



Разработка моделей для расчета задержек доступа к контенту с учетом кеширования в сетях NDN

С. И. Якименко^а, преподаватель, младший научный сотрудник, orcid.org/0000-0002-0698-1298, syakimenko@hse.ru

^аМосковский институт электроники и математики им. А. Н. Тихонова НИУ ВШЭ, Таллинская ул., 34, Москва, 123458, РФ

Введение: кеширование позволяет снизить задержку доступа к контенту с помощью его приближения к пользователям и разгрузки облачного уровня сети. Для этого применяются граничные серверы, однако стандартный стек TCP/IP не использует возможности кеширования полностью, поэтому необходимо рассмотреть модель кеша, который задействует более приспособленный для данной функции стек Named Data Networking. **Цель:** разработать модели для оценки задержек доступа к контенту с учетом особенностей кеширования на граничном узле в Named Data Networking. **Результаты:** предложены имитационная модель с реализацией алгоритма кеширования Least Recently Used на основе дискретно-событийного моделирования и математическая модель задержек получения контента. Модели дают возможность адекватно оценить задержку на каждом сегменте от пользователя до облака и в целом для системы, а также вычислить основные характеристики работы кеша: вероятность попадания в кеш, нагрузку на внешние и внутренние каналы. Кеширование на основе граничного сервера позволило снизить задержку доступа к контенту до 65 мс и уменьшить издержки оператора на обращения в удаленное облако до 60 %. **Практическая значимость:** результаты работы могут быть применены интернет-провайдерами и операторами связи для разгрузки магистральных каналов и повышения показателей качества обслуживания абонентов при скачивании популярного контента. **Обсуждение:** в исследовании рассмотрен сценарий с большим каталогом и пуассоновским потоком запросов на граничный кеш, что соответствует режиму, в котором вероятность агрегации запросов в таблице Pending Interest Table мала. Для изучения динамического эффекта агрегации на задержки доступа к контенту требуется более сложная модель трафика (например, ON-OFF или модель дробового шума), что является предметом дальнейших исследований.

Ключевые слова — кеширование, Least Recently Used, задержка, non-zero download delay, граничное кеширование, Named Data Networking, имитационное моделирование, облако.

Для цитирования: Якименко С. И. Разработка моделей для расчета задержек доступа к контенту с учетом кеширования в сетях NDN. *Информационно-управляющие системы*, 2026, № 2, с. 36–48. doi:10.31799/1684-8853-2026-2-36-48, EDN: TYQXCS

For citation: Iakimenko S. I. Development of models for calculating content access delays with account taken of caching in NDN networks. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 2, pp. 36–48 (In Russian). doi:10.31799/1684-8853-2026-2-36-48, EDN: TYQXCS

Введение

В последнее время создание пользователями Интернета все большего и большего объема «тяжелого» цифрового контента, а также повышение скорости передачи данных в беспроводных сетях вызывает чрезмерный спрос на ограниченные сетевые ресурсы. Высокая нагрузка на сетевую инфраструктуру и пиринговые стыки приводит к тому, что в сетях ухудшаются параметры качества обслуживания абонентов.

Одним из решений для разгрузки транзитных каналов является перенос вычислительных мощностей из ядра и удаленного облака на границу сети в рамках концепции периферийных вычислений (edge computing) [1]. В задачи, которые выполняются на Edge-уровне сети, могут входить фильтрация, предварительная обработка данных, а также их кеширование и вычисления в реальном времени.

Согласно Стратегии развития отрасли связи Российской Федерации (<http://static.government.ru/media/files/Pc7fHuejbNvqv17b0RJNv0RIqTo201UV.pdf>), доля выполняемых центрами обработки данных «граничных» вычислений к 2035 г. составит до 30 % общего объема вычислений в центрах обработки данных, а доля трафика между центрами обработки данных заметно возрастет. В общем случае архитектура гетерогенной сети с учетом Edge-уровня может включать несколько дополнительных уровней, между которыми с помощью проводных и беспроводных интерфейсов устанавливается множество горизонтальных и вертикальных связей. Например, в нее могут быть включены шлюзы доступа к сети для оконечных устройств, несколько звеньев Edge-серверов различной мощности, а также центры обработки данных для сбора перед облаком. Немало работ исследователей посвящены различным аспектам применения новых решений на границе сети [2], например обеспечению вы-

числений [3, 4], безопасности в автомобильных сетях [5].

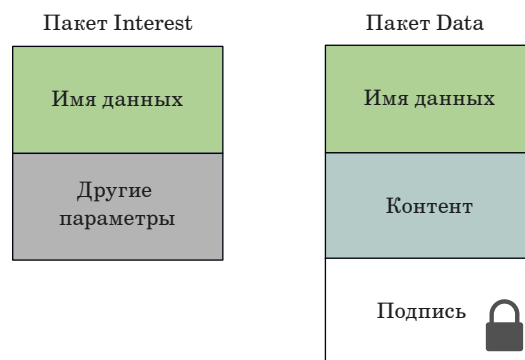
Перспективным решением для разгрузки удаленного облака при обработке запросов на ресурсы сети является граничное кеширование. Кеширование трафика контент-ориентированных социальных сетей (VK Video, Rutube, YAPPY и т. д.) может быть особенно полезным, поскольку существует высокая корреляция в контенте, который запрашивают пользователи. В итоге на замкнутой территории один и тот же популярный видеофайл могут просматривать несколько пользователей одновременно. При этом стандартное для стека TCP/IP прокси-кеширование в условиях мобильности пользователей и динамической беспроводной среды менее эффективно по сравнению со стационарным подключением [1].

Среди иностранных крупных исследователей, заявивших о концептуальной необходимости перехода от стека TCP/IP к сетям, ориентированным на информацию, или информационно-центричным сетям (ICN), можно выделить Дж. Куроуза (J. Kurose) [6], В. Якобсена (V. Jacobsen) [7] и Л. Чжанг (L. Zhang) [8]. На предпосылки разработки новых протоколов сетевого уровня в сетях поколения 5G/6G указывали такие известные российские ученые, как А. Е. Кучерявый, Р. В. Киричек, А. Ю. Гребешков [9, 10]. Работы авторов Г. Хасслингера (G. Hasslinger) [11], К. Фрикер (C. Fricker) [12] известны сильным вкладом в теоретические основы кеширования в ICN, а статьи М. Гаретто (M. Garetto) [13], М. Амадео (M. Amadeo) [2, 14] посвящены внедрению информационно-ориентированных концепций в телекоммуникационную инфраструктуру.

Сети именованных данных

Сети именованных данных (Named Data Networking, NDN) являются одной из новых альтернативных архитектур, упоминаемой в рекомендации RFC 8793 “Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology” (<https://datatracker.ietf.org/doc/rfc8793/>). Начало передачи данных в NDN инициируется принимающей стороной, т. е. потребителем данных (Data Consumer). В свою очередь производит данные отправляющая сторона, т. е. производитель данных (Data Producer). Пакет с запросом данных называется Interest. Пакет с данными, которые «запрашивает» Interest, называется Data (рис. 1).

Таблица Forwarding Information Base (FIB) является аналогом стандартной таблицы маршрутизации имен для NDN-сети. Таблица ожидания Pending Interest Table (PIT) указывает на список



■ **Рис. 1.** Структура пакетов Interest и Data в сети NDN

■ **Fig. 1.** Structure of Interest and Data packets in the NDN network

запросов Interest, которые еще не были удовлетворены, и интерфейсы, ожидающие на них ответ. Также на каждом NDN-узле существует структура Content Store (CS), которая и является кешем определенного размера. Кеширование пакетов на промежуточных и конечных устройствах является одним из преимуществ NDN-сети. Результативный запрос в кеш NDN, т. е. нахождение пакета с необходимым именем в кеше, называется попаданием кеша (cache hit). В случае, если нахождение пакета не произошло, говорят о промахе кеша (cache miss) [15].

В статьях [16–19] результаты для оценки влияния кеширования на работу сети были получены с помощью имитационного моделирования в симуляторах. Утверждалось [16], что NDN-сеть без кеша на узлах сама по себе не может удовлетворять требованиям по задержкам большинства современных приложений. Некоторые соображения по учету различного типа задержек приведены в статье [17] применительно к их влиянию на пропускную способность смоделированной сети NDN. В работе [18] был сделан общий вывод, что различные стратегии кеширования позволяют снизить задержку доставки объектов данных получателю и нагрузку на канал. К похожему результату пришли авторы статьи [19], где говорится, что применение механизмов NDN в mesh-сети помогает сохранять задержку на уровне не более 50 мс, причем сокращение становится более значительным для контента, популярность которого ранжирована по закону Ципфа.

Модели кеширования разделяются на категории ZDD (Zero-Download-Delay) и Non-ZDD (Non-Zero-Download-Delay). В модели ZDD для упрощения предполагается, что если нужный элемент контента (ЭК) отсутствует в кеше, он мгновенно загружается с удаленного ресурса. Второй случай более реалистичен и предполагает, что с момента запроса на ЭК до получения пользовате-

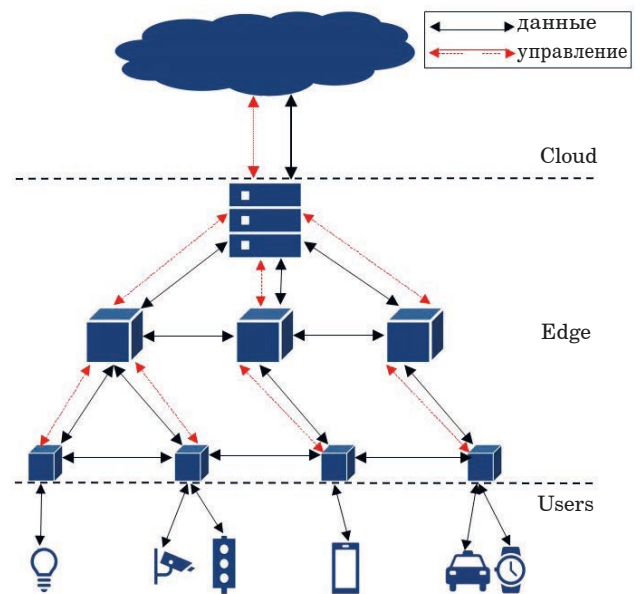
лем данного ЭК происходит некоторая задержка. Важную роль в данном процессе играет PIT: если в течение времени загрузки контента на него приходит еще несколько запросов, то все они не отправляются на уровень вверх, а накапливаются в данной таблице. После поступления пакета Data на узел благодаря накоплению запросов в PIT пакет рассылается всем устройствам, которые ранее его запросили. Задержка загрузки, таким образом, является параметром, влияющим на эффективность кеширования.

При этом в научной литературе зачастую задержки задаются значениями-константами, которые берутся из измерений на сети или даются без обоснования. Общим недостатком в таком случае является разрозненность значений, которые выбираются авторами. Так, например, в работе [20] рассматриваются три сценария с выбором все более удаленного облака. Задержка до уровня Edge во всех случаях равна 1 мс, а задержка до облака Cloud последовательно увеличивается от 20–30 до 80 мс, причем задается реальными измерениями на сети. В статье [21] задаются задержки между базовой станцией и Edge-сервером (0,5 мс), а также Edge-сервером и облаком (2 мс), без учета пользовательской задержки. Похожий подход предложен и в [22], где берутся во внимание только задержки между Edge-уровнем и Cloud (от 10 до 100 мс). Константы для задержек (в микросекундах) на каждом из сегментов 5G-сети с поддержкой Edge-уровня предлагаются в статье [23]. Методика по обоснованию значений задержки для обработки видеопотока реального времени на Edge-сервере приведена в статье [24], которую можно считать самой близкой к текущей работе. Однако статья [24] посвящена TCP/IP-сети и не учитывает влияние кеширования внутри сети на результаты, что в данной работе учтено.

Таким образом, существует необходимость в развитии математического аппарата для моделей работы сети в альтернативных TCP/IP-архитектурах для обоснования выигрыша для задержки от кеширования. Актуальной становится задача разработки математической и имитационной модели для обоснования расчета задержек, которая учитывает особенности кеширования на базе NDN.

Системная и математическая модели кеширования с учетом условия Non-ZDD

Рассмотрим процесс получения данных из кеша, работающего через протокол NDN, с учетом задержки (рис. 2). Уровень Edge для упрощения представлен единственным сервером. На первом этапе пакет-запрос Interest отправляется на уровень Edge с размером кеша C . Если на данном



■ **Рис. 2.** Архитектура гетерогенной сети с учетом Edge-уровня
 ■ **Fig. 2.** Architecture of a heterogeneous network considering the Edge layer

уровне происходит попадание кеша, то в сторону оконечного устройства (ОУ; User Equipment, UE) обратно продвигается пакет Data. Если происходит промах кеша на Edge, запрос отправляется на уровень Cloud, где он удовлетворяется в любом случае. Согласно стандартам NDN, данный пакет должен пройти по тем же промежуточным узлам, по которым двигался и пакет Interest.

Предположим, что узел Cloud содержит все элементы контента из каталога размером M . Предположим также, что каждый оконечный узел соединен с узлом Cloud через линейную топологию, подразумевающую отсутствие горизонтальных связей между узлами Edge одного уровня.

Для разработки модели примем следующие допущения.

1. На вход кеша поступает пуассоновский поток запросов с общей интенсивностью Λ , при этом интенсивность запроса элемента $i \in [1, M]$, $i \in \mathbb{N}$, пропорциональна популярности:

$$\lambda_i = \Lambda p_i. \quad (1)$$

2. Популярность p_i запросов одинакова в любой момент времени симуляции и в любой географической точке, а также не зависит от предыдущих запросов. Для упрощения размеры всех элементов контента равны [5] (условие Independent Reference Model), что является общепринятым допущением для исследований алгоритмов кеширования [11].

3. Популярность p_i запросов на элементы контента распределена по закону Ципфа с коэффициентом асимметрии $\alpha = 0,8$. Распределение Ципфа точно характеризует популярность запросов на пользовательский контент, при этом значение коэффициента α для разных сценариев колеблется в пределах от 0,65 до 1,2, а $\alpha = 0,8$ является одним из чаще всего используемых в литературе [12]:

$$p_i = \frac{i^{-\alpha}}{\sum_{k=1}^M k^{-\alpha}}. \quad (2)$$

4. Политика вытеснения кеша – Least Recently Used (LRU), политика же репликации контента – Leave Copy Everywhere (LCE) [25], т. е. при обращении на уровень Cloud ЭК не только отправляется в сторону ОУ, но и реплицируется на уровне Edge для следующих запросов с вероятностью 100 %.

5. Как показано в работе [26], при пуассоновском потоке заявок значимое влияние РИТ на сокращение восходящего трафика наблюдается при относительно небольшом размере каталога контента или высокой интенсивности запросов на отдельный объект. В рассматриваемой модели РИТ практически не ограничивает поток запросов к кешу и облаку.

При данных предположениях для оценки вероятности попадания в кеш LRU $p_{hit}(i)$ применяется аппроксимация Че [13]. Согласно ей, для каждого элемента i в каталоге выполняется приближенное равенство

$$p_{hit}(i) \approx 1 - e^{-\lambda_i T_C}, \quad (3)$$

где T_C – среднее время жизни («характеристическое время» Че) элементов в кеше размера C , $C \leq M$. При этом время T_C предполагается одинаковым для всех элементов.

Однако с учетом условия Non-ZDD формула для вероятности попадания требует модификации. В работе [27] предложена следующая корректировка, учитывающая среднее время скачивания объекта $M(D_{Down})$:

$$p_{hit}(i) \approx \frac{e^{\lambda_i T_C} - 1}{\lambda_i M(D_{Down}) + e^{\lambda_i T_C}}. \quad (4)$$

Поскольку «характеристическое время» входит в выражения (3), (4) для вычисления $p_{hit}(i)$ нелинейно, это значение можно вычислить численными методами. Например, для условия ZDD имеем уравнение относительно T_C

$$\sum_{i=1}^M p_{hit}(i) = \sum_{i=1}^M \left(1 - e^{-\lambda_i T_C}\right) = C. \quad (5)$$

Для получения общего значения p_{hit} по кешу используется средневзвешенная сумма

$$p_{hit} = \sum_{i=1}^M P_i p_{hit}(i). \quad (6)$$

Общая задержка D_{Down} – это время, которое проходит с момента отправки запроса Interest до получения пакета информации Data. Оно может быть разделено на четыре части:

1) время запроса (D_1) – время с момента генерации запроса Interest ОУ до получения данного запроса узлом Edge;

2) время отправки пакета Data на ОУ (D_2);

3) время запроса Interest на уровень вверх (D_3) – опциональное время, необходимое для пересылки запроса Interest на Cloud, если он не удовлетворен в кеше Edge;

4) время отправки пакета Data на уровень вниз (D_4) – опциональное; Data-пакет движется по обратному пути.

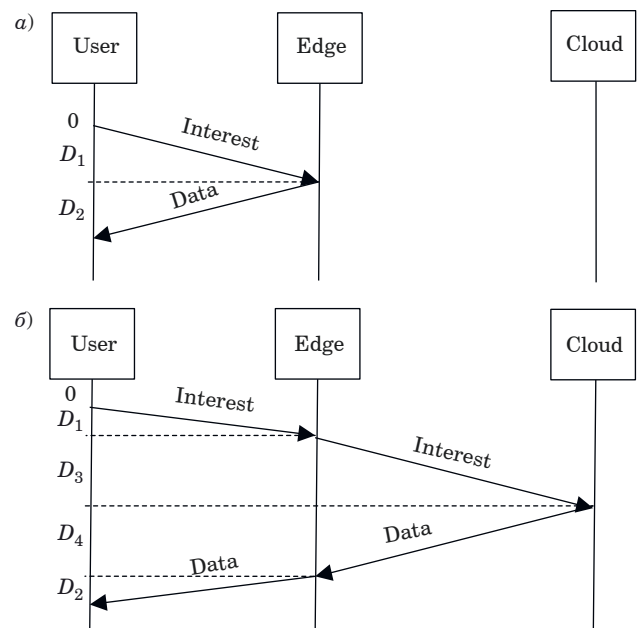
Временная диаграмма, отражающая попадание в кеш, представлена на рис. 3, а, временная диаграмма для промаха кеша приведена на рис. 3, б.

Таким образом, в случае попадания в кеш имеем соотношение

$$D_{Down_hit} = D_1 + D_2, \quad (7)$$

в случае же промаха кеша имеем

$$D_{Down_miss} = D_1 + D_2 + D_3 + D_4. \quad (8)$$



■ Рис. 3. Временная диаграмма работы кеша: а – попадание; б – промах

■ Fig. 3. Cache operation timing diagram: a – hit; б – miss

Примем задержку на каждом этапе случайной величиной, распределенной экспоненциально со средними значениями μ_1, μ_2, μ_3 и μ_4 (где μ_i – интенсивность обслуживания на i -м этапе соответственно в секундах в минус первой степени). Экспоненциальное распределение адекватно моделирует время обслуживания в системах, где возможны случайные задержки из-за ошибок передачи, повторных запросов или флуктуации нагрузки, что типично для сетевых каналов [28]. Распределение суммы независимых экспоненциальных случайных величин подчиняется распределению Кокса. Этот класс распределений является плотным в множестве всех распределений на неотрицательной полуоси [29]. Это означает, что, варьируя параметры модели (число этапов, интенсивность), можно аппроксимировать произвольное распределение задержек, что сохраняет общность подхода. Распределение Кокса также позволяет оценивать общую задержку через ее математическое ожидание, которое является устойчивой и интерпретируемой характеристикой для сравнения сценариев.

Общее время обработки пакета, таким образом, можно описать с помощью фазовой модели, представленной в виде диаграммы Кокса (рис. 4).

Характеристическую функцию (эквивалентную преобразованию Лапласа – Стилтгеса) времени задержки для данной фазовой модели можно получить, анализируя граф переходов [29]:

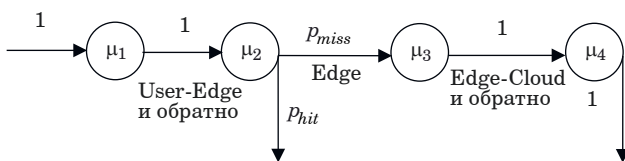
$$F(s) = p_{hit} \left(\frac{\mu_1}{\mu_1 + s} \right) \left(\frac{\mu_2}{\mu_2 + s} \right) + p_{miss} \left(\frac{\mu_1}{\mu_1 + s} \right) \left(\frac{\mu_2}{\mu_2 + s} \right) \left(\frac{\mu_3}{\mu_3 + s} \right) \left(\frac{\mu_4}{\mu_4 + s} \right), \quad (9)$$

где s – комплексный параметр преобразования Лапласа – Стилтгеса ($s \in C$).

Математическое ожидание случайной величины задержки находится по формуле [29]

$$M(D_{Down}) = \sum_{i=1}^4 \frac{q_i}{\mu_i} = \frac{1}{\mu_1} + \frac{1}{\mu_2} + \frac{p_{miss}}{\mu_3} + \frac{p_{miss}}{\mu_4}, \quad (10)$$

где q_i – вероятность того, что пакет пройдет i -ю фазу на своем пути. Для данной диаграммы $q_1 = 1, q_2 = 1, q_3 = p_{miss}, q_4 = p_{miss}$.



■ **Рис. 4.** Четырехфазная диаграмма Кокса, моделирующая путь пакета

■ **Fig. 4.** 4-phase Cox diagram modeling the packet path

Дисперсия случайной величины задержки [29]

$$D(D_{Down}) = 2 \sum_{i=1}^4 \left(\left(\sum_{j=1}^i \frac{1}{\mu_j} \right) \frac{q_i}{\mu_i} \right) - M^2(D_{Down}). \quad (11)$$

Окончательно получим

$$D(D_{Down}) = 2 \left(\begin{aligned} & \frac{1}{\mu_1^2} + \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \right) \frac{1}{\mu_2} + \\ & \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} + \frac{1}{\mu_3} \right) \frac{p_{miss}}{\mu_3} + \\ & \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} + \frac{1}{\mu_3} + \frac{1}{\mu_4} \right) \frac{p_{miss}}{\mu_4} \end{aligned} \right) - M^2(D_{Down}). \quad (12)$$

Полученные аналитические выражения (10), (12) задают первые два момента распределения времени доставки контента. Сравнение теоретических значений с результатами симуляции представлено далее.

Реализация модели

Для нахождения величин $D_1 = \mu_1^{-1}, D_2 = \mu_2^{-1}, D_3 = \mu_3^{-1}$ и $D_4 = \mu_4^{-1}$ (в секундах) примем, что на каждом этапе $D_n, n \in [1, 4]$ задержка передачи пакета в общем случае состоит из следующих компонентов:

- 1) задержки доступа пакета к каналу D_{ac} – времени между его генерацией и концом текущего кадра (опциональное, относится только к первому этапу);
- 2) задержки сериализации D_{ser} , или постановки пакета в очередь на передачу;
- 3) задержки ожидания в очереди $D_{queuing}$;
- 4) задержки передачи пакета D_{tr} :

$$D_n = D_{ac} + D_{ser} + D_{queuing} + D_{tr}. \quad (13)$$

Для вычисления задержки генерации пакета рассмотрим систему TDMA (Time Division Multiple Access), состоящую из K пользователей, каждый из которых передает пакеты одинаковой длины по S бит каждый. Если общая скорость передачи составляет R бит/с, то время сериализации пакета $D_{ser} = S/R$, что примем за размер слота. Длительность всего цикла, таким образом, составляет $D_{II} = KS/R$. Предполагаем, что процессы прибытия новых пакетов у разных пользователей независимы, тогда очереди пакетов от каждого пользователя не влияют на очереди других пользователей. Пользователь передает пакет, если он у него есть, каждые D_{II} секунд, независимо от любого события в любой из других

очереди других пользователей. Следовательно, в дальнейшем мы сосредоточимся на характеристиках одного пользователя и, не умаляя общности, предположим, что пользователь передает пакет, если он у него есть, в первом слоте каждого кадра.

Поскольку все кадры имеют одинаковую длину, среднее время между временем генерации пакета и концом текущего кадра составляет $D_{ac} = 0,5D_{ц} = KS/2R$.

Будем считать, что линии связи обладают большим запасом пропускной способности, поэтому задержкой обработки в очереди $D_{queuing}$ пренебрежем.

Пусть L — физическое расстояние между узлами, v — скорость распространения бит по линии связи. При этом в оптоволокне, ведущем на уровень Cloud, имеем $v = nc$, где n — показатель преломления света.

Пусть также длина одного участка оптоволокна $L_{hop} = 100$ км. При передаче пакета каждые $n_{hop} = \lfloor L/L_{hop} \rfloor$ участков пакет также испытывает дополнительную задержку, связанную с оптическим-электрооптическим (О/Е/О) преобразованием сигнала и обработкой в таблицах маршрутизации на промежуточном роутере D_{proc} .

Итого для задержки передачи пакета имеем

$$D_{tr} = \frac{L}{v} + n_{hop}D_{proc}. \quad (14)$$

На участке сети между граничным маршрутизатором и конечным пользователем (часто называемом «последней милей») $L_{edge} \ll L$, поэтому для запроса, ведущего к уровню Edge и обратно, данной задержкой пренебрежем. Тогда формулу (7) можно переписать так:

$$D_{Down_hit} = D_{ac} + 2D_{ser}. \quad (15)$$

В свою очередь формула (8) принимает вид

$$D_{Down_miss} = D_{ac} + 4D_{ser} + 2D_{tr}. \quad (16)$$

Средние значения времени обслуживания на каждом этапе соответственно равны:

$$\mu_1^{-1} = \frac{KS}{2R} + \frac{S}{R} = \frac{KS + 2S}{2R}; \quad (17)$$

$$\mu_2^{-1} = \frac{S}{R}; \quad (18)$$

$$\mu_3^{-1} = \mu_4^{-1} = \frac{S}{R} + \frac{L}{nc} + n_{hop}D_{proc}. \quad (19)$$

Таким образом, предложенная параметризация (17)–(19) устанавливает связь между харак-

теристиками сети и параметрами распределения Кокса μ_i , что позволит в дальнейшем исследовать влияние физических параметров сети на результирующую задержку доставки контента.

Имитационное моделирование

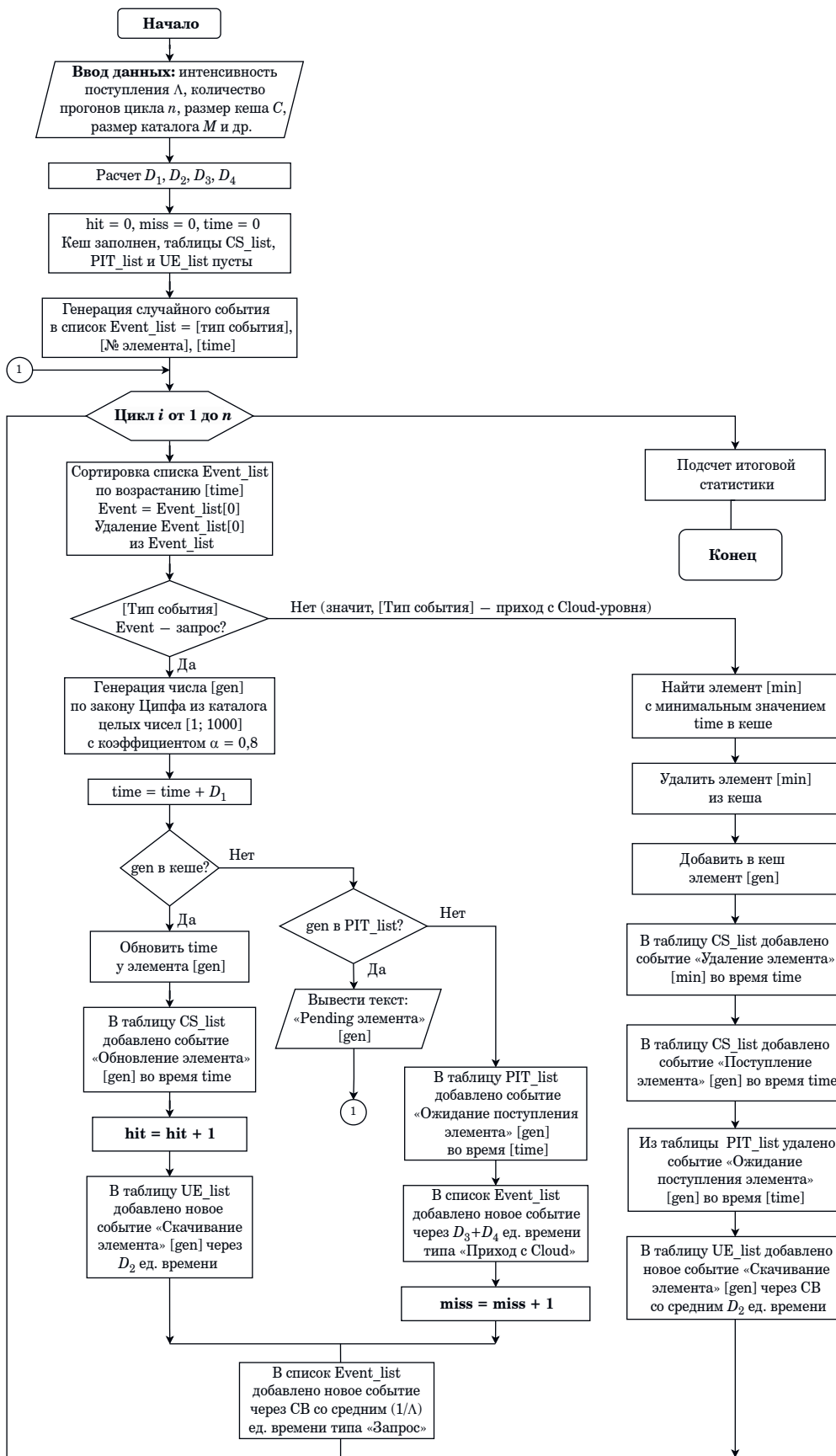
Автором разработан алгоритм, учитывающий особенности как кеша LRU, так и всех компонентов узла NDN (рис. 5). Для имитационного моделирования работы политики LRU использован подход дискретно-событийного моделирования с переменным шагом продвижения модельного времени.

На каждом прогоне к переменной продвижения модельного времени *time* добавляется случайная величина, распределенная по экспоненциальному закону со средним $1/\Lambda$ единиц. Список *Event_list* определяет ближайшее по модельному времени событие, которое будет обрабатываться алгоритмом. Перед основным циклом кеш предварительно заполнен недублирующимися элементами контента для установления стационарного режима [11]; в блок-схеме для экономии места этот процесс не описан.

Имитационная модель разработана на языке Python 3.11 в среде Jupyter Notebook. Численные расчеты и генерация случайных величин выполнены с помощью библиотек NumPy и SciPy. Все эксперименты проводились на рабочей станции с ОС Ubuntu 20.04 LTS, процессором Intel Core i5 и 16 ГБ ОЗУ. Код модели доступен по запросу у автора.

Состояния кеша реализованы с помощью класса OrderedDict модуля Collections в языке Python — для упорядочиваемых словарей. В словарях хранятся элементы вида {ключ: значение}. Ключом в словаре является число *gen*, значением — время *time*. Таким образом, пара {*gen*; *time*} обозначает состояние кеша, при котором в нем хранится элемент ранга *gen*, последний запрос к которому произошел во время *time*. Если в кеш попадает ЭК с таким же ключом, его время будет обновлено. Если же такого ключа в кеше нет, то будет вытеснен элемент с наименьшим значением *time*, а на его место попадет новый ЭК.

На вход кеша поступает поток заявок по закону Пуассона с интенсивностью Λ заявок/с. Заявки промаркированы с помощью числа *gen*. Данное число сгенерировано по закону Ципфа из каталога целых чисел 1, 2, ..., M с коэффициентом асимметрии $\alpha = 0,8$ и обозначает ранг элемента в каталоге M . Входные параметры, упоминаемые в блок-схеме: размер кеша $C = 100$, размер каталога $M = 500$, количество прогонов основного цикла $n = 2000$. Остальные численные параметры указаны в таблице.



■ **Рис. 5.** Алгоритм основного цикла для кеша NDN по правилу LRU: CB – случайная величина
 ■ **Fig. 5.** The main loop algorithm for the NDN cache using the LRU rule: CB is a random variable

- Начальные численные параметры для вычисления задержек
- Initial numerical parameters for calculating delays

Параметр	Значение
Скорость передачи данных R на сегменте до пользователя, Мбит/с	100
Размер пакета Interest $S_{Interest}$, байт	50
Размер пакета Data S_{Data} , байт	1500
Пользователи K на уровне Users	5
Скорость света c в вакууме, м/с	$3 \cdot 10^8$
Показатель преломления света n в оптоволокне	1,46
Общее расстояние L до уровня Cloud, км	2000
Длина сегмента оптоволокна L_{hop} , км	100
Дополнительная задержка на сегменте D_{proc} , мс	1,5
Общая интенсивность запросов на кеш Λ , заявок/с	15

В блоке «Подсчет итоговой статистики» рассчитываются следующие параметры:

- время «жизни» пакета i в кеше, T_i , его среднее «характеристическое значение» $T_C = M(T_i)$ и СКО $\sigma(T_i)$;

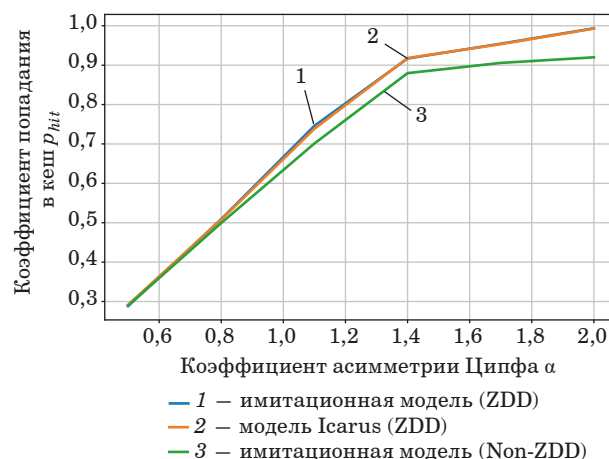
- коэффициент попадания в кеш $hit/(hit + miss)$ по результатам моделирования и по формуле (4) с учетом Non-ZDD;

- размер кеша по формуле Че по формуле (5);
- средняя задержка скачивания пакета Data, ее дисперсия и математическое ожидание с помощью функций $variance()$ и $stdev()$ модуля Statistics. В симуляторе также реализованы списки cs_list , pit_list , ue_list , хранящие основные события для сбора статистики в структурах Content Store и PIT на узле Edge, а также OУ.

Валидация модели

Валидация модели

Для проверки корректности реализации имитационной модели была выполнена серия тестовых экспериментов. Так, проведено сравнение с симулятором кэширования Icarus (<https://icarus-sim.github.io/>) по коэффициенту попадания в кеш LRU. В Icarus не поддерживается моделирование условия Non-ZDD, а параметры задаются через конфигурацию эксперимента (модули *workload* и *topology*). Для модели в Icarus были реализованы идентичные условия из таблицы (топология, размер каталога и кеша, нагрузка), за переменный параметр был взят коэффициент $\alpha \in [0,5; 2]$. При отключенном условии Non-ZDD ($D_3 = D_4 = 0$) вероятность попадания в кеш имитационной мо-



■ Рис. 6. Вероятность попадания в кеш p_{hit} как функция коэффициента Ципфа α

■ Fig. 6. Cache hit probability p_{hit} as a function of the Zipf coefficient α

дели количественно согласуется с результатами в Icarus (рис. 6, кривые 1 и 2). Разница не более 3 % позволяет утверждать, что алгоритм LRU в имитационной модели реализован корректно, а дальнейшие эксперименты с учетом условия Non-ZDD отражают именно влияние задержки скачивания, а не побочные эффекты.

Дополнительно на рис. 6 приведена кривая 3 для модели с включенным условием длительной загрузки ($M(D_{down}) = 0,1$ с). При активации условия Non-ZDD наблюдается снижение вероятности попадания в кеш. Этот эффект усиливается с ростом α : при слабом перекосе ($\alpha < 1$) снижение составляет около 1–2 %, при сильном перекосе ($\alpha = 2$) достигает 8 %. В условиях такого перекоса популярности основная доля запросов приходится на небольшую группу ЭК, которые при большой задержке доступа систематически не успевают попадать в кеш после вытеснения.

Результаты моделирования

При первоначальных параметрах получим результаты, согласующиеся с реальными измерениями на Edge-Cloud-сети [20]: $\mu_1^{-1} = 14$ мс; $\mu_2^{-1} = 4$ мс; $\mu_3^{-1} = \mu_4^{-1} = 38,6$ мс; математическое ожидание задержки $M(D_{Down}) \approx 56,86$ мс; дисперсия задержки $D(D_{Down}) \approx 3,2$ мс; средняя задержка при попадании в кеш $D_{Down_hit} = 18$ мс; при промахе кеша $D_{Down_miss} = 95$ мс. Таким образом, основная часть задержки получения контента формируется при обращении в удаленное облако.

Начнем эксперименты со сменой параметров. При $\alpha = 1,1$ будем увеличивать размер кеша от 25 до 225 элементов с шагом 25 при одинако-

вом размере каталога. В данном эксперименте проверим точность формул дисперсии (12) и математического ожидания (10) задержек в математической модели в сравнении с разработанным симулятором (рис. 7).

Как видим, увеличение размера кеша на уровне Edge постепенно уменьшает среднюю задержку скачивания контента. Получены результаты уменьшения задержки порядка десятков миллисекунд (от 57 до 27 мс).

При вычислении средней абсолютной ошибки (mean absolute error, MAE) и средней относительной ошибки (mean relation error, MRE) соответствующих задержек в математической и имитационной моделях получены значения:

$$MAE(M(D_{Down})) = 0,000709;$$

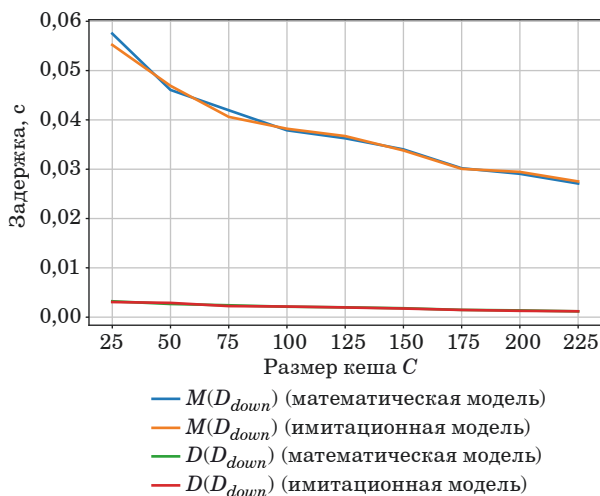
$$MAE(D(D_{Down})) = 0,000087.$$

$$MRE(M(D_{Down})) = 1,69 \%; MRE(D(D_{Down})) = 3,9 \%.$$

Таким образом, сравнение математической и имитационной моделей показывает высокую степень их адекватности.

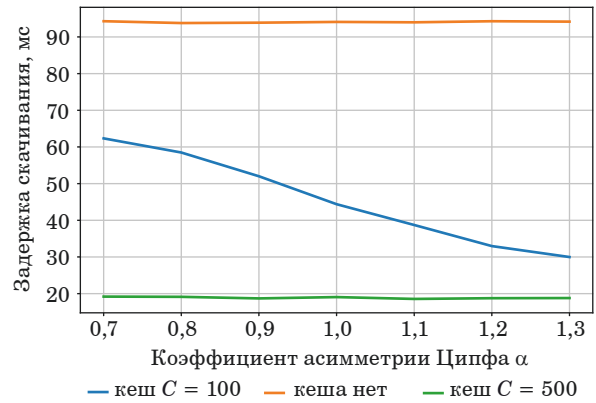
В следующем эксперименте (рис. 8) сначала удалим кеш с уровня Edge, чтобы проверить эффективность уменьшения задержек (все запросы идут на уровень Cloud). Потом добавим кеш с тем же размером, что и каталог M (500 элементов), чтобы все запросы удовлетворялись на уровне Edge. Эксперимент проведен с коэффициентом асимметрии α от 0,7 до 1,3 с шагом 0,1.

Как известно, увеличение α приводит к возрастанию коэффициента попадания в кеш LRU, из-за чего средняя задержка уменьшается. При $\alpha > 1$ небольшая часть самых популярных объектов получает непропорционально большую



■ Рис. 7. Сравнение имитационной и математической моделей

■ Fig. 7. Comparison of simulation and mathematical models



■ Рис. 8. Задержка скачивания элементов контента из системы с кешем LRU

■ Fig. 8. Delay in downloading content elements from a system with an LRU cache

долю запросов и находится в кеше длительное время. Выигрыш (уменьшение задержки) при использовании кеша NDN здесь составляет от 32 мс при $\alpha = 0,8$ до 65 мс при $\alpha = 1,3$.

Рассмотрим эксперимент по увеличению количества пользователей на уровне Users в системе. Пусть все пользователи отправляют запрос на скачивание файла размером S_{Data} и получают его по сети NDN. Тогда пропускная способность системы $throughput$ может быть определена как количество байт, которое проходит через нее за единицу времени в целях запроса и получения данных:

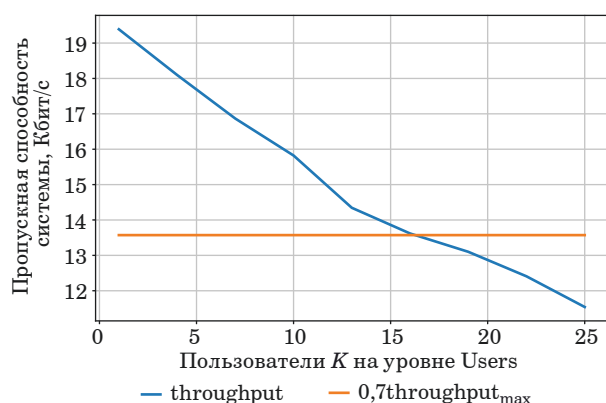
$$throughput = \frac{S_{Interest} N_{Interest} + S_{Data} N_{Data}}{T}, \quad (20)$$

где T – время проведения имитационного моделирования.

Пусть максимальная пропускная способность системы $throughput_{max}$ достигается без конкуренции, при $K = 1$ пользователь, и пусть система стабильно работает при значениях $throughput > 0,7 throughput_{max}$. Рассчитаем данный показатель при увеличении количества пользователей K до 25 (рис. 9).

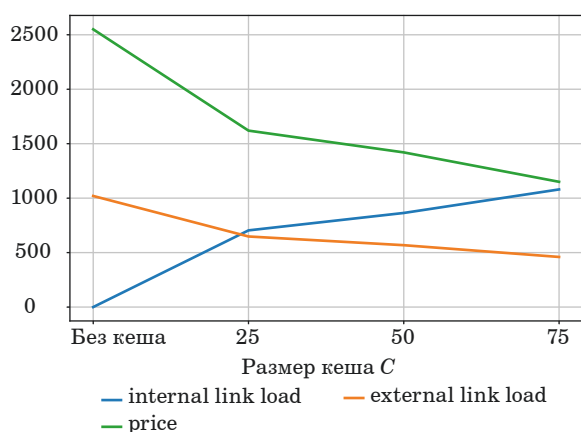
Таким образом, видим, что конкуренция пользователей за доступ к каналу выводит систему из стабильного состояния при $K > 16$ пользователей. При этом задержка доступа к каналу в слоте растет пропорционально K и достигает более $D_{ac} = K S_{Interest} / 2R = 32$ мс, что также увеличивает среднюю задержку D_{Down} практически до 100 мс, по этой причине добавлять пользователей в систему также нет смысла.

Приведем эксперимент в так называемом режиме малого кеша (по отношению к размеру каталога), увеличивая C от состояния «без кеша» при $C = 0$ до $C = 75$. Назовем показателем «нагрузка внутренних каналов» (*internal link load*)



■ **Рис. 9.** Исследование пропускной способности кеш-системы

■ **Fig. 9.** Research of cache system throughput



■ **Рис. 10.** Вычисление показателей internal link load, external link load и price

■ **Fig. 10.** Calculation of internal link load, external link load and price indicators

количество обращений к каналам до второго хопа, т. е. с уровня Users до уровня Edge и обратно [15]. Соответственно, показатель «нагрузка внешних каналов» (*external link load*) — это количество обращений к каналам, начиная со второго хопа, т. е. с уровня Edge на уровень Cloud и обратно. Пусть за обращение в облако провайдер платит цену $B = 2,5$ у.е. Стоимость «промахов кеша» вследствие его неоптимальной работы будет соответственно равна $Price = B \cdot external\ link\ load$ (рис. 10).

Таким образом, добавление даже небольшого кеша уменьшает показатель *external link load* и,

соответственно, издержки провайдеров (практически на 1500 у.е. по сравнению с издержками при отсутствии кеша, что обозначает уменьшение издержек до 60 %).

Заключение

В данной работе изучено влияние кеширования на задержку получения контента в сетях NDN с учетом условия Non-ZDD. В отличие от существующих работ, на основе разработанных моделей подтверждено, что в сетях NDN выигрыш от кеширования по снижению задержки является количественно ограниченным и может быть предсказан. Кроме того, предложен метод, позволяющий количественно оценить издержки провайдера, администрирующего кеш.

Для различных сценариев задержка уменьшалась от 20 до 65 мс в зависимости от размера кеша и распределения популярности элементов. Математическая фазовая модель времени обслуживания на разных сегментах системы, представленная в виде диаграммы Кокса, позволила рассчитать математическое ожидание и дисперсию задержек для случаев попадания и промаха кеша. Значения задержек (18 мс при попадании в кеш и 95 мс при промахе кеша) при таком подходе согласуются с работами, где производились замеры на реальной сети с Edge- и Cloud-уровнем, что является подтверждением того, что предложенная модель адекватно отражает соотношение задержек в реальных сценариях.

В будущем задают направление для дальнейших исследований вопросы преодоления ограничений модели, расширения среды имитационного моделирования для исследования и анализа более сложных алгоритмов вытеснения кеша, например k-LRU [13] или S3-FIFO. Предстоящая работа может быть направлена также на переход к сегментно-ориентированной модели, где для полного скачивания файла необходимо получить несколько пакетов Data.

Финансовая поддержка

Исследование выполнено за счет гранта Российского научного фонда № 24-19-00299, <https://rscf.ru/project/24-19-00299/>

Литература

1. Якименко С. И., Кучерявый Е. А. Кооперативное кеширование контента в NDN-сценариях для сетей VANET. *Электросвязь*, 2023, № 9, с. 9–17. doi:10.34832/ELSV.2023.46.9.002, EDN: HUNFMD

2. Amadeo M., Ruggeri G. Exploring in-network computing with information-centric networking: Review and research opportunities. *Future Internet*, 2025, vol. 17 (1), no. 42, pp. 1–28. doi:10.3390/fi17010042

3. Kovalchukov R., Glazkov R., Srikanteswara S., Zhang Y., Moltchanov D., Arrobo G., Feng H.,

- Spoczynski M., Himayat N. In-network dynamic compute orchestration over mobile edge systems. *Proceedings of the 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–7. doi:10.1109/VTC2023-Spring57618.2023.10200992
4. Pirmagomedov R., Srikanteswara S., Moltchanov D., Arrobo G., Zhang Y., Himayat N., Koucheryavy Y. Augmented computing at the edge using named data networking. *Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6. doi:10.1109/GCWkshps50303.2020.9367585
 5. Vladyko A., Tambovtsev G., Podgornaya E., Cheloung S. A., Alkanhel R., Plotnikov P. Cluster-based vehicle-to-everything model with a shared cache. *Mathematics*, 2023, vol. 11(13), no. 3017, pp. 1–19. doi:10.3390/math11133017
 6. Kurose J. Information-centric networking: The evolution from circuits to packets to content. *Computer Networks*, 2014, vol. 66, pp. 112–120. doi:10.1016/j.comnet.2014.04.002
 7. Jacobson V., Smetters D. K., Thornton J. D., Plass M. F., Briggs N. H., Braynard R. L. Networking named content. *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, 2009, pp. 1–12. doi:10.1145/1658939.1658941
 8. Zhang L., Afanasyev A., Burke J., Jacobson V., Claffy K. C., Crowley P., Papadopoulos C., Wang L., Zhang B. Named data networking. *ACM SIGCOMM Computer Communication Review*, 2014, vol. 44, no. 3, pp. 66–73. doi:10.1145/2656877.2656887
 9. Бородин А. С., Рожков М. А., Киричек Р. В., Кучерявый А. Е. New IP от 5G к 6G: нужна ли смена парадигмы? *Электросвязь*, 2020, № 7, с. 15–21. doi:10.34832/ELSV.2020.8.7.002, EDN: CSEHIN
 10. Гребешков А. Ю., Боровская Я. А. Построение информационно-ориентированных сетей 5G-ICN. *Вестник связи*, 2021, № 11, с. 13–18. EDN: IDYVMD
 11. Hasslinger G., Ntougias K., Hasslinger F., Hohlfeld O. Analysis of the LRU cache startup phase and convergence time and error bounds on approximations by Fagin and Che. *IEEE 2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2022, pp. 254–261. doi:10.23919/wiopt56218.2022.9930556
 12. Fricker C., Robert P., Roberts J., Sbihi N. Impact of traffic mix on caching performance in a content-centric network. *Proceedings of the 2012 IEEE INFOCOM Workshops*, 2012, pp. 310–315. doi:10.1109/INFOCOMW.2012.6193511
 13. Garetto M., Leonardi E., Martina V. A unified approach to the performance analysis of caching systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2016, vol. 1, no. 3, pp. 1–28. doi:10.1145/2896380
 14. Amadeo M., Campolo C., Ruggeri G., Molinaro A. Beyond edge caching: Freshness and popularity aware IoT data caching via NDN at internet-scale. *IEEE Transactions on Green Communications and Networking*, 2021, vol. 6, no. 1, pp. 352–364. doi:10.1109/TGCN.2021.3124452
 15. Якименко С. Аспекты кэширования в информационно-ориентированных сетях. *Современные сетевые технологии (MONETEC-2022): тр. 4-й Междунар. конф.*, Москва, 27–29 октября 2022 г., М., с. 70–76. EDN: NUPDPT
 16. Блинников М. А., Пирмагомедов Р. Я., Молчанов Д. А., Кучерявый Е. А. Применение технологий именованных данных в беспроводных ячеистых сетях. *Электросвязь*, 2019, № 11, с. 22–28. EDN: RHQFJM
 17. Wibowo T. A., Syambas N. R., Hendrawan L. V., Yovita R. A. Closer towards named data networking implementation. *International Journal of Intelligent Engineering and Systems*, 2023, vol. 16, no. 1, pp. 265–276. doi:10.22266/ijies2023.0228.24
 18. Путевская И. В. Исследование эффективности стратегий кэширования данных в информационно-ориентированных сетях. *Вестник Чувашского университета*, 2024, № 4, с. 128–140. doi:10.47026/1810-1909-2024-4-128-140, EDN: SXLOVC
 19. Glazkov R., Moltchanov D., Srikanteswara S., Samuylov A., Arrobo G., Zhang Y., Feng H., Himayat N., Spoczynski M., Koucheryavy Y. Provisioning of fog computing over named-data networking in dynamic wireless mesh systems. *Sensors*, 2024, vol. 24 (4), no. 1120, pp. 1–24. doi:10.3390/s24041120
 20. Ali-Eldin A., Wang B., Shenoy P. The hidden cost of the edge: A performance comparison of edge and cloud latencies. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–12. doi:10.1145/3458817.3476142
 21. Kovacevic I., Harjula E., Glisic S., Lorenzo B., Ylianttila M. Cloud and edge computation offloading for latency limited services. *IEEE Access*, 2021, no. 9, pp. 55764–55776. doi:10.1109/access.2021.3071848
 22. Drolia U., Guo K., Tan J., Gandhi R., Narasimhan P. Cachier: Edge-caching for recognition applications. *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 276–286. doi:10.1109/icdcs.2017.94
 23. Bassoli R., Granelli F., Arzo S. T., Di Renzo M. Toward 5G cloud radio access network: An energy and latency perspective. *Transactions on Emerging Telecommunications Technologies*, 2021, vol. 32, no. 1, pp. 1–16. doi:10.1002/ett.3669
 24. Rajalakshmi P., Tammana P. Optimizing latency for real-time traffic and road safety applications through MEC-based V2X system. *IEEE 2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2023, pp. 1–8. doi:10.1109/smartnets58706.2023.10215515
 25. Ioannou A., Weber S. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Communications Surveys & Tuto-*

rials, 2016, vol. 18, no. 4, pp. 2847–2886. doi:10.1109/COMST.2016.2565541

- 26. Ahmadi M., Roberts J., Leonardi E., Movaghar A.** Poster: Impact of traffic characteristics on request aggregation in an NDN router. *Proceedings of the 2019 IFIP Networking Conference (IFIP Networking)*, 2019, pp. 1–2. doi:10.13140/RG.2.2.30376.11528
- 27. Dehghan M., Jiang B., Dabirmoghaddam A., Towsley D.** On the analysis of caches with pending interest tables. *Proceedings of the 2nd ACM Conference on In-*

formation-Centric Networking, 2015, pp. 69–78. doi:10.1145/2810156.2810161

- 28. Wallace R., Andrade X. G., Kayser P., Luo Z., Mukherjee H., Nunes R., Warrior M.** Models of network delay. *Proceedings of the International Workshop on Statistical Modelling*, 2024, pp. 231–238. doi:10.1007/978-3-031-65723-8_36
- 29. Iversen V. B.** *Telettraffice Engineering and Network Planning*. DTU Fotonik, 2015. 382 p.

UDC 004.77

doi:10.31799/1684-8853-2026-2-36-48

EDN: TYQXCS

Development of models for calculating content access delays with account taken of caching in NDN networks

S. I. Iakimenko^a, Lecturer, Junior Researcher, orcid.org/0000-0002-0698-1298, syakimenko@hse.ru

^aA. N. Tikhonov Moscow Institute of Electronics and Mathematics (MIEM HSE), 34, Tallinskaya St., 123458, Moscow, Russian Federation

Introduction: Caching reduces content access latency by bringing it closer to users and offloading the cloud layer of the network. Edge servers are used for this purpose, but the standard TCP/IP stack does not fully utilize caching capabilities. Therefore, it is necessary to consider a cache model using the Named Data Networking stack, which is more suited for this function. **Purpose:** To develop models for estimating content access latency, taking into account the specifics of caching at an intermediate node in NDN. **Results:** We propose a simulation model implementing the Least Recently Used caching algorithm based on discrete event modeling, as well as a mathematical model of delays in receiving content. These models allow for an adequate assessment of the latency on each segment from the user to the cloud and for the system as a whole. Additionally, they make it possible to calculate key cache performance characteristics: cache hit probability and load on external and internal channels. Edge server-based caching decreases content access latency to 65 ms and reduces operator costs for accessing the remote cloud to 60%. **Practical relevance:** The results of this study can be used by internet service providers and telecom operators to reduce backbone bandwidth congestion and improve subscriber quality of service when downloading popular content. **Discussion:** The study used a scenario with a large catalog and a Poisson flow of edge cache requests, which corresponds to a regime in which the probability of request aggregation in the Pending Interest Table is low. Studying the dynamic effect of aggregation on content access latency requires a more complex traffic model (e.g., an ON-OFF or shot-noise model), which is the subject of further research.

Keywords – caching, Least Recently Used, latency, non-zero download delay, edge caching, Named Data Networking, simulation modeling, cloud.

For citation: Iakimenko S. I. Development of models for calculating content access delays with account taken of caching in NDN networks. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 2, pp. 36–48 (In Russian). doi:10.31799/1684-8853-2026-2-36-48, EDN: TYQXCS

Financial support

The research was supported by a grant from the Russian Science Foundation No. 24-19-00299, <https://rscf.ru/project/24-19-00299/>

References

1. Yakimenko S. I., Koucheryavy Y. A. Cooperative caching of content in NDN scenarios for VANET. *Electrosvyaz*, 2023, no. 9, pp. 9–17 (In Russian). doi:10.34832/ELSV.2023.46.9.002, EDN: HUNFMD
2. Amadeo M., Ruggeri G. Exploring in-network computing with information-centric networking: Review and research opportunities. *Future Internet*, 2025, vol. 17 (1), no. 42, pp. 1–28. doi:10.3390/fi17010042
3. Kovalchukov R., Glazkov R., Srikanteswara S., Zhang Y., Moltchanov D., Arrobo G., Feng H., Spoczynski M., Himayat N. In-network dynamic compute orchestration over mobile edge systems. *Proceedings of the 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–7. doi:10.1109/VTC2023-Spring57618.2023.10200992
4. Pirmagomedov R., Srikanteswara S., Moltchanov D., Arrobo G., Zhang Y., Himayat N., Koucheryavy Y. Augmented computing at the edge using named data networking. *Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6. doi:10.1109/GCWkshps50303.2020.9367585
5. Vladyko A., Tambovtsev G., Podgornaya E., Chelloug S. A., Alkanhel R., Plotnikov P. Cluster-based vehicle-to-everything model with a shared cache. *Mathematics*, 2023, vol. 11(13), no. 3017, pp. 1–19. doi:10.3390/math11133017
6. Kurose J. Information-centric networking: The evolution from circuits to packets to content. *Computer Networks*, 2014, vol. 66, pp. 112–120. doi:10.1016/j.comnet.2014.04.002
7. Jacobson V., Smetters D. K., Thornton J. D., Plass M. F., Briggs N. H., Braynard R. L. Networking named content. *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, 2009, pp. 1–12. doi:10.1145/1658939.1658941
8. Zhang L., Afanasyev A., Burke J., Jacobson V., Claffy K. C., Crowley P., Papadopoulos C., Wang L., Zhang B. Named data networking. *ACM SIGCOMM Computer Communication Review*, 2014, vol. 44, no. 3, pp. 66–73. doi:10.1145/2656877.2656887
9. Borodin A. S., Rozhkov M. A., Kirichek R. V., Koucheryavy A. E. New IP, from 5G to 6G: Do we need a paradigm change? *Electrosvyaz*, 2020, no. 7, pp. 15–21 (In Russian). doi:10.34832/ELSV.2020.8.7.002, EDN: CSEHIN
10. Grebeshkov A. Y., Borovskaya Y. A. Constructing of information-oriented 5G-ICN networks. *Vestnik svyazi*, 2021, no. 11, pp. 13–18 (In Russian). EDN: IDYVMD
11. Hasslinger G., Ntougias K., Hasslinger F., Hohlfeld O. Analysis of the LRU cache startup phase and convergence time and error bounds on approximations by Fagin and Che. *IEEE 2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2022, pp. 254–261. doi:10.23919/wiopt56218.2022.9930556
12. Fricker C., Robert P., Roberts J., Sbihi N. Impact of traffic mix on caching performance in a content-centric network.

- Proceedings of the 2012 IEEE INFOCOM Workshops*, 2012, pp. 310–315. doi:10.1109/INFCOMW.2012.6193511
13. Garetto M., Leonardi E., Martina V. A unified approach to the performance analysis of caching systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2016, vol. 1, no. 3, pp. 1–28. doi:10.1145/2896380
 14. Amadeo M., Campolo C., Ruggeri G., Molinaro A. Beyond edge caching: Freshness and popularity aware IoT data caching via NDN at internet-scale. *IEEE Transactions on Green Communications and Networking*, 2021, vol. 6, no. 1, pp. 352–364. doi:10.1109/TGCN.2021.3124452
 15. Yakimenko S. Aspects of caching in information-oriented networks. *Trudy 4-j Mezhdunarodnoj konferencii "Sovremennye setevye tekhnologii (MONETEC-2022)"* [Proc. 4th Int. Conf. "Modern Network Technologies-2022 (MONETEC-2022)"]. Moscow, 2022, pp. 70–76 (In Russian). EDN: NUPDPT
 16. Blinnikov M. A., Pirmagomedov R. Ya., Moltchanov D. A., Koucheryavy A. E. Application of named-data technologies in wireless mesh networks. *Electrosvyaz*, 2019, no. 11, pp. 22–28 (In Russian). EDN: RHQFJM
 17. Wibowo T. A., Syambas N. R., Hendrawan L. V., Yovita R. A. Closer towards named data networking implementation. *International Journal of Intelligent Engineering and Systems*, 2023, vol. 16, no. 1, pp. 265–276. doi:10.22266/ijies2023.0228.24
 18. Putevskaya I. V. Studying efficiency of data caching strategies in information-centric networks. *Vestnik Chuvashskogo Universiteta*, 2024, no. 4, pp. 128–140. doi:10.47026/1810-1909-2024-4-128-140, EDN: SXLOVC
 19. Glazkov R., Moltchanov D., Srikanteswara S., Samuylov A., Arrobo G., Zhang Y., Feng H., Himayat N., Spoczynski M., Koucheryavy Y. Provisioning of fog computing over named-data networking in dynamic wireless mesh systems. *Sensors*, 2024, vol. 24 (4), no. 1120, pp. 1–24. doi:10.3390/s24041120
 20. Ali-Eldin A., Wang B., Shenoy P. The hidden cost of the edge: A performance comparison of edge and cloud latencies. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–12. doi:10.1145/3458817.3476142
 21. Kovacevic I., Harjula E., Glisic S., Lorenzo B., Ylianttila M. Cloud and edge computation offloading for latency limited services. *IEEE Access*, 2021, no. 9, pp. 55764–55776. doi:10.1109/access.2021.3071848
 22. Drolia U., Guo K., Tan J., Gandhi R., Narasimhan P. Cacher: Edge-caching for recognition applications. *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 276–286. doi:10.1109/icdcs.2017.94
 23. Bassoli R., Granelli F., Arzo S. T., Di Renzo M. Toward 5G cloud radio access network: An energy and latency perspective. *Transactions on Emerging Telecommunications Technologies*, 2021, vol. 32, no. 1, pp. 1–16. doi:10.1002/ett.3669
 24. Rajalakshmi P., Tammana P. Optimizing latency for real-time traffic and road safety applications through MEC-based V2X system. *IEEE 2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2023, pp. 1–8. doi:10.1109/smartnets58706.2023.10215515
 25. Ioannou A., Weber S. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, 2016, vol. 18, no. 4, pp. 2847–2886. doi:10.1109/COMST.2016.2565541
 26. Ahmadi M., Roberts J., Leonardi E., Movaghar A. Poster: Impact of traffic characteristics on request aggregation in an NDN router. *Proceedings of the 2019 IFIP Networking Conference (IFIP Networking)*, 2019, pp. 1–2. doi:10.13140/RG.2.2.30376.11528
 27. Dehghan M., Jiang B., Dabirmoghaddam A., Towsley D. On the analysis of caches with pending interest tables. *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, 2015, pp. 69–78. doi:10.1145/2810156.2810161
 28. Wallace R., Andrade X. G., Kayser P., Luo Z., Mukherjee H., Nunes R., Warrior M. Models of network delay. *Proceedings of the International Workshop on Statistical Modelling*, 2024, pp. 231–238. doi:10.1007/978-3-031-65723-8_36
 29. Iversen V. B. *Teletraffic Engineering and Network Planning*. DTU Fotonik, 2015. 382 p.