

УДК 316.776

**Я. Ю. Навроцкий, Н. В. Пацей**

Белорусский государственный технологический университет

**РЕАЛИЗАЦИЯ ПОЛИТИК КЭШИРОВАНИЯ  
В ИНФОРМАЦИОННО-ОРИЕНТИРОВАННЫХ СЕТЯХ**

Статья посвящена исследованию системы кэширования информационно-ориентированных сетей (ICN). Рассмотрены основные практически пригодные для ICN политики кэширования: Leave Copy Everywhere, Leave Copy Down, Copy With Probability и Probabilistic Cache. Помимо политик кэширования немаловажную роль играет стратегия замещения кэша. Рассмотрены алгоритмы замещения: семантическая замена, замена случайной выборкой и замена Least Recently Used. Построена программная модель ICN сети с возможностью выбора политики кэширования и стратегии замещения из числа перечисленных выше, количества активных пользователей, размера кэша и с другими характеристиками.

Представлены результаты математического моделирования работы модели ICN сети. Оценивались вероятности попадания в кэш в зависимости от количества запросов пользователей и зависимость «протяженности» (процента пройденного пути для получения данных) от количества запросов для реализованных политик кэширования сети. Проведенные исследования показали, что Leave Copy Everywhere и Probabilistic Cache больше всего подходят для организации ICN, обеспечивая вероятность попадания в кэш на 7–12% выше и минимальный путь для извлечения данных на 5–18% меньше по сравнению с остальными.

**Ключевые слова:** ICN, кэширование, алгоритм, сети.

**Ya. Yu. Navrotskiy, N. V. Patsei**

Belarusian State Technological University

**CACHING POLICIES IMPLEMENTATION IN INFORMATION-CENTRIC NETWORKS**

The article is devoted to the investigation of the caching system for information-oriented networks (ICN). The main practical ICN caching policies are considered: Leave Copy Everywhere, Leave Copy Down, Copy With Probability and Probabilistic Cache. In addition to cache policies, the cache replacement strategy plays an important role. A semantic replacement, replacement by random sampling and the replacement of Least Recently Used were considered. A software model of the ICN network was built with a choice of cache policy and substitution strategy from the number listed above, the number of active users, the size of the cache and other characteristics.

The results of mathematical modeling of the ICN network model operation are presented. Estimated the probability of cache hit, depending on the number of user requests and the dependency of the “length” (percentage of the distance traveled to retrieve data) from the number of requests for implemented network caching policies. There are shown that Leave Copy Everywhere and Probabilistic Cache are most suitable for organizing ICN. It provide the probability of caching hit by 7–12% higher and the minimum way to extract data is 5–18% less compared to the rest.

**Key words:** ICN, caching, algorithm, networks.

**Введение.** Информационно-ориентированная сеть (ICN) – это новая концепция сетевой архитектуры, разработанная для замены, текущей широко используемой хост-ориентированной архитектуры. ICN обладает рядом преимуществ в обеспечении безопасности данных и клиентов сети. Так, ICN обеспечивает конфиденциальность происхождения данных и конфиденциальность пользователя, чего в хост-ориентированной архитектуре достичь достаточно сложно [1].

Основная идея информационно-ориентированной сети заключается в том, что данные более важны, чем узел, предоставляющий доступ к ним. Это произошло из-за изменения характера использования Интернета, который сейчас более

ориентирован на контент, чем на то, кто предоставляет доступ к данным. Это, например, совместное использование файлов, социальные сети или поиск агрегированных данных. Первоначально концепция ICN была предложена в рамках проекта Стэнфордского университета TRIAD. В рамках данного проекта предлагалось использовать адресацию на основе имени. С тех пор исследователи предложили несколько архитектур (рис. 1). Так, в 2007 году был разработан проект сетевой архитектуры DONA (UC Berkeley), которая улучшила безопасность и архитектуру, предложенную в проекте TRIAD. Архитектура Publish Subscribe Internet Technology (PURSUIT), является продолжением архитектуры Publish Subscribe Internet Routing Paradigm

(PSIRP), оба из которых финансируются программой EU Framework 7 (FP7). Для реализации данных архитектур предложили стек протокола публикации/подписки (publish/subscribe), который должен в дальнейшем заменить стек протокола IP. Ван Якобсон, научный сотрудник PARC, предложил архитектуру Content-Centric Network (CCN). В настоящее время ведется работа по оптимизации архитектуры CCN, называемой Named-Data-Networks (NDN) [1, 2].

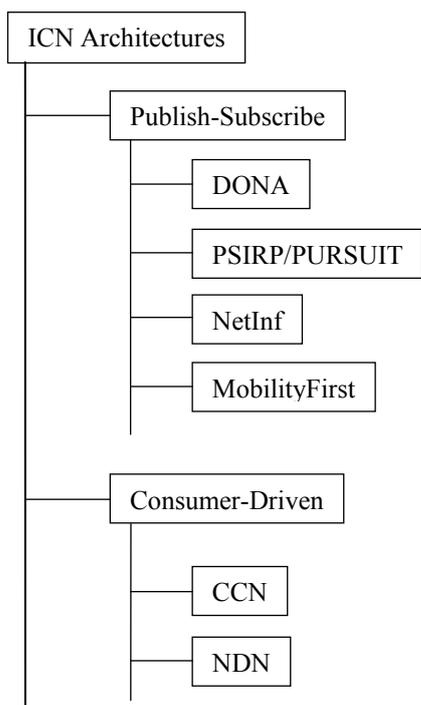


Рис. 1. Архитектуры информационно-ориентированных сетей

Существующие подходы к информационно-ориентированным сетям отличаются с точки зрения реализации, но они имеют одинаковую цель – улучшить производительность и обеспечить более удобный доступ к данным.

ICN уже длительное время остается исследовательским проектом по созданию сетей нового поколения из-за нерешенных проблем, связанных с именованнием и маршрутизацией объектов передачи, безопасностью, мобильностью и кэшированием данных. Таким образом, для полного перехода на ICN нужно решить ряд проблем, связанных с безопасностью и конфиденциальностью, именованнием объектов, мобильностью и кэшированием. В настоящей статье предлагается программная модель ICN и анализируются политики кэширования данных.

**Основная часть.** Для того чтобы увеличить производительность в ICN используется внутреннее хранилище данных, эффективность которого зависит от политики кэширования и стратегии замены данных в кэше.

Кэш представляет собой аппаратный или программный компонент, способный хранить данные, которые могут быть запрошены с большей вероятностью. Доступ к данным в кэше осуществляется быстрее, чем выборка исходных данных из более медленной памяти или удаленного источника, однако объем кэша значительно ограничен по сравнению с хранилищем исходных данных. Данные, хранящиеся в кэше, могут быть результатом более раннего вычисления или дублирования данных, хранящихся в другом месте. Чем больше данных можно забрать из кэша, тем быстрее работает система. Обычно размер кэша относительно невелик, чтобы быть экономически выгодным и обеспечить эффективное использование данных.

**Принцип работы кэша.** Кэш состоит из набора записей. Каждая запись хранит данные, которые являются копией данных из удаленного хранилища. Также каждая запись имеет тег, который указывает на оригинал данных из хранилища. Когда пользователь запрашивает данные, система обработки запроса сначала проверяет кэш. Если в кэше найдена запись с тегом запрашиваемых данных, то система вернет данные из кэша. Ситуация, когда запрашиваемые данные найдены в кэше, называется попаданием кэша. Обратная ситуация, когда происходит обращение к кэшу и кэш не содержит запрашиваемых данных, называется промахом кэша. В случае промаха в кэше данные получаются из удаленного хранилища записываются в кэш, и будут возвращены при следующем запросе. При поступлении данных в кэш система обычно удаляет определенные записи, чтобы освободить место для новых данных. Эвристика, используемая для выбора удаляемой записи, называется политикой замещения. Для эффективности кэша используют политики замещения в зависимости от размера хранимого содержимого, а также задержек и пропускной способности как для кэша, так и для удаленного хранилища. Рассмотрим основные политики [3, 4] кэширования, которые будут реализованы в программной модели сети.

**Политика кэширования Leave Copy Everywhere (LCE).** При использовании политики кэширования Leave Copy Everywhere данные будут сохранены в каждом узле на пути от источника данных к пользователю. Пример работы политики кэширования представлен на рис. 2.

Пользователь отправляет запрос, роутер  $R_0$  проверяет свой кэш на наличие запрашиваемых данных, если происходит промах кэша, роутер передает запрос дальше по сети до тех пор, пока не произойдет попадание кэша. Например,

попадание кэша происходит в роутере  $R_2$ . Роутер возвращает запрашиваемые данные и по пути от роутера  $R_2$  к пользователю данные будут сохранены в роутерах  $R_1$  и  $R_0$ . При следующем же запросе данные будут получены из  $R_0$ .

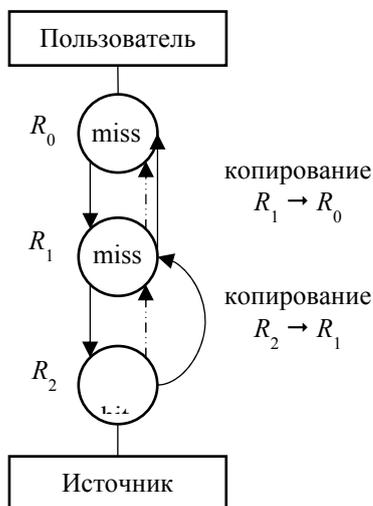


Рис. 2. Схема работы политики кэширования LCE

*Политика кэширования Leave Copy Down (LCD).* При использовании данной политики кэширования данные будут сохранены только в соседнем узле от узла-источника, на пути от источника данных к пользователю. Схема работы политики кэширования представлен на рис. 3.

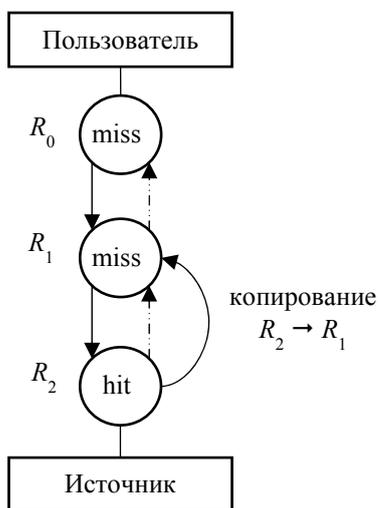


Рис. 3. Схема работы политики кэширования LCD

После попадания кэша в роутер  $R_2$  данные будут сохранены в роутере  $R_1$ , так как данный роутер является прямым соседом  $R_2$ . При последующем запросе данные будут получены из  $R_1$  и сохранены в  $R_0$ . Таким образом, с каждым запросом данные будут кэшироваться ближе к пользователю.

*Политика кэширования Copy with Probability (Prob).* При использовании политики кэширования Copy With Probability данные будут сохраняться в каждом узле на пути от источника к пользователю с заданной вероятностью  $p$ . Схема работы данного алгоритма представлен на рис. 4.

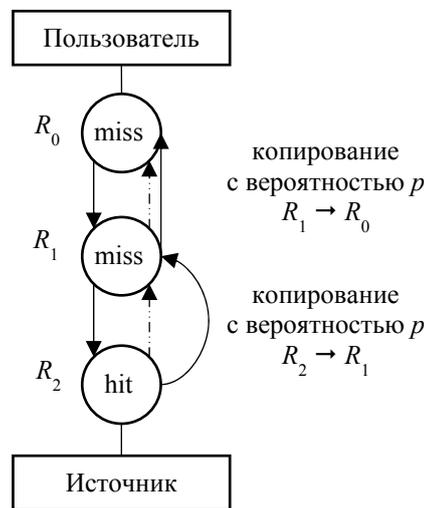


Рис. 4. Схема работы политики кэширования Prob

При заданной вероятности  $p = 1$ , эта политика кэширования будет работать как LCE.

*Политика кэширования Probabilistic Cache (ProbCache).* При Probabilistic Cache данные будут сохранены в каждом узле на пути от источника к пользователю, но с определенной вероятностью  $p$ , которая рассчитывается на основании расстояния между пользователем и источником данных. Вероятность, с которой данные будут кэшированы в узле, определяется отношением количества хопов от источника данных к пользователю к количеству хопов от пользователя к источнику данных.

**Стратегии замещения данных в кэше.** Размер кэша не позволяет хранить данные долго. Основное различие между кэшем и основным хранилищем – время, которое данные могут находиться в памяти. В настоящее время изучены и предложены несколько политик замещения в ICN.

*Семантическая замена.* Семантическая замена – это стратегия замены данных в кэше, основанная на определенном синтаксисе и правилах. В данной стратегии администратор кэша определяет теги, поля, состояния, по которым будет проходить отбор данных на замещение. В семантической замене основным недостатком является строгий набор правил, по которым происходит отбор, так, при добавлении нового типа данных потребуются полная реорганизация установленных правил [3, 5].

*Замена случайной выборкой.* В этом случае замена происходит на основе равномерного распределения. Как правило, именно эту политику замены оспаривают, поскольку иногда замещаются наиболее важные данные.

*Замена Least Recently Used (LRU).* Стратегия LRU является наиболее эффективной политикой замены данных в кэше. При нехватке памяти будут замещены те данные, к которым обращались меньше всего.

*Анализ характеристик политик кэширования.* Эффективность политики кэширования можно оценить по тому, как часто данные берутся из кэша. Так, чем больше данных можно вернуть из кэша, тем эффективнее реализованная политика кэширования [5]. Эффективность политики кэширования определяется по формуле [5, 6]

$$\text{CacheHit} = \frac{\sum_{i=1}^N \text{hits}_i}{\left(\sum_{i=1}^N \text{hits}_i\right) + \left(\sum_{i=1}^N \text{miss}_i\right)}, \quad (1)$$

где  $N$  – это количество узлов в сети;  $\text{hits}_i$  – количество попаданий в кэш в узле  $i$ ;  $\text{miss}_i$  – количество промахов в кэше в узле  $i$ .

Еще одна характеристика – «протяженность». Ее можно рассматривать как процент пройденного пути для получения данных. Чем меньше пройденный путь, тем эффективнее политика кэширования. Протяженность определяется по формуле [5, 6]:

$$\text{Stretch} = \frac{\sum_{i=1}^R \text{hops\_walked}_i}{\sum_{i=1}^R \text{total\_hops}_i}, \quad (2)$$

где  $R$  – это количество запросов;  $\text{hops\_walked}_i$  – количество хопов от пользователя до узла, предоставляющего данные;  $\text{total\_hops}_i$  – количество хопов от пользователя до узла с оригинальными данными [6].

При анализе характеристик политик кэширования должны учитываться такие параметры, как топология сети, количество запросов (пользователей), размер каталога, размер кэша. Поэтому была разработана упрощенная программная модель на языке программирования высокого уровня контент-ориентированной сети с возможностью настройки размеров кэша и каталога, политики кэширования. В качестве стратегии замены кэша была установлена *Least Recently Used (LRU)*, согласно статье [4]. Количество пользователей для ускорения полного цикла моделирования было снижено (до семи активных пользователей). В связи с этим были установлены размеры кэша и размеры каталога 100 и 10 000 соответственно. Остальные параметры варьировались в диапазонах, указанных далее в таблице.

Параметры ICN

Параметр	Значение
Размер каталога, Мб	$10^4$
Размер кэша, Мб	$10^2$
Количество запросов от пользователя, ед.	$10^5, 10^4, 10^3, 10^2$
Количество пользователей, ед.	7
Анализируемые политики кэширования	LCE, LCD, Prob ( $p = 0,25$ ), ProbCache
Политика замены в кэше	LRU

На рис. 5 и 6 представлены соответственно зависимость вероятности попадания в кэш (CacheHit) в зависимости от количества запросов и протяженности (Stretch).

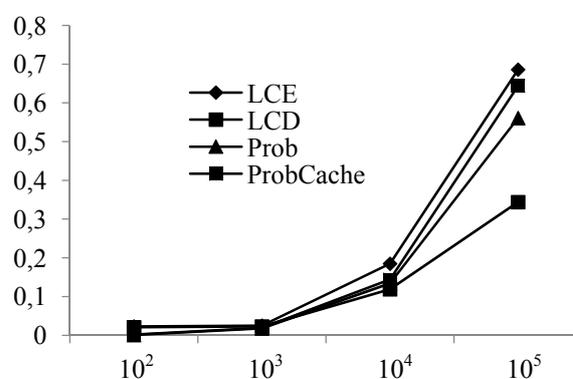


Рис. 5. Вероятности попадания в кэш в зависимости от количества запросов

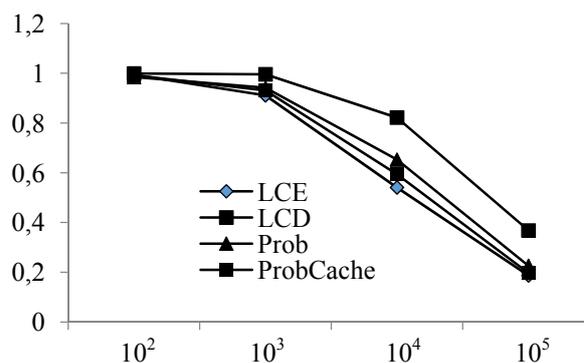


Рис. 6. Зависимость «протяженности» от количества запросов

**Заключение.** В настоящей работе были исследованы новые особенности организации кэша ICN.

Как видно из графиков, реализация политики LCE показывает лучший результат, но в этом случае будет большая избыточность данных, так как данные будут сохраняться на всем пути, и нет гарантии, что они потребуются повторно. Стратегия кэширования ProbCache обес-

печивает почти такой же результат, как и LCE, но обладает рядом преимуществ. Использование данной стратегии кэширования позволит хранить данные в ближайшем узле, сохраняя место в нисходящих узлах для более востребованного контента.

Политика кэширования определяет эффективность работы всей ICN сети, и выбрать одну определенную довольно сложно, требуется учитывать достаточно много факторов, которые пока не учитывались. Исходя из результа-

тов анализа полученных зависимостей, можно сказать, что политики кэширования LCE и ProbCache подходят для организации ICN больше всего, обеспечивая вероятность попадания в кэш на 7–30% выше, а минимальный путь для извлечения данных на 5–25% короче по сравнению с остальными.

Исследование ICN кэширования все еще находится на ранней стадии. Существует много нерешенных как теоретических, так и технических вопросов.

### Литература

1. Xylomenos G., Ververidis C. N., Siris V. A. A Survey of Information-Centric Networking Research // *IEEE Communications Surveys & Tutorials*. 2014. Vol. 16. P. 204–215.
2. Jaber G., Patsei N. V. Information Centric Networking for web-based content distribution and manipulation // *Proceedings of BSTU. Series 3, Physics and mathematics. Informatics*. 2017. No. 2. P. 88–91.
3. Psaras I., Chai W. K., Pavlou G. Probabilistic in-network caching for information-centric networks // *ACM SIGCOMM Workshop ICN*, 2012. P. 345–351.
4. Zhang G., Li Y., Lin T. Caching in information centric networking: A survey // *Computer Networks*. 2013. Vol. 57, no. 16. P. 3128–3141.
5. Rossini G., Rossi D. A dive into the caching performance of content centric networking // *CAMAD. IEEE*. 2012. P. 105–109.
6. Abdullahi I., Arif S., Hassan S. Survey on caching approaches in information centric networking // *Journal of Network and Computer Applications*. 2015. Vol. 56. P. 48–59.

### References

1. Xylomenos G., Ververidis C. N., Siris V. A. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys & Tutorials*, 2014, vol. 16, pp. 204–215.
2. Jaber G., Patsei N. V. Information Centric Networking for web-based content distribution and manipulation. *Proceedings of BSTU, series 3, Physics and mathematics. Informatics*, 2017, no. 2, pp. 88–91.
3. Psaras I., Chai W. K., Pavlou G. Probabilistic in-network caching for information-centric networks. *ACM SIGCOMM Workshop ICN*, 2012, pp. 345–351.
4. Zhang G., Li Y., Lin T. Caching in information centric networking: A survey. *Computer Networks*, 2013, vol. 57, no. 16, pp. 3128–3141.
5. Rossini G., Rossi D. A dive into the caching performance of content centric networking. *CAMAD. IEEE*, 2012, pp. 105–109.
6. Abdullahi I., Arif S., Hassan S. Survey on caching approaches in information centric networking. *Journal of Network and Computer Applications*, 2015, vol. 56, pp. 48–59.

### Информация об авторах

**Навроцкий Ярослав Юрьевич** – магистрант. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: yaroslav.navrotskiy.yn@gmail.com

**Пацей Наталья Владимировна** – кандидат технических наук, исполняющая обязанности заведующего кафедрой программной инженерии. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: n.patsei@belstu.by

### Information about the authors

**Navrotskiy Yaroslav Yur'yevich** – Master's degree student. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: yaroslav.navrotskiy.yn@gmail.com

**Patsei Nataliya Vladimirovna** – PhD (Engineering), Acting Head of the Department of Programming engineering. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: n.patsei@belstu.by

Поступила 28.11.2017