбінування моделей у процесі розвитку комп'ютерної мережі.

Мета статті та методика дослідження. Метою статті є систематизація моделей розрахунку характеристик комп'ютерних мереж за їх функціональною роллю в процесі управління структурними змінами та визначення умов доцільності застосування різних класів моделей протягом життєвого циклу мережі.

Для досягнення поставленої мети необхідно формалізувати співвідношення між стадією життєвого циклу та структурно-конфігураційним станом мережі; визначити характер переходів між такими станами; встановити місце вимог SLA/SLO у процедурі перевірки нового стану; проаналізувати можливості та обмеження аналітичних, нейромережевих, графових і дифузійних моделей; а також визначити їх функціональні ролі у процесах генерації, оцінювання, верифікації та оптимізації структурних рішень.

Методика відбору джерел. Дослідження має оглядово-методологічний характер, тому для формування літературної бази застосовано структурований тематичний пошук із подальшим аналізом посилань відібраних робіт. Як базове вторинне джерело використано актуальний огляд моделювання характеристик мережі [3], у якому систематизовано моделі, ідентифіковані авторами шляхом пошуку в IEEE Xplore, ACM Digital Library та ScienceDirect із подальшим аналізом прямих і зворотних цитувань. Для актуалізації та розширення корпусу щодо окремих класів моделей додатково проведено структурований пошук у arXiv станом на 25.09.2026.

Пошук здійснювався за вісьмома тематичними комбінаціями: “network performance modeling”, RouteNet, “time-sensitive network” AND “network calculus”, “network slicing” AND “life cycle” AND “machine learning”, “network digital twin” AND management, “network traffic generation” AND diffusion, “ad hoc network” AND topology AND diffusion та “generative diffusion models” AND “network optimization”. Сукупно отримано 76 пошукових повернень. Основний часовий інтервал відбору становив 2020–2026 рр.; більш ранні роботи включалися у випадку їх фундаментального значення для відповідного класу моделей.

До аналізу включалися публікації, у яких розглядається оцінювання або прогнозування характеристик комп'ютерних мереж, визначення граничних характеристик, генерація мережевого трафіку чи топологічних конфігурацій, сценарний аналіз або підтримка оптимізаційних рішень. Виключалися роботи, що стосувалися некомунікаційних типів мереж, а також дослідження, орієнтовані виключно на класифікацію трафіку або виявлення аномалій без зв'язку із задачами структурного управління. Додатково застосовано аналіз посилань у відібраних оглядових і фундаментальних роботах, а нормативні документи відбиралися з офіційних ресурсів ISO та IETF. У результаті сформовано корпус із 21 джерела, що охоплює нормативну основу життєвого циклу, аналітичні, нейромережеві, графові та генеративні моделі, а також приклади їх комбінованого використання.

У роботі стадія життєвого циклу не розглядається як безпосередній критерій вибору моделі. Вона визначає контекст задачі, який подається у вигляді:

$$\Phi_l \rightarrow (\Delta SC, D, G, B), \quad (1)$$

де Φ_l – фаза життєвого циклу; ΔSC – характер зміни структурно-конфігураційного стану мережі; D – режим доступності даних; G – необхідний тип гарантії щодо характеристик нового стану; B – допустимий обчислювальний бюджет.

На основі цієї сукупності характеристик визначається функціональна роль моделі та доцільний клас методів. У роботі виділяються чотири функціональні ролі: генератор (generator) – формування кандидатних станів або сценаріїв; оцінювач (evaluator) – оцінювання характеристик кандидата; верифікатор (verifier) – перевірка виконання заданих обмежень; оптимізатор (optimizer) – пошук допустимого рішення відповідно до заданого критерію оптимізації. Порівняння моделей спрямоване не на визначення універсально найкращого класу, а на встановлення умов, за яких окремі моделі або їх комбінації є доцільними для конкретної задачі управління мережею.

Формалізація фаз життєвого циклу та структурно-конфігураційних станів комп'ютерної мережі. Комп'ютерна мережа як програмно-апаратна система змінюється протягом усього періоду свого існування. Для формалізації цього процесу в роботі розрізняються два рівні опису: стадія життєвого циклу системи та структурно-конфігураційний стан мережі.

Як нормативну основу для опису життєвого циклу використано положення ISO/IEC/IEEE 24748-1, що визначає загальні принципи керування життєвим циклом систем і допускає адаптацію моделі життєвого циклу до конкретного предметного середовища [1]. Як допоміжне джерело використовується ISO/IEC/IEEE 15288, який визначає процеси життєвого циклу системи та не встановлює єдиної обов'язкової послідовності їх виконання [2].

Для формалізації макrorівня введемо множину стадій життєвого циклу:

$$\Phi = \{\Phi_C, \Phi_D, \Phi_P, \Phi_U, \Phi_S, \Phi_R\}, \quad (2)$$

де Φ_C відповідає стадії Concept; Φ_D – Development; Φ_P – Production; Φ_U – Utilization; Φ_S – Support, а Φ_R – Retirement [2].

Стадії Utilization і Support відповідно до [2] можуть виконуватися паралельно; у табл. 1 вони подані окремо, оскільки відрізняються характером типових структурних змін і складом доступних даних. Стадія Φ_l характеризує макrorівень розвитку системи, однак не визначає однозначно поточну конфігурацію мережі.

Для опису мікрорівня вводиться структурно-конфігураційний стан SC_i , який характеризує апаратну структуру та структуру мережевого трафіку:

$$SC_i = \{SC_i^{HW}, SC_i^{TR}\}, \quad (3)$$

де SC_i^{HW} – структура апаратних компонентів мережі, а SC_i^{TR} – структура мережевого трафіку. Апаратну складову можна представити у вигляді:

$$SC_i^{HW} = \{V_i, E_i, P_i^{HW}\}, \quad (4)$$

де V_i – множина абонентів і комунікаційних пристроїв мережі; E_i – множина каналів зв'язку між ними; P_i^{HW} – множина параметрів апаратної структури, зокрема пропускних здатностей каналів, характеристик мережевих пристроїв та інших параметрів, що впливають на функціонування мережі.

Структуру мережевого трафіку подамо як:

$$SC_i^{TR} = \{F_i, P_i^{TR}\}, \quad (5)$$

де F_i – множина мережевих потоків, а P_i^{TR} – множина їх параметрів, до яких можуть належати інтенсивність, обсяг переданих даних, часові характеристики, маршрути проходження та інші параметри, необхідні для розрахунку характеристик функціонування мережі.

У межах однієї стадії життєвого циклу мережа може проходити через послідовність структурно-конфігураційних станів:

$$SC_1 \rightarrow SC_2 \rightarrow \dots \rightarrow SC_K. \quad (6)$$

Для опису переходу введемо оператор:

$$T : (SC_i, \Delta SC_i) \rightarrow SC_{i+1}, \quad (7)$$

Тоді новий стан визначається як:

$$SC_{i+1} = T(SC_i, \Delta SC_i), \quad (8)$$

де

$$\Delta SC_i = \{\Delta SC_i^{HW}, \Delta SC_i^{TR}\} \quad (9)$$

описує структурну або трафікову зміну.

Оператор T узагальнює додавання, вилучення або модифікацію компонентів стану. Наприклад, зміна множини вузлів може бути представлена як:

$$V_{i+1} = (V_i \cup V_i^+) - V_i^-, \quad (10)$$

де V_i^+ – множина вузлів, що додаються, а V_i^- – множина вузлів, що вилучаються.

$SC_i \rightarrow SC_{i+1}$ не є тотожним переходу $\Phi_i \rightarrow \Phi_{i+1}$. Протягом однієї стадії життєвого циклу мережа може багаторазово змінювати конфігурацію, маршрутизацію, пропускну здатність каналів і структуру трафіку.

Стадія життєвого циклу впливає на умови, за яких виконується вибір моделі відповідно до (1). Конкретизація цієї проєкції наведена в Таблиці 1.

Праві стовпці Таблиці 1 є запропонованою в роботі аналітичною проєкцією стадій життєвого циклу на задачі моделювання комп'ютерної мережі та не є дослівним змістом стандарту [1].

Така проєкція показує, що стадія життєвого циклу не визначає конкретний клас моделі безпосередньо. Вона задає простір потенційних змін, доступність даних, вимоги до результату моделювання та допустимий режим обчислень. Ці характеристики надалі використовуються як критерії вибору функціональної ролі та класу моделі.

SLO-обмеження та постановка задачі структурної оптимізації. Перехід мережі до нового структурно-конфігураційного стану є допустимим лише за умови, що характеристики її функціонування відповідають установленим вимогам до рівня обслуговування. У цьому контексті необхідно розрізнити поняття SLA, SLO та SLI. SLA визначає загальні зобов'язання щодо рівня сервісу, SLO – конкретну кількісну ціль, а SLI – фактично вимірюваний показник, за яким оцінюється виконання відповідного SLO.

Нехай $X_j(SC)$ позначає j -ту характеристику функціонування мережі в стані SC , яка в загальному випадку може мати стохастичну природу. Для перевірки відповідної вимоги використовується показник

$$Q_j(SC) = f_j(X_j(SC)) \quad (11)$$

де $f_j()$ задає спосіб отримання величини, яка безпосередньо зіставляється з установленим SLO.

Таблиця 1. Проекція стадій життєвого циклу на умови вибору моделі

Стадія	Характер можливих ΔSC	Режим даних D	Тип оцінки або гарантії G	Обчислювальний режим B
Concept	Альтернативні топології, ємність, вибір технологій	Експлуатаційні дані цільової мережі відсутні	попередня оцінка здійсненності, аналітичні або граничні оцінки	переважно офлайн-режим
Development	Уточнення топології, маршрутів, параметрів обладнання та конфігурації	Імітаційне моделювання, дані прототипу, дані тестового стенда та дані аналогічних систем	Прогнозування, перевірка виконання вимог, підтвердження адекватності	Офлайн-режим / пакетний режим
Production	Інтеграція компонентів, початкове конфігурування, допустимі заміни	Інтеграція та тестові дані	Перевірка відповідності проєктним характеристикам	Переважає офлайн-режим
Utilization	Трафік, маршрутизація, пропускна здатність, параметри конфігурації	Експлуатаційна телеметрія та SLI	Прогнозування ключових показників, контроль SLO, сценарний аналіз	Режим, близький до реального часу / офлайн-режим
Support	Заміна обладнання, резервування, розширення пропускної здатності, реконфігурація	Телеметрія + історія технічного обслуговування та відмов	Прогноз сценарного аналізу, валідація, безперервність функціонування	Режим, близький до реального часу / пакетний режим
Retirement	Вилучення вузлів і каналів, міграція, консолідація	Накопичені експлуатаційні та конфігураційні дані	Перевірка допустимості залишкової конфігурації та ризиків переходу	Офлайн-режим

Джерело: розроблено авторами

Залежно від типу вимоги f_j може визначати математичне сподівання характеристики, квантиль заданого рівня, ймовірність перевищення порогового значення, детерміновану верхню межу або іншу характеристику розподілу. Отже, тип SLO безпосередньо визначає, який вид результату повинна формувати модель.

Множину контрольованих характеристик позначимо:

$$J = J_{max} \cup J_{min}, \quad (12)$$

де J_{max} – множина показників із верхньою допустимою межею, а J_{min} – множина показників із нижньою допустимою межею.

Для j з множини J_{max} умова допустимості має вигляд:

$$Q_j(SC) \leq Q_j^{SLO} \quad (13)$$

а для j з множини J_{min} виконується:

$$Q_j(SC) \geq Q_j^{SLO}. \quad (14)$$

До першої групи можуть належати, наприклад, затримка, джитер або частка втрат пакетів, тоді як до другої – пропускна здатність, доступність або інші характеристики, для яких більше значення відповідає кращому рівню обслуговування.

Наприклад, якщо характеристика з індексом j відповідає часу виконання мережевої транзакції абонента a , то

$$Q_j(SC) = T_a(SC), \quad (15)$$

а відповідна вимога має вигляд

$$T_a(SC) \leq T_a^{SLO}, \quad (16)$$

де $T_a(SC)$ – значення часу виконання транзакції в стані SC , а T_a^{SLO} – його максимально допустиме значення.

Введемо бінарний індикатор допустимості стану

$$S(SC) = \begin{cases} 1, & \text{якщо всі встановлені SLO виконуються,} \\ 0, & \text{якщо хоча б одне SLO порушено.} \end{cases} \quad (17)$$

Множину станів, що відповідають установленим SLO, позначимо

$$A_{SLO} = \{SC : S(SC) = 1\}. \quad (18)$$

Множина $R_{\Phi_l}(SC_i)$ містить структурно-конфігураційні стани, до яких може бути здійснений перехід зі стану SC_i у межах типів структурних змін, допустимих у поточному контексті життєвого циклу. Позначимо через M_{Φ_l} множину таких змін для стадії Φ_l . Її зміст визначається відповідним рядком табл. 1, зокрема характером можливих ΔSC . Тоді

$$R_{\Phi_l}(SC_i) = \{T(SC_i, \Delta SC) : \Delta SC \in M_{\Phi_l}\}. \quad (19)$$

Таким чином, індекс Φ_l відображає залежність простору досяжних станів від стадії життєвого циклу: зміна контексту життєвого циклу може змінювати як доступні типи ΔSC , так і множину кандидатних станів, що підлягають подальшому оцінюванню.

Тоді задача структурної оптимізації набуває вигляду

$$SC^* = \arg \min_{SC \in R_{\Phi_l}(SC_i) \cap A_{SLO}} C(SC_i, SC), \quad (20)$$

де $C(SC_i, SC)$ – вартість переходу від поточного стану SC_i до кандидатного стану SC .

Таке представлення дозволяє враховувати не лише вартість кінцевої конфігурації, а саме витрати на структурну зміну: додавання або заміну обладнання, зміну пропускну здатності, реконфігурацію маршрутів, створення резервних з'єднань та інші дії.

Вид результату, який формує модель, повинен відповідати формі SLO, що перевіряється. Нехай для характеристики X_j встановлена ймовірнісна вимога

$$P\left(\frac{X_j(SC)}{C(SC_i, SC)} \leq x_j^{SLO}\right) \geq 1 - \epsilon_j, \quad (21)$$

де ϵ_j – допустима ймовірність порушення відповідного SLO. Використовувана модель реалізує функцію:

$$f_j(X_j) = E[X_j] C(SC_i, SC), \quad (22)$$

і, відповідно, формує лише точкову оцінку:

$$Q_j(SC) = E[X_j(SC)]. \quad (23)$$

Без додаткової інформації про розподіл $X_j(SC)$ така оцінка не дозволяє зробити висновок про виконання заданої ймовірнісної вимоги або про дотримання детермінованої верхньої межі. Отже, визначальним є не лише рівень похибки моделі, а й відповідність типу її результату типу вимоги, що перевіряється.

Це дозволяє формально розмежувати дві функціональні ролі. Оцінювач визначає або прогнозує необхідні характеристики кандидатного стану, тоді як верифікатор встановлює, чи є отриманий результат достатнім для підтвердження конкретної вимоги. Один метод може поєднувати обидві ролі лише тоді, коли тип сформованого ним результату безпосередньо дозволяє перевірити відповідний SLO.

оцінювач $\neq$ верифікатор.

Таким чином, підставою для розмежування цих ролей є не сам спосіб побудови моделі, а можливе неспівпадіння між типом оцінки, яку вона формує, і типом гарантії, необхідної для перевірки SLO.

Водночас це розмежування не означає, що для кожної задачі обов'язково потрібні дві різні моделі. Якщо оцінювач безпосередньо формує потрібний квантиль, ймовірнісну оцінку або гарантовану межу, його результат може одночасно використовуватися для верифікації відповідного SLO.

Не всі SLO мають ймовірнісну або перцентильну форму. Частина вимог може визначатися середніми значеннями, коефіцієнтами доступності або іншими показниками. Тому необхідність окремої функції верифікатора визначається семантикою конкретного SLO та типом результату, який формує модель.

Окремо необхідно врахувати випадок відсутності допустимого структурного рішення. Якщо

$$R_{\Phi I}(SC_i) \cap A_{SLO} = \emptyset, \quad (24)$$

задача оптимізації не має розв'язку в межах поточного простору структурних рішень. Це означає, що жодна зміна з множини $M_{\Phi I}$, доступної в поточному контексті життєвого циклу, не приводить до стану, який задовольняє встановлені SLO. У такому випадку може знадобитися розширення множини допустимих структурних змін, перегляд вимог, суттєвіша реконфігурація мережі або перехід до іншого набору процесів життєвого циклу. Отже, контекст життєвого циклу визначає простір структурного пошуку, але результати оптимізації, своєю чергою, можуть вказувати на недостатність цього простору.

Для компактного представлення всієї множини контрольованих характеристик введемо

$$X(SC) = \{X_j(SC) : j \in J\}, \quad Q(SC) = \{Q_j(SC) : j \in J\}. \quad (25)$$

Тоді процес оцінювання структурної зміни можна представити як

$$SC_i \xrightarrow{T} SC_{i+1} \rightarrow X(SC_{i+1}) \rightarrow Q(SC_{i+1}) \rightarrow S(SC_{i+1}). \quad (26)$$

Якщо $S(SC_{i+1}) = 1$ кандидатний стан може бути прийнятий з погляду встановлених SLO. Якщо $S(SC_{i+1}) = 0$ необхідний пошук іншого стану в межах $R_{\Phi I}(SC_i)$ або розширення самого простору структурних рішень.

Таким чином, ефективність управління структурою мережі залежить не лише від точності прогнозування її характеристик, а й від відповідності типу результату моделі тій формі оцінки або гарантії, яка потрібна для конкретного SLO. Це створює основу для подальшого аналізу класів моделей відповідно до їх здатності виконувати функції оцінювання, верифікації, генерації та оптимізації структурних рішень.

Аналіз моделей оцінювання та підтримки структурних рішень. Відповідно до сформованої постановки, модель характеристик мережі повинна розглядатися не ізольовано, а з урахуванням функції, яку вона виконує під час переходу між структурно-конфігураційними станами. Різні класи моделей відрізняються типом вхідних даних, характером одержуваного результату, обчислювальною вартістю та можливістю врахування топологічних і трафікових властивостей мережі. Сучасні підходи до моделювання характеристик мережі охоплюють аналітичні моделі, дискретно-подієве моделювання, методи машинного навчання та їх гібридні комбінації [3].

У межах цієї роботи основна увага приділяється моделям теорії масового обслуговування, мережевому численню, класичним нейромережевим моделям, графовим нейронним мережам і дифузійним моделям. Імітаційні моделі використовуються як референсний підхід для детального оцінювання та валідації, але їх розгорнутий аналіз виходить за межі роботи.

Моделі теорії масового обслуговування. Теорія масового обслуговування є традиційним аналітичним засобом оцінювання характеристик комп'ютерних і телекомунікаційних мереж. У такому представленні мережеві пристрої, канали або інші ресурси розглядаються як системи обслуговування, а пакети, запити чи потоки — як заявки. Це дає змогу аналітично визначати затримку, довжину черги, завантаження ресурсів, імовірність блокування та інші характеристики функціонування мережі [3].

Для найпростішої одноканальної системи $M/M/1$ коефіцієнт використання ресурсу визначається як

$$u = \frac{\lambda}{\mu}, \quad (27)$$

де λ — середня інтенсивність надходження заявок, а μ — середня інтенсивність їх обслуговування. За умови $u < 1$ середній час перебування заявки в системі становить

$$W = \frac{1}{\mu - \lambda}. \quad (28)$$

Такі співвідношення демонструють основну перевагу ТМО — можливість безпосередньо встановити зв'язок між параметрами навантаження, ресурсами мережі та її характеристиками без формування навчального датасету. Це робить аналітичні моделі особливо корисними на стадіях Concept і Development, коли експлуатаційні дані для нової конфігурації ще відсутні. На стадіях Utilization і Support їх параметри можуть уточнюватися за фактичною телеметрією.

Водночас точність ТМО залежить від того, наскільки прийняті припущення відповідають властивостям реального трафіку. Найпростіші марковські моделі часто спираються на пуассонівські процеси надходження та експоненційні розподіли часу обслуговування, тоді як реальний мережевий трафік може характеризуватися пачковістю, кореляцією та довготривалою залежністю [18]. Для кращого опису таких властивостей застосовуються складніші марковські процеси, однак це одночасно підвищує складність аналітичної моделі [8], [9], [10].

Отже, моделі ТМО доцільно розглядати насамперед як оцінювач, що забезпечує швидке й інтерпретоване оцінювання характеристик мережі за умови прийнятої адекватності математичних припущень. Їх результат не слід автоматично трактувати як гарантію виконання SLO: можливість використання конкретної моделі як верифікатор залежить від типу отримуваної оцінки та форми відповідної вимоги.

Мережеве числення (Network Calculus). Мережеве числення є аналітичним підходом до визначення граничних характеристик потоків і систем обслуговування в комп'ютерних мережах. У детермінованому варіанті вхідний трафік описується кривою надходження $a(t)$, а доступний мережевий ресурс — кривою обслуговування $B(t)$. На основі співвідношення цих функцій можуть визначатися верхні межі затримки та обсягу накопичених у черзі даних [11].

Зокрема, верхню межу затримки можна подати як

$$d_{\max} = \sup_{t \geq 0} \inf \{d \geq 0 : \alpha(t) \leq \beta(t+d)\}. \quad (29)$$

На відміну від моделей, результатом яких є середнє або прогнозоване значення характеристики, Мережеве числення дозволяє отримувати детерміновану граничну оцінку за виконання прийнятих припущень. Тому такий результат може безпосередньо зіставлятися з SLO, заданим у вигляді максимально допустимої затримки або іншого граничного показника.

Сучасні застосування мережевого числення, зокрема для мережі, що чутливі до часу (Time-Sensitive Networks, TSN), також використовують цей підхід для визначення гарантованих меж затримки [12].

Як і моделі теорії масового обслуговування, мережеве числення не потребує формування навчального датасету. Водночас отримані межі можуть бути консервативними, а аналіз великих топологій і складної взаємодії потоків – вимагати додаткових апроксимацій. У запропонованій рольовій схемі мережевого числення доцільно розглядати насамперед як верифікатор, оскільки тип сформованого ним результату безпосередньо відповідає вимогам, для яких необхідне підтвердження граничного значення характеристики.

Класичні нейромережеві моделі. Класичні нейромережеві моделі застосовуються для апроксимації складних нелінійних залежностей між параметрами мережевого стану та характеристиками функціонування мережі. До цієї групи можна віднести багатопарові перцептрони, рекурентні мережі, LSTM/GRU та інші моделі, що працюють із векторними або послідовними представленнями телеметрії. Після навчання такі моделі здатні швидко оцінювати затримку, завантаження, пропускну здатність та інші KPI для станів, подібних до представлених у навчальних даних.

На відміну від аналітичних моделей, нейромережі не потребують явного задавання закону надходження трафіку або аналітичної залежності між параметрами системи. Це дає змогу враховувати складні взаємозв'язки в телеметричних даних, однак переносить основні вимоги з математичних припущень на якість і репрезентативність навчального датасету. Тому їх застосування є найбільш природним на стадіях Utilization і Support, коли вже накопичено достатній обсяг експлуатаційних даних.

Основним обмеженням класичних нейромережевих моделей у задачах структурної зміни мережі є відсутність явного представлення топології. Конфігурація мережі зазвичай подається у вигляді фіксованого вектора ознак, унаслідок чого зміна кількості вузлів, каналів або їх взаємозв'язків може вимагати зміни вхідного представлення та повторного навчання моделі. Це обмежує їх придатність до сценарного аналізу станів із суттєво новою структурою.

У запропонованій рольовій схемі класичні нейромережеві моделі виконують насамперед функцію оцінювача. Вони можуть забезпечувати швидке прогнозування характеристик мережі за наявності достатнього навчального набору, проте сам по собі точковий прогноз не є формальною гарантією виконання SLO. Тому для вимог, що потребують детермінованої або ймовірнісної верифікації, результат такої моделі має доповнюватися відповідним верифікатором.

Графові нейронні мережі. Графові нейронні мережі є природним розвитком підходів на основі даних для задач, у яких структура комп'ютерної мережі безпосередньо впливає на її характеристики. На відміну від класичних нейромережевих моделей із фіксованим векторним входом, GNN опрацьовують мережу як граф, у якому вузли й канали зв'язку можуть мати власні ознаки, а процес обміну повідомленнями між елементами графа дозволяє враховувати топологічні залежності [13].

Для оцінювання характеристик мережі вхід GNN може містити топологію, параметри каналів, маршрутизацію та характеристики трафіку. Такий підхід реалізовано, зокрема, у сімействі RouteNet, де графові моделі використовуються для прогнозування мережевих KPI на основі взаємодії топології, маршрутів і навантаження [14], [15]. Це робить GNN придатними до сценарного оцінювання кандидатних структурно-конфігураційних станів, коли зміна $SC_i \rightarrow SC_{i+1}$ одночасно змінює структуру графа та умови проходження трафіку.

Важливою перевагою GNN є те, що структура моделі не прив'язана безпосередньо до фіксованої кількості вузлів або каналів. У роботах сімейства RouteNet продемонстровано можливість застосування навчених моделей до топологій і конфігурацій, які не були представлені в навчальному наборі [14], [15]. Однак така здатність не є автоматичною властивістю будь-якої GNN: якість перенесення залежить від архітектури, способу представлення графа та різноманітності навчальних даних.

З погляду життєвого циклу GNN найбільш доцільні тоді, коли вже доступні достатні дані про взаємозв'язок між структурою мережі, трафіком і спостережуваними характеристиками. Тому їх застосування є природним для Development, Utilization і Support, а за наявності синтетичних або імітаційних даних – також для попереднього аналізу нових конфігурацій.

У запропонованій рольовій схемі GNN насамперед виконує функцію оцінювача, забезпечуючи швидке оцінювання характеристик великої кількості кандидатних станів. Водночас така модель може використовуватися як наближена модель для швидкого оцінювання усередині оптимізатора, прискорюючи пошук структурного рішення. Як і для класичних нейромереж, прогноз GNN сам по собі не є формальною гарантією виконання SLO, тому для виконання ролі верифікатора потрібен результат відповідного типу або додатковий метод перевірки.

Дифузійні моделі. Дифузійні моделі належать до генеративних моделей, у яких нові зразки формуються шляхом послідовного відновлення даних із зашумленого стану. Після навчання модель може генерувати реалізації, статистично узгоджені з навчальним розподілом, а в умовному варіанті – з урахуванням заданих параметрів або обмежень [16], [17].

У задачах управління комп'ютерними мережами доцільно розрізняти три напрями їх використання.

Перший – генерація мережевого трафіку, коли результатом є синтетичні трафікові послідовності або сценарії навантаження [19].

Другий – генерація структурних кандидатів, зокрема альтернативних топологій або наборів параметрів конфігурації [20].

Третій – використання дифузійної моделі як складової оптимізаційного пошуку, коли вона формує або спрямовує множину кандидатних станів, що надалі оцінюються іншими моделями [21]. У перших двох випадках основною функціональною роллю дифузійної моделі є генератор, у третьому – компонент оптимізатора.

При цьому правдоподібність згенерованого кандидата відносно навчального розподілу не означає його фізичної або експлуатаційної допустимості. Для топологічних кандидатів необхідно перевіряти, зокрема, зв'язність мережі, допустимість вузлів і каналів та виконання заданих структурних обмежень. Для трафікових сценаріїв мають виконуватися допустимі діапазони інтенсивностей, параметрів потоків і ресурсних обмежень. Кандидати, що порушують такі умови, виключаються до подальшого оцінювання.

Після перевірки структурної допустимості характеристики кандидатного стану визначає оцінювач, а виконання встановленого SLO підтверджує незалежний верифікатор відповідного типу. Залежно від форми вимоги таким верифікатором може бути, наприклад, мережеве числення для детермінованих граничних характеристик або інший метод, результат якого безпосередньо відповідає формі SLO. Отже, дифузійна модель сама по собі в загальному випадку не є верифікатором і не повинна використовуватися для підтвердження допустимості сформованого рішення лише на основі ймовірності або якості генерації.

Порівняльний аналіз функціональних ролей моделей. Проведений аналіз показує, що розглянуті класи моделей не є взаємозамінними. Їх доцільність визначається не лише точністю оцінювання, а відповідністю між характером структурної зміни ΔSC , доступними даними D , необхідним типом оцінки або гарантії G , обчислювальним бюджетом B та функціональною роллю моделі. Узагальнення цих властивостей наведено в Таблиці 2.

Операціоналізація критеріїв вибору моделі. Для переходу від порівняльної характеристики класів моделей до практичної процедури їх вибору параметри контексту задачі, введені у (1), доцільно подати у вигляді скінченної множини операційних категорій. При цьому стадія життєвого циклу Φ_i не потребує додаткової категоризації, оскільки її значення та зв'язок із характером допустимих структурних змін, режимом даних і обчислювальними умовами вже визначено в Таблиці 1. Операційні значення параметрів ΔSC , D , G і B наведено в Таблиці 3.

Таблиця 2. Відповідність класів моделей умовам задачі та їх функціональним ролям

Клас моделі	Характер доцільних ΔSC	Вимоги до даних D	Тип результату моделі	Обчислювальний режим B	Основна роль
Теорія масового обслуговування	Зміни навантаження, пропускної здатності, параметрів обслуговування	Параметри потоків і ресурсів; навчальний датасет не потрібний	Середні або ймовірнісні характеристики залежно від моделі	Низькі / помірні обчислювальні витрати	Оцінювач
Network Calculus	Зміни потоків, маршрутів і ресурсів, для яких можуть бути задані формальні обмеження	Характеристики потоків і функцій обслуговування	Детерміновані граничні оцінки	Низькі / помірні обчислювальні витрати	Перифікатор
Класичні NN	Зміни параметрів у межах представлення, близького до навчальних даних	Репрезентативний навчальний набір	Прогноз KPI або інша навчена оцінка	Витратне навчання в офлайн-режимі; швидке оцінювання після навчання	Оцінювач
GNN	Зміни топології, маршрутів, параметрів каналів і трафіку	Графове представлення та дані про характеристики мережі	Прогноз KPI для структурованого мережевого стану	Витратне навчання в офлайн-режимі; швидкий inference	Оцінювач; компонент оптимізатора
Дифузійні моделі	Формування альтернативних структурних або трафікових сценаріїв	Репрезентативні дані про цільовий розподіл та, за потреби, умови генерації	Множина кандидатних сценаріїв	Послідовне семплювання; підвищені витрати генерації	Генератор для трафікових і топологічних кандидатів; компонент оптимізатора; потребує окремого оцінювача/верифікатора для оцінки та перевірки допустимості

Джерело: розроблено авторами

Процедура вибору виконується послідовно. На першому етапі за характером задачі визначається необхідна функціональна роль моделі. Якщо необхідно сформувати множину альтернативних структурних або трафікових станів, використовується генератор. Якщо кандидатний стан уже заданий і необхідно визначити його характеристики, потрібен оцінювач. Якщо необхідно підтвердити виконання SLO результатом заданого типу, застосовується верифікатор. Якщо після оцінювання та перевірки залишається декілька допустимих станів і задано критерій вартості переходу $C(SC_i, SC)$, використовується оптимізатор. Одна задача може потребувати послідовного виконання декількох таких ролей.

На другому етапі з розглянутих у табл. 2 класів моделей відбираються ті, що сумісні з конкретними значеннями ΔSC , D , G і B . Характер ΔSC визначає придатність моделі до відповідного типу структурної зміни; D – можливість її застосування за наявного обсягу та структури даних; G – відповідність типу сформованого результату вимозі, що перевіряється; B – допустимість моделі за обчислювальними витратами. Невідповідність хоча б одному обов'язковому параметру виключає модель із подальшого розгляду для відповідної функціональної ролі.

Таблиця 3. Операційні категорії параметрів вибору моделі

Параметр	Категорія	Характеристика
ΔSC	Δ_P	Параметрична зміна навантаження, пропускної здатності, параметрів обладнання або обслуговування
	Δ_R	Зміна маршрутизації або розподілу мережевих потоків
	Δ_T	Топологічна зміна: додавання, вилучення або заміна вузлів і каналів
	Δ_A	Формування множини альтернативних структурних або трафікових станів
D	D_0	Навчальний набір відсутній; доступні параметри мережі та характеристики ресурсів
	D_1	Доступні обмежені, синтетичні, імітаційні або дані тестових стендів
	D_2	Доступний репрезентативний векторний або часовий набір експлуатаційних даних
	D_3	Доступні репрезентативні графові дані, що містять топологію, маршрути, трафік і характеристики мережі
G	G_E	Точкова, середня або інша оцінка характеристики
	G_P	Ймовірнісна характеристика: квантиль, імовірність перевищення або інша характеристика розподілу
	G_D	Детермінована гранична оцінка або результат, придатний для формальної перевірки обмеження
B	B_1	Близький до реального часу режим із жорстким обмеженням часу прийняття рішення
	B_2	Помірний пакетний режим або режим близький до реального часу
	B_3	Щфлайн-режим, у якому допускаються підвищені обчислювальні витрати

Джерело: розроблено авторами

Якщо умовам задачі відповідає декілька моделей одного функціонального типу, спочатку перевіряється відповідність результату вимозі G , далі – можливість роботи з доступними даними D та виконання обмежень B . Якщо після такого відбору залишається декілька допустимих моделей і наявні валідаційні дані, додатковим критерієм є точність оцінювання. За відсутності таких даних доцільно використовувати модель із меншими обчислювальними витратами за умови, що вона забезпечує необхідний тип результату.

Якщо відібрані моделі виконують різні функціональні ролі, вони не є взаємовиключними. У такому випадку вони можуть формувати гібридну послідовність, у якій генератор створює кандидатні стани, модель-оцінювач оцінює їх характеристики, верифікатор перевіряє виконання встановлених SLO, а оптимізатор здійснює вибір серед допустимих конфігурацій відповідно до заданого критерію.

Спільний розгляд Таблиці 1, Таблиці 2 та Таблиці 3 формує послідовну процедуру вибору моделі. Таблиця 1 визначає вплив контексту життєвого циклу на умови задачі, Таблиця 2 характеризує придатність окремих класів моделей до різних типів структурних змін і функціональних ролей, а Таблиця 3 задає операційні категорії параметрів, за якими здійснюється їх відбір. Таким чином, стадія життєвого циклу сама по собі не визначає конкретний клас моделі; вибір здійснюється через конкретну комбінацію характеру структурної зміни, доступності даних, необхідного типу результату та допустимого обчислювального режиму.

Наприклад, за відсутності експлуатаційного датасету та необхідності швидко оцінити альтернативну конфігурацію доцільним може бути використання аналітичної моделі. Якщо ж необхідно перевірити детерміновану межу затримки, перевагу має модель, здатна безпосередньо сформулювати відповідний тип гарантії. За наявності достатнього обсягу даних і

складної топологічної залежності роль оцінювача може виконувати GNN, тоді як генеративна модель може формувати множину кандидатних станів для подальшого оцінювання.

Таким чином, окремі моделі доцільно розглядати не як альтернативи, з яких потрібно обрати одну універсально найкращу, а як компоненти єдиного процесу підтримки структурного рішення. У загальному випадку такий процес може бути представлений як

генератор → оцінювач → верифікатор → оптимізатор.

Ця схема не означає, що всі чотири ролі обов'язково мають реалізовуватися різними моделями або бути присутніми в кожній задачі. Одна модель може поєднувати декілька ролей, а в простих випадках окремі етапи можуть бути відсутні. Принциповим є те, що результат кожного компонента повинен відповідати функції, яку він виконує в процесі прийняття структурного рішення.

Роль оптимізатора у запропонованій схемі можуть виконувати методи математичного програмування, евристичні та метаевристичні алгоритми, а також методи навчання з підкріпленням. Генеративні моделі можуть використовуватися як складова такого процесу, формуючи множину кандидатних структурних або трафікових рішень для подальшого оцінювання [21]. Розгорнутий аналіз алгоритмів оптимізації виходить за межі цієї роботи. У контексті запропонованої рольової схеми принциповим є те, що оптимізатор використовує результати оцінювача і верифікатора для порівняння кандидатних станів, перевірки їх допустимості та пошуку рішення відповідно до заданого критерію.

Для складних задач перспективним є гібридний підхід. Наприклад, генератор може формувати множину кандидатних SC, GNN – швидко оцінювати їх характеристики, після чого верифікатор перевіряє найперспективніші варіанти відносно SLO. Саме такий розподіл дозволяє поєднати швидкість моделей на основі даних із формальністю аналітичних методів без вимоги, щоб одна модель одночасно виконувала всі функції.

Наскрізний приклад застосування алгоритму вибору моделі. Для демонстрації запропонованої процедури розглянемо ілюстративний сценарій модернізації перевантаженого каналу в діючій комп'ютерній мережі. Нехай у поточному стані SC_i пропускна здатність каналу становить 100 Мбіт/с, тоді як середня інтенсивність трафіку відповідає 120 Мбіт/с. Задача виконується на стадії Support і полягає у виборі одного з доступних варіантів розширення пропускної здатності до 150, 200 або 300 Мбіт/с. Відповідно до Таблиці 3 контекст задачі характеризується параметричною зміною ΔP , відсутністю навчального набору D_0 , необхідністю детермінованого підтвердження максимальної затримки G_D та помірним обчислювальним режимом B_2 .

Оскільки множина кандидатних конфігурацій визначається доступними варіантами модернізації, окремий генератор у цьому сценарії не потрібний. Для попереднього оцінювання кандидатів може бути використана модель теорії масового обслуговування, яка не потребує навчального датасету, а для підтвердження граничного SLO – мережеве числення. Остаточний вибір серед допустимих станів здійснюється відповідно до критерію вартості переходу, введеного в (20).

Для розрахункового прикладу приймемо середній розмір пакета 1000 байт. Тоді інтенсивності трафіку 120 Мбіт/с відповідає $\lambda = 15000$ пакетів/с. Для кожного кандидата інтенсивність обслуговування μ визначається його пропускною здатністю. За (27) і (28) визначаються коефіцієнт використання ресурсу u та середній час перебування заявки в системі W . Для перевірки SLO трафік додатково описується кривою надходження типу «маркерного відра» (token bucket) із тривалою інтенсивністю $r = 120$ Мбіт/с і максимальною пачкою $\sigma = 2$ Мбіт, а ресурс – крива обслуговування типу «швидкість–затримка» (rate-latency) з фіксованою складовою затримки $T = 5$ мс. Встановимо вимогу $d_{\max} \leq 15$ мс. Результати оцінювання наведено в Таблиці 4.

Таблиця 4. Приклад оцінювання кандидатних структурно-конфігураційних станів

Кандидатний стан	Пропускна здатність, Мбіт/с	u	W , мс	$d_{\max}$, мс	Виконання SLO	$C(SC_i, SC)$, ум. од
$SC^{(1)}$	150	0,80	0,267	18,33	Ні	1,0
$SC^{(2)}$	200	0,60	0,100	15,00	Так	1,6
$SC^{(3)}$	300	0,40	0,044	11,67	Так	2,8

Джерело: розраховано авторами для ілюстративного сценарію

Результати попереднього оцінювання показують, що для всіх трьох кандидатних станів виконується умова $u < 1$, тобто після модернізації система є стійкою в межах прийнятої моделі М/М/1. Збільшення пропускної здатності також послідовно зменшує середній час перебування заявки в системі. Однак ці середні значення самі по собі не підтверджують виконання детермінованої вимоги щодо максимальної затримки.

Перевірка за допомогою мережевого числення відповідно до (29) показує, що для $SC^{(1)}$ гранична затримка становить 18,33 мс і перевищує встановлене SLO. Стани $SC^{(2)}$ і $SC^{(3)}$ забезпечують відповідно 15,00 і 11,67 мс та утворюють множину допустимих кандидатів. Відповідно до задачі оптимізації (20) серед них обирається $SC^* = SC^{(2)}$, оскільки цей стан задовольняє встановлений SLO і має меншу вартість переходу, ніж $SC^{(3)}$.

Наведений приклад демонструє послідовність застосування запропонованого підходу: формування множини доступних структурних змін, вибір оцінювача відповідно до режиму даних, попереднє оцінювання кандидатів, незалежна перевірка необхідного типу SLO та вибір допустимого стану за критерієм вартості. При цьому використання окремого генератора не є обов'язковим, якщо множина кандидатних конфігурацій формується безпосередньо з доступних структурних рішень.

Обмежена ретроспективна перевірка узгодженості запропонованої процедури

Для додаткової перевірки практичної узгодженості запропонованої процедури її ретроспективно застосовано до чотирьох опублікованих сценаріїв використання різних класів моделей у задачах комп'ютерних мереж: RouteNet-Fermi [15], DeepNC [7], NetDiffus [19] та NetDiff [20]. Для кожного сценарію за постановкою задачі визначалися характер необхідної зміни або дії, режим доступності даних, тип необхідного результату та функціональна роль моделі. Отриманий за запропонованою процедурою вибір порівнювався з фактичним способом використання відповідного класу моделей у вихідному дослідженні. Результати наведено в Таблиці 5.

У всіх чотирьох розглянутих випадках функціональна роль і клас моделі, визначені запропонованою процедурою, узгоджуються з фактичним способом використання моделей у відповідних дослідженнях. Зокрема, DeepNC [7] підтверджує доцільність розмежування ролей оцінювача і верифікатора: модель на основі даних використовується для попереднього оцінювання кандидатів, тоді як формальна перевірка виконується засобами мережевого числення. Аналогічно NetDiffus [19] і NetDiff [20] демонструють генеративну роль дифузійних моделей відповідно для мережевого трафіку та топологічних станів, без ототожнення процесу генерації з перевіркою допустимості отриманого результату.

Отримані результати слід розглядати як обмежену перевірку зовнішньої узгодженості, а не як статистичне підтвердження універсальності запропонованої процедури. Час прийняття рішення та точність прогнозування окремо не оцінювалися, оскільки перший показник потребує контролюваного користувацького експерименту, а другий характеризує конкретну прогнозу модель, а не процедуру вибору класу моделей.

Обговорення результатів. Проведений аналіз показує, що вибір моделі для оцінювання структурних змін комп'ютерної мережі доцільно здійснювати не за належністю методу до певного класу, а за його функціональною відповідністю конкретній задачі. Відповідно до (1), запропонована проєкція параметрів задачі дозволяє пов'язати контекст життєвого циклу

мережі з характером структурної зміни, доступністю даних, необхідним типом оцінки або гарантії та допустимим обчислювальним бюджетом. На цій основі модель може бути віднесена до однієї або декількох функціональних ролей: генератор, оцінювач, верифікатор та оптимізатор.

Таблиця 5. Ретроспективна перевірка узгодженості процедури вибору моделей

Джерело	Характер задачі	Результат запропонованої процедури	Фактичне використання моделі	Узгодженість
RouteNet-Fermi [15]	Оцінювання характеристик мережі за топологією, маршрутами та трафіком	Оцінювач; GNN	GNN використовується для прогнозування характеристик мережі	Так
DeepNC [7]	Швидке оцінювання конфігурацій із подальшою перевіркою часових обмежень	Оцінювач + верифікатор; GNN + мережеве числення	GNN використовується для попереднього оцінювання, мережеве числення — для формальної перевірки	Так
NetDiffus [19]	Формування синтетичних реалізацій мережевого трафіку	Генератор; дифузійна модель	Дифузійна модель використовується для генерації мережевого трафіку	Так
NetDiff [20]	Формування альтернативних мережевих топологій	Генератор з подальшою перевіркою допустимості; дифузійна модель	Дифузійна модель використовується для генерації топологічних кандидатів, які потребують подальшої перевірки	Так

Джерело: розроблено авторами

Таке представлення усуває необхідність розглядати аналітичні й методи на основі даних як взаємовиключні альтернативи. Модель теорії масового обслуговування або нейромережева модель може використовуватися для швидкого оцінювання характеристик стану, Мережеве числення – для перевірки граничних вимог, а генеративна модель – для формування множини кандидатних сценаріїв. За необхідності ці компоненти можуть бути об’єднані в єдиний процес підтримки структурного рішення.

Суттєвим є також розмежування точності прогнозування та можливості верифікації вимоги. Навіть модель із малою похибкою не обов’язково може підтвердити виконання SLO, якщо тип її вихідного результату не відповідає формі заданого обмеження. У введеній формалізації ця відповідність визначається способом перетворення характеристики функціонування мережі на показник, який безпосередньо використовується для перевірки SLO.

Запропонований підхід не передбачає фіксованого набору моделей для кожної стадії життєвого циклу. Стадія Φ_l визначає контекст і простір доступних структурних змін, однак конкретний клас моделі обирається відповідно до параметрів задачі. Це дозволяє застосовувати однаковий принцип як до попереднього аналізу нової конфігурації на стадіях Concept або Development, так і до реконфігурації діючої мережі на стадіях Utilization і Support.

Водночас сформований підхід має методологічний характер і не визначає універсальних числових меж, за яких один клас моделей повинен замінювати інший. Такі межі залежать від конкретної мережі, структури доступних даних, вимог SLO та допустимого часу прийняття рішення. Подальшим напрямом досліджень є експериментальна перевірка запропонованої рольової схеми для конкретних задач структурної оптимізації та оцінювання ефективності гібридних комбінацій моделей.

Висновки. У роботі сформовано підхід до вибору моделей розрахунку характеристик комп’ютерних мереж у задачах управління їх структурою протягом життєвого циклу. Запропоновано розділяти стадію життєвого циклу та структурно-конфігураційний стан

мережі, що дозволяє розглядати структурні зміни як окремі переходи в межах відповідного контексту життєвого циклу.

Встановлено, що вибір моделі доцільно здійснювати з урахуванням характеру структурної зміни, доступності даних, необхідного типу оцінки або гарантії та допустимого обчислювального бюджету. Такий підхід дозволяє уникнути прямого закріплення певного класу моделей за конкретною стадією життєвого циклу та орієнтувати вибір на властивості конкретної задачі.

Запропоновано функціональний поділ моделей на ролі генератор, оцінювач, верифікатор та оптимізатор. Моделі теорії масового обслуговування, класичні нейромережеві та графові нейромережеві моделі доцільно застосовувати переважно для оцінювання характеристик мережевого стану; Мережеве числення – для отримання граничних оцінок і перевірки відповідних обмежень; дифузійні моделі – для генерації кандидатних сценаріїв та як складову процедур оптимізаційного пошуку. Один метод при цьому може поєднувати декілька функціональних ролей.

Показано, що точність прогнозування характеристики сама по собі не визначає можливість підтвердження виконання SLO. Для верифікації необхідна відповідність між типом результату, який формує модель, та формою вимоги, що перевіряється. Це обґрунтовує доцільність комбінування аналітичних і моделей на основі даних у межах єдиного процесу підтримки структурного рішення.

Таким чином, результатом роботи є рольова схема вибору та комбінування моделей для оцінювання структурних змін комп'ютерної мережі протягом її життєвого циклу. Подальші дослідження доцільно спрямувати на експериментальну перевірку запропонованого підходу для конкретних сценаріїв реконфігурації мережі та кількісне порівняння комбінацій моделей за точністю, швидкістю оцінювання й здатністю підтверджувати встановлені SLO.

Конфлікт інтересів: автори заявляють про відсутність конфлікту інтересів

Фінансування: дослідження виконано без фінансової підтримки

Доступність даних: у межах цього дослідження окремі набори даних не створювалися та не аналізувалися. Робота має методологічний і аналітичний характер та ґрунтується на опублікованих наукових працях, стандартах і відкритих джерелах, наведених у списку літератури. Усі матеріали, необхідні для відтворення представленого аналізу та запропонованих формалізацій, наведені в тексті статті та процитованих джерелах

Використання штучного інтелекту: під час підготовки та редагування рукопису використовувався інструмент OpenAI ChatGPT (модель GPT-5.6 Sol) для мовно-стилістичного редагування, структурного аналізу тексту, перевірки логічної узгодженості, уточнення формулювань і підготовки редакційних пропозицій. Усі наукові положення, математичні формалізації, використані джерела, запропоновані класифікації та висновки були перевірені й остаточно затверджені авторами. Відповідальність за зміст роботи повністю несуть автори

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “ISO/IEC/IEEE 24748-1:2024. Systems and software engineering – Life cycle management. Part 1: Guidelines for life cycle management”. 2nd ed. *International Organization for Standardization*. Geneva. 2024. – Available from: <https://www.iso.org/standard/84709.html>. – [Accessed: 14.05.2026].
2. “ISO/IEC/IEEE 15288:2023. Systems and software engineering. System life cycle processes”. 2nd ed. *International Organization for Standardization*. Geneva. 2023. – Available from: <https://www.iso.org/standard/81702.html>. – [Accessed: 14.05.2026].
3. Güemes-Palau, C., Ferriol-Galmés, M., Paillisse-Vilanova, J., Barlet-Ros, P. & Cabellos-Aparicio, A. “From simulation to deep learning: Survey on network performance modeling approaches”. *Computer Networks*. 2026; 282: 112253. DOI: <https://doi.org/10.1016/j.comnet.2026.112253>.
4. Donatti, A. W., Corrêa, S. L., Martins, J. S. B., Abelém, A. J. G., Both, C. B., Silva, F. O., Suruagy, J. A., Pasquini, R., Moreira, R., Cardoso, K. V. & Carvalho, T. C. M. B. “Survey on machine learning-enabled network slicing: Covering the entire life cycle”. *IEEE Transactions on Network and Service Management*. 2024; 21 (1): 994–1011. DOI: <https://doi.org/10.1109/TNSM.2023.3287651>.

5. Thomatos, E., Sgora, A., Tsipis, A. & Chatzimisios, P. “AI methods in network slice life-cycle phases: A survey”. *Electronics*. 2025; 14 (20): 4053. DOI: <https://doi.org/10.3390/electronics14204053>.
6. Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M. & Jacquenet, C. “Network digital twin: Concepts and reference architecture”. *Internet-Draft Draft-Irtf-Nmrg-Network-Digital-Twin-Arch-12*. Work in Progress. *IRTF Network Management Research Group*. 2026. – Available from: <https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/> – [Accessed: 14.05.2026].
7. Zhu, J., Lin, J., Tian, Z., He, F. & Zhao, L. “DeepNC: A fast GNN-based pre-verification surrogate for TSN configuration”. *arXiv*. 2026. DOI: <https://doi.org/10.48550/arXiv.2607.24398>.
8. Klemm, A., Lindemann, C. & Lohmann, M. “Modeling IP traffic using the batch Markovian arrival process”. *Performance Evaluation*. 2003; 54 (2): 149–173. DOI: [https://doi.org/10.1016/S0166-5316\(03\)00067-1](https://doi.org/10.1016/S0166-5316(03)00067-1).
9. Muscariello, L., Mellia, M., Meo, M., Ajmone Marsan, M. & Lo Cigno, R. “Markov models of Internet traffic and a new hierarchical MMPP model”. *Computer Communications*. 2005; 28 (16): 1835–1851. DOI: <https://doi.org/10.1016/j.comcom.2005.02.012>.
10. Schwefel, H.-P., Antonios, I. & Lipsky, L. “Understanding the relationship between network traffic correlation and queueing behavior: A review based on the N-Burst ON/OFF model”. *Performance Evaluation*. 2017; 115: 68–91. DOI: <https://doi.org/10.1016/j.peva.2017.07.002>.
11. Le Boudec, J.-Y. & Thiran, P. “Network Calculus: A Theory of Deterministic Queueing Systems for the Internet”. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer. 2001; 2050. DOI: <https://doi.org/10.1007/3-540-45318-0>.
12. Mohammadpour, E., Stai, E. & Le Boudec, J.-Y. “Improved network-calculus nodal delay-bounds in time-sensitive networks”. *IEEE/ACM Transactions on Networking*. 2023; 31 (6): 2902–2917. DOI: <https://doi.org/10.1109/TNET.2023.3275910>.
13. Suárez-Varela, J., Almasan, P., Ferriol-Galmés, M., Rusek, K., Geyer, F., Cheng, X., Shi, X., Xiao, S., Scarselli, F., Cabellos-Aparicio, A. & Barlet-Ros, P. “Graph neural networks for communication networks: Context, use cases and opportunities”. *IEEE Network*. 2023; 37 (3): 146–153. DOI: <https://doi.org/10.1109/MNET.123.2100773>.
14. Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P. & Cabellos-Aparicio, A. “RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN”. *IEEE Journal on Selected Areas in Communications*. 2020; 38 (10): 2260–2270. DOI: <https://doi.org/10.1109/JSAC.2020.3000405>.
15. Ferriol-Galmés, M., Paillissé, J., Suárez-Varela, J., Rusek, K., Xiao, S., Shi, X., Cheng, X., Barlet-Ros, P. & Cabellos-Aparicio, A. “RouteNet-Fermi: Network modeling with graph neural networks”. *IEEE/ACM Transactions on Networking*. 2023; 31 (6): 3080–3095. DOI: <https://doi.org/10.1109/TNET.2023.3269983>.
16. Ho, J., Jain, A. & Abbeel, P. “Denoising diffusion probabilistic models”. *Advances in Neural Information Processing Systems*. 2020; 33: 6840–6851. DOI: <https://doi.org/10.48550/arXiv.2006.11239>.
17. Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. & Poole, B. “Score-based generative modeling through stochastic differential equations”. *9th International Conference on Learning Representations (ICLR 2021)*. 2021. DOI: <https://doi.org/10.48550/arXiv.2011.13456>.
18. Leland, W. E., Taqqu, M. S., Willinger, W. & Wilson, D. V. “On the self-similar nature of Ethernet traffic (extended version)”. *IEEE/ACM Transactions on Networking*. 1994; 2 (1): 1–15. DOI: <https://doi.org/10.1109/90.282603>.
19. Sivaroopan, N., Bandara, D., Madarasingha, C., Jourjon, G., Jayasumana, A. P. & Thilakarathna, K. “NetDiffus: Network traffic generation by diffusion models through time-series imaging”. *Computer Networks*. 2024; 251: 110616. DOI: <https://doi.org/10.1016/j.comnet.2024.110616>.
20. Marcoccia, F., Adjih, C. & Mühlethaler, P. “NetDiff: Deep graph denoising diffusion for ad hoc network topology generation”. *arXiv*. 2024. DOI: [10.48550/arXiv.2410.08238](https://doi.org/10.48550/arXiv.2410.08238).

21. Du, H., Zhang, R., Liu, Y., Wang, J., Lin, Y., Li, Z., Niyato, D., Kang, J., Xiong, Z., Cui, S., Ai, B., Zhou, H. & Kim, D. I. “Enhancing deep reinforcement learning: A tutorial on generative diffusion models in network optimization”. *IEEE Communications Surveys & Tutorials*. 2024; 26 (4): 2611–2646. DOI: <https://doi.org/10.1109/COMST.2024.3400011>.

Отримано 03.09.2026

Отримано після перегляду 05.10.2026

Прийнято 12.10.2026

DOI: <https://doi.org/10.15276/ict.03.2026.74>

UDC 004.8:004.7

Analysis of computer network performance models for structural management tasks throughout the life cycle

Roman I. Naumenko ¹⁾

ORCID: <https://orcid.org/0009-0003-7612-1773>; 9530675@stud.op.edu.ua

Serhii A. Nesterenko ¹⁾

ORCID: <https://orcid.org/0000-0002-3757-6594>; san@op.edu.ua

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The article addresses the problem of selecting models for calculating computer network characteristics in tasks related to managing network structure throughout the life cycle. It is proposed to distinguish between the system life-cycle stage and its structural and configuration state, which makes it possible to separately describe the network development context and specific structural changes. The study follows a review-and-methodological approach; the literature base was formed through a structured thematic search and analysis of fundamental and recent studies on network performance modeling, analytical approaches, neural networks, graph-based models, and generative methods. Model selection is proposed to be performed with regard to the nature of the structural and configuration change, the availability and structure of data, the required type of estimate or guarantee, and the allowable computational budget. The capabilities of queueing theory models, Network Calculus, classical neural networks, graph neural networks, and diffusion models are analyzed. Their functional division into the roles of generator, evaluator, verifier, and optimizer is proposed, together with an operational procedure for mapping task parameters to an appropriate functional role and model class. It is shown that the suitability of a model for verifying established Service Level Objectives (SLOs) is determined not only by prediction accuracy but also by the correspondence between the type of model output and the form of the requirement being verified. An illustrative case of upgrading an overloaded network link demonstrates the sequence of candidate configuration evaluation, SLO verification, and selection of an admissible structural solution. In addition, a limited retrospective assessment of the proposed procedure was performed using published scenarios involving GNNs, Network Calculus, and diffusion models; in the considered cases, the functional roles identified by the procedure were consistent with the actual use of the models in the corresponding studies. The feasibility of combining analytical and data-driven models within a unified process for supporting structural decisions is substantiated. The proposed approach can be used to select and combine models when evaluating alternative states of a computer network throughout its life cycle.

Keywords: Computer networks; life cycle; structural and configuration state; SLO, queueing theory; Network Calculus; graph neural networks; diffusion models; network optimization

ІНФОРМАЦІЯ ПРО АВТОРІВ



Науменко Роман Іванович - аспірант кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь досліджень: методи та системи штучного інтелекту

Roman I. Naumenko - PhD student, Department of Computer Intelligent Systems and Networks. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0003-7612-1773>; 9530675@stud.op.edu.ua

Research field: artificial intelligence methods and systems



Нестеренко Сергій Анатолійович - доктор технічних наук, професор кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь дослідження: комп'ютерні мережі та інформаційні технології

Serhii A. Nesterenko - Doctor of Technical Sciences, Professor, Professor at the Department of Computer Intelligent Systems and Networks. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0000-0002-3757-6594>; san@op.edu.ua

Research field: computer networks and information technologies