формационной безопасности. Таким образом, интеграция структурированных моделей представления знаний, векторного поиска на основе косинусного сходства и механизмов дополненной генерации позволяет создавать масштабируемые и безопасные интеллектуальные интерфейсы, пригодные для эксплуатации в профессиональных экспертных системах.

ЛИТЕРАТУРА

1. Методы представления знаний и фреймовые структуры: лекция.URL: http://www.machinelearning.ru/wiki/images/1/1d/Lect_8_ai_mdv.pdf (дата обращения: 25.11.2025).

2. Семантические сети и обработка естественного языка; Knowledge Graph. URL: https://www.osp.ru/os/2017/02/13052229 (дата обращения: 11.12.2025).

3. Методы представления информации в простых семантических сетях; классификация семантических сетей / О. В. Мосин [и др.] // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20, № 3. С. 382–393. URL: https://cyberleninka.ru/article/n/metody-predstavleniya-informatsii-v-prostyh-semanticheskih-setyah (дата обращения: 16.12.2025). DOI: 10.17586/2226-1494-2020-20-3-382-393.

УДК 004.77; 004.72

Е.А. Гончар ассист.
(БГТУ, г. Минск)

ОБЗОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИМУЛЯЦИИ NDN СЕТЕЙ

NDN (Named Data Networking) [1–5] – это архитектура сети, которая ориентирована на данные, а не на их местоположение. В отличие от традиционных IP-сетей, где коммуникация основана на адресах устройств (IP-адресах), в NDN фокус смещается на сами данные и их имена. Это позволяет создавать более гибкие, безопасные и эффективные сети, особенно в условиях современного интернета, где запросы на контент играют ключевую роль.

Основные цели NDN:

1. Ориентация на контент: В NDN пользователи запрашивают данные по их именам, а не по адресам устройств. Это упрощает доступ к информации, особенно в условиях, когда данные могут быть доступны из множества источников.

2. Повышение безопасности: Данные в NDN подписываются на уровне архитектуры, что обеспечивает их аутентичность и целостность.

3. Эффективность: NDN позволяет кэшировать данные на промежуточных узлах сети, что снижает нагрузку на серверы и ускоряет доставку контента.

4. Устойчивость к изменениям топологии сети: поскольку данные идентифицируются по именам, а не по местоположению, сеть может адаптироваться к изменениям, таким как отказ узлов или изменение маршрутов.

Для исследований и разработки в области NDN было создано несколько специализированных инструментов, среди которых выделяются симулятор ndnSIM [6], эмулятор Mini-NDN [7] и универсальная среда моделирования OMNeT++ [8] с соответствующими модулями. Выбор конкретного инструмента целиком зависит от задач исследователя: нужно ли ему протестировать новый сетевой алгоритм на огромной топологии, отладить работу реального приложения или же визуализировать и изучить каждый аспект взаимодействия протоколов.

Симулятор ndnSIM является самым популярным инструментом для моделирования NDN и фактически признан стандартом в академической среде. Он реализован в качестве модуля для NS-3 – мощного симулятора с дискретно-событийным механизмом. Ключевое преимущество ndnSIM заключается в том, что он не просто имитирует концепции NDN, а использует реальный код ключевых компонентов архитектуры: библиотеки ndn-cxx и демона форвардинга NFD [9]. Однако это не запуск готовых бинарных файлов, а интеграция их кода для симуляции поведения, что обеспечивает высокую достоверность результатов. Симуляция работает не в реальном времени, а дискретно-событийно, что позволяет проводить эксперименты значительно быстрее, чем они протекали бы в реальности. Благодаря этому, а также поддержке параллельных вычислений с использованием MPI, ndnSIM способен моделировать очень крупные сети, включая тысячи узлов. Кроме того, наследуя возможности NS-3, он предоставляет готовые и хорошо проработанные модели для различных сред передачи данных, включая проводные и беспроводные технологии, такие как WiFi и LTE. Сбор и анализ данных в ndnSIM хорошо автоматизированы – доступны встроенные трассировщики и скрипты для пост-обработки логов. Тем не менее, у этого подхода есть и минусы. Сценарии для ndnSIM пишутся на языке C++, что может быть сложнее для исследователей, привыкших к скриптовым языкам. Сами приложения также

необходимо разрабатывать на C++ с использованием библиотеки ndn-cxx, и для переноса реального приложения в симуляцию потребуется его адаптация и внесение изменений в код. Наконец, взаимодействие с симулированной сетью ограничено просмотром статистики – невозможно напрямую подключиться к процессу NFD для отладки.

Альтернативой симуляции выступает эмуляция, и главный инструмент здесь – Mini-NDN. Он построен на базе Mininet и представляет собой легковесный эмулятор, который запускает на одном физическом компьютере полноценную сеть NDN, где на каждом виртуальном узле работают реальные демоны – Named Data Networking Forwarding Daemon (NFD) и протокол маршрутизации NLSR. Mini-NDN использует сетевые пространства имен Linux для изоляции процессов и виртуальные Ethernet-соединения для связи между узлами. Параметры каналов (задержка, потери, пропускная способность) настраиваются с помощью утилиты Linux tc на виртуальных интерфейсах. Это означает, что эмулируемая сеть ведет себя почти как реальная, так как в ней выполняются неизмененные бинарные файлы NDN. Главное достоинство Mini-NDN – возможность "бесшовного" переноса кода: приложения, написанные и отлаженные в эмуляторе с использованием клиентских библиотек на C++, Python, Java или JavaScript, можно без каких-либо изменений разворачивать на реальном тестовом стенде. Сценарии эмуляции пишутся на Python, что делает их более простыми и быстрыми в разработке. Кроме того, исследователь может напрямую интерактивно взаимодействовать с процессами NFD или NLSR на любом узле в ходе эксперимента, что дает уникальные возможности для отладки и мониторинга. Однако у эмуляции есть и недостатки. Mini-NDN работает в реальном времени, что ограничивает масштаб и скорость экспериментов по сравнению с дискретно-событийной симуляцией. Хотя он подходит для сетей среднего и крупного размера, с появлением кластерной версии, моделирование с тысячами узлов может быть затруднительным. Встроенных средств визуализации у Mini-NDN нет, а для сбора статистики (например, о потерях пакетов или задержках) исследователю необходимо самостоятельно настраивать сбор логов NFD и NLSR или использовать утилиты вроде ndndump, а затем писать собственные скрипты для их обработки. Кроме того, на данный момент поддержка беспроводных технологий в Mini-NDN находится в стадии разработки.

Третий инструмент, OMNeT++, стоит несколько особняком. Это не специализированное решение для NDN, а мощная среда дискретно-событийного моделирования с модульной архитектурой. Для моделирования NDN под нее разрабатываются отдельные фреймворки, такие

как inbaverSim или фреймворк для оценки NDN в Интернете вещей (IoT). OMNeT++ позволяет собирать модель сети как конструктор из отдельных модулей, что дает исследователю непревзойденную гибкость в настройке протоколов и алгоритмов. Одним из главных его преимуществ являются великолепные инструменты визуализации и отладки, позволяющие наблюдать за работой сети в графическом интерфейсе вплоть до уровня отдельных пакетов и событий. Это делает OMNeT++ идеальным выбором для глубокого академического изучения новых протоколов и механизмов NDN, где требуется детальное понимание каждого аспекта их работы. Однако, эта гибкость и детализация могут обернуться более высокой сложностью освоения и меньшей производительностью при моделировании очень больших сетей по сравнению с узкоспециализированным ndnSIM.

Таким образом, выбор инструмента диктуется конкретной задачей. Если ваша цель – исследование масштабируемости нового алгоритма маршрутизации в сети с тысячами узлов, тестирование стратегий кэширования или моделирование работы NDN поверх беспроводных каналов, ваш выбор – ndnSIM. Если же вы разрабатываете реальное NDN-приложение, хотите протестировать его в условиях, максимально приближенных к реальности, отладить взаимодействие компонентов, а затем легко перенести его на действующий стенд, вам идеально подойдет Mini-NDN. Наконец, для сложных исследовательских проектов, где требуется абсолютная прозрачность процессов, возможность детальной настройки протоколов и наглядная визуализация, наилучшей средой станет OMNeT++ с его NDN-модулями. Каждый из этих инструментов занимает свою нишу, и вместе они формируют мощную экосистему для развития и внедрения архитектуры Named Data Networking.

ЛИТЕРАТУРА

1. Jacobson V.,Smetters D.K., Thornton J.D., Plass M.F., Briggs N.H., Braynard R. L. Networking named content // CoNEXT '09: Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies, New York, 2009. P. 1–12.

2. Mastorakis S., Afanasyev A., Moiseenko I., Zhang L. NdnSIM 2: An updated NDN simulator for NS-3. NDN // Technical Report NDN-0028, Revision 2., Los Angeles, 2011, P. 1–8.

3. Zhang L., Estrin D., Burke J., Jacobson V., Thornton J. D., Smetters D. K., Study of Censorship in Named Data Networking. // Advanced Multimedia and Ubiquitous Engineering: Future Information Technology, 2016, vol. 2, P. 145–152. DOI:10.1007/978-3-662-47895-0-18

4. Zhang L., Afanasyev A., Burke J., Jacobson V., Claffy K. C., Crowley P., Zhang B. Named data networking. SIGCOMM, series 44, Computer Communication Review Named Data Networking, 2014, issue 3, P. 66–73.

5. Afanasyev A., Shi J., Zhang B., Zhang L., Moiseenko I., Yu Y., Wang, L. NFD developer's guide. // Technical Report NDN-0028, Revision 2, Los Angeles, 2016, P. 29–31.

6. Riley G. F., Henderson T. R. The ns-3 network simulator. Modeling and tools for network simulation, 2010, pp. 15–34. DOI:10.1007/978-3-642-12331-3_2.

7. Budiana M. S. et al. Impact of the Content Store Scaling toward the LRU and FIFO Cache Replacements on NDN using Mini-NDN //2021 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA). – IEEE, 2021. – С. 1-5.

8. Varga A. OMNeT++ //Modeling and tools for network simulation. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. – С. 35-59.

9. Afanasyev A. et al. NFD developer's guide //Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021. – 2014. – Т. 29. – С. 31.

УДК 004

Н.И. Белодед, доц., канд. техн. наук
(БГТУ, г. Минск)

ИСПОЛЬЗОВАНИЕ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ В ОБРАЗОВАНИИ

Показатели обучения.

Учебный план в системе высшего образования определяет дисциплины, формы отчётности и компетенции, которые должен освоить выпускник. Современный результат обучения выражается не только в оценках, но и в сформированных компетенциях.

Компетенции формируются на основе государственных стандартов и требований работодателей. Они включают:

– Универсальные (УК) – коммуникация, критическое мышление, работа в команде.

– Базовые профессиональные (БПК) – умения, обеспечивающие профессиональную деятельность.

– Специализированные (СК) – углублённые знания по профилю.