

ПРИНЦИПЫ МАРШРУТИЗАЦИИ В ПРОЕКТАХ ИНФОРМАЦИОННО-ОРИЕНТИРОВАННЫХ СЕТЕЙ

Адресация определяет доступность информационных объектов в сети путем сопоставления имен с местами размещения (процесс называется разрешение имен). Это эквивалентно сетевому уровню и включает размещение контента, запросы маршрутизации, пересылку при распространении и доставке контента. В информационно-ориентированных сетях (ИОС) для большинства архитектур используется коммуникационная парадигма, основанная на шаблоне *publish/subscribe* (публикации/подписки) [1]. Шаблон *publish/subscribe* основан на двух функциональных шагах: *REGISTER* (или *PUBLISH*) и *FIND* (или *SUBSCRIBE*). ICN может накладываться на любой уровень пересылки, включая IP. При реализации маршрутизации используют два основных метода:

– однофазное разрешение имен: маршрутизация на основе имени (*NBR – Name-Based Routing*) [2], переадресация на основе *FIB (FIB-based forwarding)* [3];

– двухфазное разрешение имен: *Name Resolver Service (NRS, служба преобразования имен)* для преобразования идентификатора объекта в один или несколько локаторов источников [4], связь на основе рандеву (*Rendezvous-based*) [5] или вероятностная маршрутизация (*probabilistic routing*) (например, с использованием фильтров Блума) [6].

При однофазном разрешении имен сообщение с запросом направляется напрямую от запрашиваемой стороны к источнику или любому кэшу, который может обслуживать запрос. Вместо обмена информацией о маршруте на основе объявлений префикса IP, как это происходит в текущей маршрутизации *BGP (Border Gateway Protocol)*, маршрутизаторы в однофазном разрешении имен объявляют имена информационных объектов, чтобы каждый маршрутизатор мог определить эффективный маршрут к объекту. Переадресация состоит в поиске наилучшего соответствия между именем запрошенного информационного объекта и записями таблиц маршрутизации.

Двухфазный подход, основанный на модели публикации/подписки, опирается на промежуточный прокси-сервер сопоставления для преобразования имен объектов в сетевые идентификаторы

(или локаторы), которые используются для маршрутизации запросов к источникам контента. Пользователи отправляют сообщение о подписке на прокси с именем нужного информационного объекта. Прокси отвечает за возврат запрошенного объекта пользователям. Протоколы маршрутизации обычно основаны на распределенных хэш-таблицах (*DHT – Distributed Hash Tables*) для распространения имени на функцию отображения локатора в глобальной сети.

В ходе анализа маршрутизации, используемой в проектах информационно-ориентированных сетях: CCN (Content-Centric Networking) [3], NDN (Named Data Networking) [7], DONA (Data-Oriented Network Architecture) [2], Network of Information (NetInf), CBCB (Combined Broadcast and Content Based) и PURSUIT (Publish Subscribe Internet Technologies) было установлено, что в проектах NetInf/MDHT, DONA, SAIL, MobilityFirst и PURSUIT используется подход разрешения имен (*NR*), а CBCB, CCN/NDN используют подход маршрутизации на основе имен (*NBR* или *RBNR – Route-By-Name Routing*).

Подход *NR* может гарантировать обнаружение любого контента в сети за ограниченное (по размеру сети) число переходов. С другой стороны, *NBR* не гарантирует обнаружение контента. Вместо этого они имеют высокую вероятность обнаружения контента, которая в большинстве случаев пропорциональна количеству посещенных узлов. Поскольку количество объектов данных очень велико (например, 10^{15} до 10^{22}), размер таблиц маршрутизации становится проблемой, так как он может быть пропорционален количеству объектов данных, если не введен механизм агрегирования. С другой стороны, *NBR* снижает общую задержку и упрощает процесс маршрутизации за счет исключения процесса разрешения.

Накладные расходы на сообщения обновления в подходах с *NR* ниже, чем в подходах *NBR*, поскольку в последнем случае для распространения обновлений требуется обновить всю сеть. Однако сбой узла в системе *NR* могут сделать часть индекса недоступным, даже если содержимое доступно. Эта проблема не существует в подходах маршрутизации *NBR*, поскольку они могут обнаруживать альтернативные пути маршрутизации из-за лавинной рассылки сообщений (в NDN) или широковещательной рассылки (в CBCB). Требования к хранилищу для *NR* намного выше, чем для подхода маршрутизации на основе *NBR*. Наконец, подход *NR* обычно требует поддержки двух баз данных: сопоставления имени и IP-адреса в системе разрешения и информации о доступности IP-адресов в системе маршрутизации, тогда как для подходов маршрутизации *NBR* требуется только поддержки-

вать сопоставление между именем и сетевым местоположением.

Для маршрутизации контента в ИОС можно определить ряд требований, сформулированных в таблице.

Таблица – Требования к маршрутизации для архитектур ИОС

<i>Требования</i>	<i>Формулировка</i>
Состояние контента	обеспечивать базовые операции на сетевом уровне с малой задержкой для регистрации контента (оригинала, реплики или кэша), обновления и удаления (данные вопросы в рассмотренных выше проектах не проработаны)
Обнаружение ближайшей копии	перенаправление запроса контента на ближайшую (на основе некоторой сетевой метрики) копию. Обеспечивает сокращение междоменного трафика
Разрешение и определение местоположения	разрешение имени и поиск не должны выходить за пределы сетевого домена, содержащего источник и контент
Гарантия обнаружения	необходимо гарантировать обнаружение любого существующего контента, независимо от частоты использования и уровня репликации
Масштабируемость	необходимо найти компромисс между длиной маршрутизации (соотношение между длиной пути маршрутизации и минимальной длиной пути) и размером таблицы маршрутизации, учитывая большое количество имен и физические ограничения, накладываемые технологиями хранения
Развертывание на сетевом уровне	оптимально процесс извлечения контента должен быть одношаговым либо путем объединения либо полного исключения части разрешения имен
Инфраструктура безопасности	обеспечения целостности и происхождения контента (рассмотренные схемы используют пары открытого/закрытого ключей, однако в контексте ИОС это в значительной степени неисследованная область)

Однако достижение их всех в одной схеме маршрутизации может оказаться труднореализуемой или невозможной

ЛИТЕРАТУРА

1. Eugster P., Felber P., Guerraoui R. The many faces of publish/subscribe // ACM Computing Surveys, vol. 35, no. 2, 2003. – p. 114-131.
2. Koponen T. A data-oriented (and beyond) network architecture // ACM SIGCOMM Computer Communication Review, vol. 37, no. 4, 2014. – p. 181-192.
3. Jacobson V., Smetters D.K., Thornton J.D, Plass M. Networking Named Content // In Proceedings of ACM CoNEXT, 2009. – p. 1-12.
4. Ambrosio M.D., Marchisio M., Vercellone V. Deliverable D6.2

of the 4WARD project // Second NetInf Architecture Description, 2010. – URL: http://www.4ward-project.eu/index.php?s=file_download&id=70 (дата обращения: 17.11.2022).

5. Conceptual Architecture of PSIRP Including Subcomponent Descriptions, Public Deliverable (D2.2) of the PSIRP project. – URL: http://www.psirp.org/files/Deliverables/FP7-INFISO-ICT-216173-PSIRP-D2.2_ConceptualArchitecture_v1.1.pdf (дата обращения: 10.07.2022).

6. Ahmed R., Boutaba R. Distributed Pattern Matching: A Key to Flexible and Efficient P2P Search // IEEE Journal on Selected Areas in Communications (JSAC) issue on Peer-to-Peer Communications and Applications, vol. 25, no. 1, 2007. – p. 73-83.

UDC 316.776

N.V. Patsei, PhD stud. G. Jaber
(BSTU, Minsk)

SEMANTIC INFORMATION-CENTRIC NETWORKING MODELING RESULTS

To compare Semantic Information-Centric Networking (SICN) with other Name Data Networking (NDN) projects it was built a Python model under some assumptions [1]. Model has following components:

- *Publisher*: main content source;
- *Subscriber*: data content user;
- *Search engine*: make data translation from informal to formal form;
- *DNS*: used to find IP of data source;
- *Cache*: saving data.

To reduce model complexity was held the following notations and assumptions: u – is the number of users ($u = 10$); n - publisher depth (it is defined as number of extended branches from root of tree from a subscriber to data source, n is variable); e - search engine depth (defined as number of branches from root of tree from a subscriber to search engine, supposed that $e = n$); d - DNS depth (is defined as number of branches from root of tree from a subscriber to DNS, $d = n/2$); c – cache depth (defined as number of branches from root of tree from a subscriber to cache, supposed that $c = n/2$); s – sharing coefficient (is defined as the ratio of shared of links by subscribers to total links. $s = 0.25$); r – sharing factor (is defined as the utilization factor from sharing paths between subscribers, $r = 1 + s(u - 1)$); L – total number of extended branches for each subscriber to data source $L = 2^{(n+1)} - 2$ (is supposed that each node has two branches).