УДК 00471; 004042

ПРОГРАММНОЕ МОДЕЛИРОВАНИЕ УСТРОЙСТВА ОБРАБОТКИ СЕТЕВОГО ТРАФИКА В ИНФОРМАЦИОННОЙ СИСТЕМЕ

© **К. И. Будников, А. В. Курочкин**

Институт автоматики и электрометрии СО РАН,
630090, г. Новосибирск, просп. Академика Коптюга, 1
E-mail: budnikov@iae.nsk.su

Представлен метод имитационного компьютерного моделирования сетевого устройства с использованием его цифрового эмулятора, для которого создаётся виртуальное окружение, предназначенное для генерации сетевого трафика. В процессе эмуляции моделируемые объекты задаются множеством функциональных и интерфейсных потоков, взаимодействующих через виртуальные линии связи, в роли которых выступают общие области памяти. Предложенный подход позволяет минимизировать временны́е издержки передачи пакетов между цифровыми объектами и сосредоточить внимание на алгоритмической составляющей моделируемого устройства. В качестве иллюстрации метода показано имитационное компьютерное моделирование работы устройства фильтрации протокола HTTP в составе информационной Web-системы.

Ключевые слова: имитационное компьютерное моделирование сетевого устройства, цифровой эмулятор, фильтрация HTTP-трафика.

DOI: 10.15372/AUT20210308

Введение. Совершенствование сетевых устройств — актуальная задача при улучшении работы компьютерных сетей с точки зрения увеличения их пропускной способности и повышения скорости передачи информации. В рамках этого процесса моделирование сетевых устройств является неотъемлемой частью их разработки. Однако в публикациях по сетевому моделированию и созданных для этого прикладных пакетах программ фокус направлен на вопросы, связанные со структурами и топологиями проектируемых компьютерных сетей, а также видами пропускаемого через них трафика [1, 2]. В то же время архитектуре разрабатываемых устройств и совершенствованию алгоритмов их работы, на наш взгляд, уделяется недостаточно внимания. Многие создаваемые эмуляторы сетевого оборудования имеют упрощённое устройство и предназначаются в основном для решения задач обучения навыкам работы с ними обслуживающего персонала или студентов [3, 4].

Современное сетевое устройство, которое обрабатывает пакеты на уровнях выше канального (маршрутизатор, фильтр, межсетевой экран), представляет собой специализированный сетевой компьютер, снабжённый операционной системой и прикладным программным обеспечением, которое реализует его функциональность. Создание его прототипа на основе стандартного компьютера с набором сетевых интерфейсов и программного обеспечения, эмулирующего работу прибора, — один из применяемых методов имитационного моделирования в настоящее время [5]. Для отработки используемых решений посредством сравнения получаемых характеристик устройства проводятся стендовые испытания. В них прототип работает под искусственно создаваемой нагрузкой, имитирующей условия, в которых разрабатываемый прибор будет функционировать в реальной системе. Для этого используется специализированный стенд, снабжённый оборудованием для моделирования. Данный подход сопряжён с временны́ми и материальными затратами. Кроме того, имею-

щиеся для создания прототипа сетевые интерфейсы могут иметь недостаточные на момент проектирования характеристики пропускной способности и рассматриваться как «узкое место» в архитектуре устройства в целом.

Цель данной работы — создание метода имитационного компьютерного моделирования сетевого устройства на базе альтернативного подхода, основанного на эмуляции в памяти компьютера как моделируемого устройства, так и стендового оборудования, используемого в испытаниях.

В этом случае посредством многопоточного приложения эмулируется как устройство (или его логическая часть, например, один из каналов), так и потоки пакетов, проходящих через него. Для подобного подхода характерны низкие затраты. Он даёт возможность создания функционирующей модели и проведения исследования свойств устройства со сложной архитектурой и алгоритмом обработки пакетов, в то время как сам прибор находится в стадии разработки. Также исключается влияние сетевой инфраструктуры на работу цифрового эмулятора, что даёт более точное понимание алгоритмов его функционирования. Представлена иллюстрация разработанного метода на примере моделирования функционирования HTTP-фильтра в составе информационной Web-системы. Аналогичные решения применяются и в других научно-технических направлениях [6, 7].

1. Описание метода. В рамках предлагаемого подхода создаётся цифровой эмулятор сетевого устройства (маршрутизатора, фильтра, межсетевого экрана) и вспомогательных приборов, необходимых для построения испытательного стенда. Множество всех потоков эмуляции S представляет собой совокупность множеств потоков модели устройства M и вспомогательных устройств $A_1, \ldots, A_n$:

$$S = M \cup A_1 \cup A_2 \cup \ldots \cup A_n. \tag{1}$$

Множество потоков каждого устройства D, под которым подразумевается как модель устройства M, так и вспомогательные устройства A_i, делится на функциональные $F = \{f_1, f_2, \ldots, f_n\}$ и интерфейсные $X = \{x_1, x_2, \ldots, x_m\}$. Функциональные потоки отвечают за функции устройства, а интерфейсные — за взаимодействие виртуальных устройств между собой:

$$D = F \cup X. \tag{2}$$

Потоки взаимодействуют посредством обмена информацией через общие области памяти компьютера. Таким способом эмулируются линии связи и исключается стадия передачи информации по физической среде (кабель, оптоволокно, радиоканал), что существенно упрощает и ускоряет процесс обмена, исключая его из списка «узких мест» при моделировании. На рис. 1 в качестве примера представлена схема многопоточного приложения-эмулятора. Исследуемое устройство NM, которое может быть мостом или шлюзом, моделируется набором из шести функциональных потоков $F_{NM} = \{f_1, f_2, \ldots, f_6\}$. Оно имеет

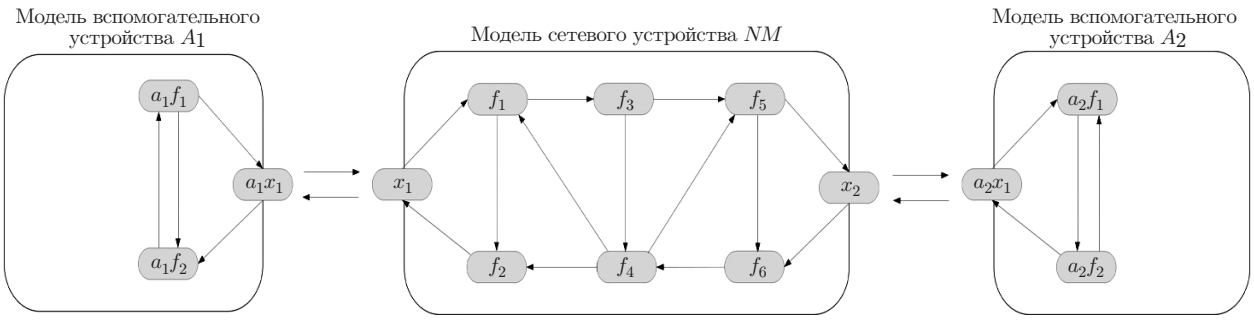

Рис. 1. Схема многопоточного приложения-эмулятора стенда устройства NM

два виртуальных сетевых интерфейса $X_{NM} = \{x_1, x_2\}$. К ним подсоединены вспомогательные устройства A_1 и A_2, которые генерируют трафик через модель, посылая встречные пакеты согласно используемому протоколу. Их функциональность обеспечивается потоками $F_{A1} = \{a_1f_1, a_1f_2\}$ и $F_{A2} = \{a_2f_1, a_2f_2\}$. С моделируемым устройством NM они взаимодействуют посредством интерфейсных потоков $X_{A1} = \{a_1x_1\}$ и $X_{A2} = \{a_2x_1\}$. В итоге цифровой стенд MS состоит из 14 моделирующих потоков:

$$MS = NM \cup A_1 \cup A_2 = F_{NM} \cup X_{NM} \cup F_{A1} \cup X_{A1} \cup F_{A2} \cup X_{A2}. \tag{3}$$

Современные компьютеры с многоядерными процессорами дают возможность создать экономичное решение для подобной эмуляции. Они позволяют разместить в своей памяти виртуальную цифровую копию реального испытательного стенда, состоящую из нескольких устройств, и запрограммировать его работу.

2. Синтез цифрового эмулятора Web-системы. Изложенный подход применён при создании виртуального стенда и имитационном моделировании работы устройства фильтрации пакетов по протоколу HTTP в информационной Web-системе. Для эмуляции была взята система, которая представляет собой совокупность Web-сервера, клиентов, посылающих запросы к расположенным на нём ресурсам, и фильтрующего устройства, регулирующего доступ клиентов к ресурсам (рис. 2).

Модель фильтра состоит из двух одинаковых каналов, которые обеспечивают прохождение через устройство трафика и его фильтрацию. Каждый канал содержит модули чтения сетевых пакетов (МЧП), сортировки пакетов (МСП) и передачи пакетов (МПП). Центральный модуль модели — анализатор пакетов (МАП), общий для обоих каналов. Взаимодействие с линиями связи происходит через модули сетевых интерфейсов МСИ1 и МСИ2. Более подробно работа фильтрующего устройства в моделируемой системе рассмотрена в [5].

Целью моделирования стало исследование метода фильтрации запроса к Интернет-ресурсу по его адресу URL [8], который является наиболее популярным в настоящее время. Его можно признать наиболее сбалансированным в отношении имеющихся достоинств и недостатков. Метод фильтрации позволяет осуществлять селективный подход, в рамках которого происходит анализ запросов к информационным ресурсам, расположенным, в том числе и на одном IP-адресе, и запрещать доступ в тех случаях, когда это необходимо. Основное внимание в исследовании было сосредоточено на проблеме наиболее быстрого прохождения запросов через фильтр. Сравнивались два алгоритма: стандартный и усовершенствованный.

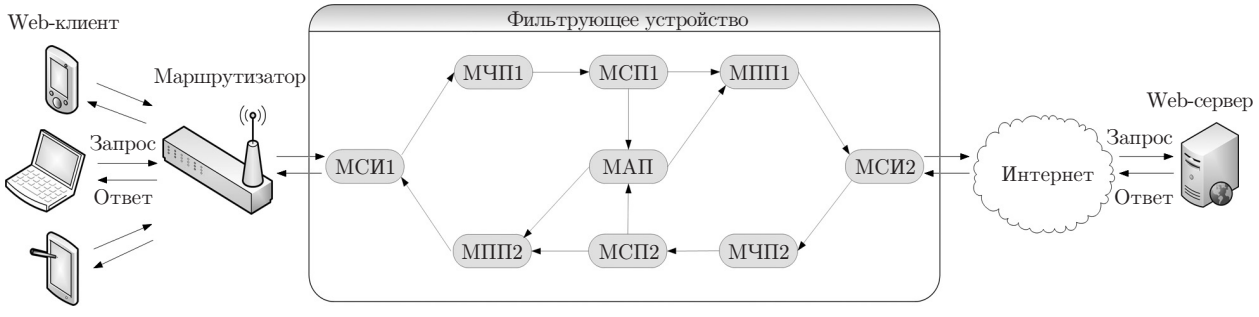

Рис. 2. Моделируемая Web-система

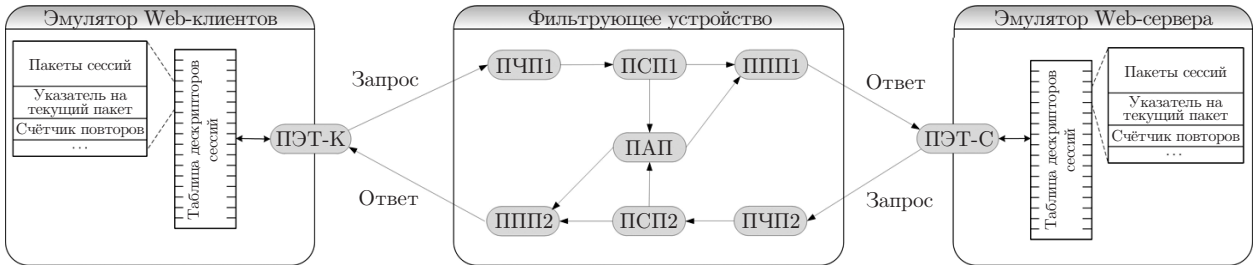

Рис. 3. Цифровой эмулятор моделируемой Web-системы (виртуальный стенд)

Стандартный алгоритм фильтрации по адресу URL, который описан, например, в [9], предполагает предварительную проверку запроса при поступлении в устройство, и только по её результатам запрос либо пропускается дальше, либо блокируется. Эта процедура занимает время, связанное с приёмом и декодированием запроса, выделением из него адреса URL и поиском его в списках запрещённых адресов. На этот период запрос задерживается фильтром. В процессе работы над уменьшением задержки авторами был создан усовершенствованный алгоритм фильтрации с постанализом запросов [10, 11]. Теоретически предполагалось, что он позволит получить сокращение времени ожидания ответа от Web-сервера и увеличение пропускной способности фильтрующего устройства и, как результат, системы в целом.

В целях сравнения алгоритмов было проведено имитационное компьютерное моделирование на основе предложенного метода. Для того чтобы получить результат, не зависящий от линий связи и сетевых интерфейсов, соединяющих компоненты системы, а содержащий только сопоставление времён обработки запросов к Web-серверу и его ответов, был создан цифровой эмулятор, в который вошли представленная выше модель фильтрующего устройства, а также эмуляторы Web-сервера и Web-клиентов (рис. 3). Перечисленные элементы составили виртуальный стенд для проведения имитационных испытаний.

Из модели фильтра были удалены модули сетевых интерфейсов (МСИ1 и МСИ2), поскольку обмен пакетами в цифровом эмуляторе системы ведётся через общие области памяти. Другие модули остались без изменения, и каждому из них был сопоставлен одноимённый поток. Это потоки чтения (ПЧП), сортировки (ПСП), анализа (ПАП) и передачи (ППП) пакетов. Эмуляторы Web-сервера и Web-клиентов построены на основе модуля эмуляции трафика (МЭТ). Оба используемых МЭТ имеют одинаковую архитектуру и алгоритм работы, а роль каждого из них в качестве Web-клиента или сервера задаётся при конфигурации цифрового эмулятора системы. Движущим компонентом МЭТ является поток эмуляции трафика (ПЭТ). Для эмулятора Web-клиентов — это ПЭТ-К, для эмулятора Web-сервера — ПЭТ-С. Таким образом, созданная модель системы MS_{Web} основана на девяти взаимодействующих друг с другом потоках:

$$MS_{\text{Web}} = \{\text{ПЧП1}, \text{ПСП1}, \text{ППП1}, \text{ПЧП2}, \text{ПСП2}, \text{ППП2}, \text{ПАП}, \text{ПЭТ-К}, \text{ПЭТ-С}\}, \tag{4}$$

$$F_{\text{Web}} = \{\text{ПЧП1}, \text{ПСП1}, \text{ППП1}, \text{ПЧП2}, \text{ПСП2}, \text{ППП2}, \text{ПАП}, \text{ПЭТ-К}, \text{ПЭТ-С}\}, \tag{5}$$

$$X_{\text{Web}} = \{\text{ПЧП1}, \text{ППП1}, \text{ПЧП2}, \text{ППП2}, \text{ПЭТ-К}, \text{ПЭТ-С}\}, \tag{6}$$

где ПСП1, ПСП2 и ПАП — только функциональные потоки. Остальные содержат как функциональную, так и интерфейсную составляющие.

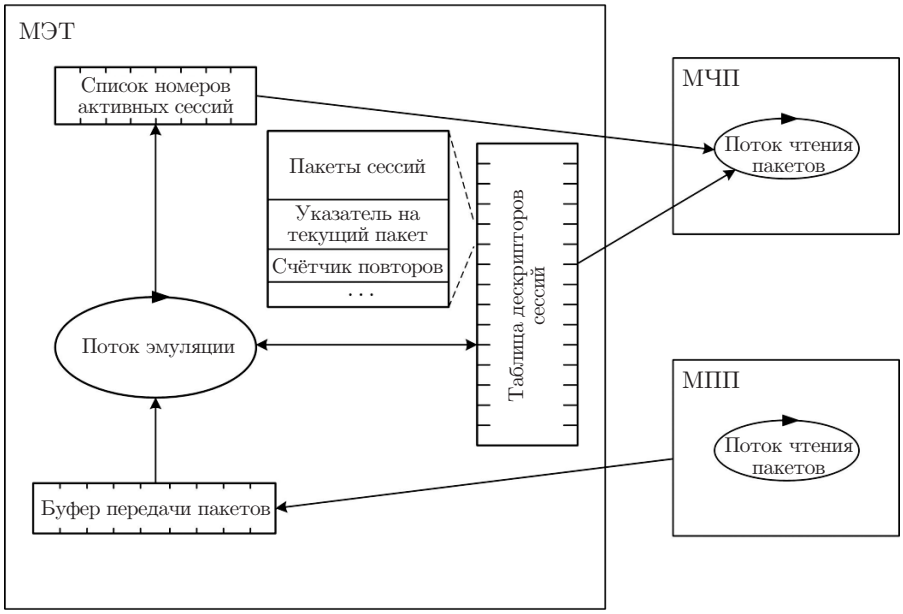

Рис. 4. Структура модуля эмуляции трафика и схема его взаимодействия с ПЧП и ППП фильтрующего устройства

3. Методика эмуляции. Для организации процесса виртуальной работы системы используются специальные файлы, содержащие копии сетевых пакетов, которыми обменивались реальные Web-клиент и Web-сервер во время HTTP-сессии. Каждый пакет в таком файле имеет ряд дополнительных признаков, главным из которых является идентификатор отправителя пакета, позволяющий организовать трафик для проведения эксперимента. Модули эмуляции трафика выполняют основную работу по его имитации для функционирования системы. На рис. 4 показана подробная схема модуля и его взаимодействие с потоками чтения и передачи пакетов фильтрующего устройства.

В каждый МЭТ входят четыре основных компонента:

1. Поток эмуляции трафика, который обеспечивает работу модуля по имитации операций приёма и передачи пакетов.

2. Таблица дескрипторов (описаний) сессий. В таблице содержится вся информация о процессе эмуляции HTTP-сессий в ходе испытания. Каждый элемент таблицы содержит уникальный вариант образа файла с пакетами сессии, указатель на текущий обрабатываемый пакет, счётчик повторов данной сессии и ряд других вспомогательных параметров.

3. Список номеров (индексов в таблице) активных сессий. HTTP-сессии, участвующие в процессе создания трафика для эмуляции функционирования Web-системы, могут находиться в активном или неактивном состоянии. В активной сессии доступен текущий пакет, предназначенный для приёма модулем чтения пакетов фильтрующего устройства. В неактивных сессиях этот пакет недоступен. При активизации сессии её номер добавляется потоком эмуляции трафика в конец списка.

4. Буфер передачи пакетов. В буфере накапливаются пакеты, пришедшие от ППП. Дисциплина обслуживания пакетов соответствует методу FIFO.

Запуск эксперимента по эмуляции функционирования Web-системы происходит в следующей последовательности. Прежде всего, загружается в память компьютера и инициируется виртуальный стенд для проведения имитационных испытаний. В рамках этого процесса оба МЭТ получают свои роли: один — клиента, другой — сервера, формируются их таблицы дескрипторов сессий. В каждый дескриптор записываются пакеты HTTP-сессии, загружаемые из заданного для текущего эксперимента файла. При этом они мо-

дифицируются таким образом, чтобы сделать любую сессию уникальной по отношению к остальным и упростить задачу определения принадлежности каждого пакета к сессии. В качестве текущего устанавливается первый пакет. Счётчик повторов генерации сессии обнуляется. Буфер передачи пакетов создаётся пустым. Потоки эмуляции ПЭТ-К и ПЭТ-С устанавливаются в состояние ожидания приёма пакетов от ППП1 и ППП2 фильтрующего устройства.

Список номеров активных сессий для каждого МЭТ формируется по-разному. В модуле, который производит обмен со стороны Web-сервера, список создаётся пустым, поскольку эмулятор находится в режиме ожидания прихода пакетов от клиентов. При загрузке МЭТ Web-клиента в данный список заносятся номера всех сессий для активации потока запросов со стороны клиентов к серверу.

После того как программой эмуляции запускается поток чтения ПЧП1, он обнаруживает активные сессии на стороне Web-клиента и, сделав случайный выбор, начинает читать текущие пакеты и посылать их через фильтр к эмулятору Web-сервера. От ППП1 пакеты поступают в МЭТ на стороне сервера. Там они запускают ПЭТ-С, который начинает их обрабатывать и отправлять ответы через ПЧП2. Ответные пакеты проходят через фильтр, поступают в МЭТ на стороне Web-клиента и пробуждают ПЭТ-К. Таким образом запускается основной цикл процесса эмуляции, в рамках которого создаются потоки пакетов HTTP-сессий, проходящих через фильтр, т. е. происходит моделирование работы системы, показанной на рис. 2.

4. Проведение имитационных экспериментов и обсуждение результатов. С помощью представленного эмулятора Web-системы проведены виртуальные испытания для сравнения алгоритмов фильтрации, ранее упомянутых в разд. 2. Благодаря созданию цифровой копии и исключению стадии передачи информации по физической среде (кабель, оптоволокно, радиоканал), в испытаниях отсутствуют зависящие от этого фактора временны́е издержки на передачу пакетов между виртуальными устройствами системы. Это позволило сосредоточить внимание на алгоритмической составляющей устройства фильтрации и провести экспериментальное сравнение стандартного и усовершенствованного алгоритмов фильтрации по адресу URL. В процессе виртуальных испытаний эмулировалась работа системы с разными алгоритмами фильтрации, различными интенсивностями потоков запросов от клиентов и размерами ответов от Web-сервера от 1 до 100 Кбайт. Эмуляция проходила на офисном компьютере с процессором Intel Core i7 870 4×2,93 Ггц и 4 Гбайт памяти. Исследовались такие характеристики фильтра, как количество обрабатываемых HTTP-сессий за 1 с, интенсивность виртуальных сетевых потоков, время ожидания клиентом ответа от Web-сервера.

На рис. 5 представлены графики результатов проведённых экспериментов по эмуляции функционирования системы при размерах ответов от Web-сервера 1 Кбайт (a) и 100 Кбайт (b). По оси абсцисс отложены интенсивности виртуальных сетевых потоков в эмулируемой системе. По оси ординат указано время ожидания ответа от Web-сервера для стандартного (штриховая кривая) и усовершенствованного с постанализом запросов алгоритмов фильтрации (сплошная кривая). В процессе испытаний суммарная интенсивность виртуального обмена достигала 13,1 Гбит/с (из них 97,7 % составляют трафик сервера, 2,3 % — клиента) при ответах сервера размером в 100 Кбайт и 4,7 Гбит (79,3 % и 20,7 % соответственно) при ответах сервера размером в 1 Кбайт. Количество запросов в секунду доходило до 360200 при ответах сервера размером 1 Кбайт и 14950 при ответах размеров 100 Кбайт. При достигнутой интенсивности трафика выявились преимущества усовершенствованного алгоритма фильтрации.

Проведённые эксперименты по эмуляции обмена между клиентами и сервером с разными алгоритмами фильтрации запросов от клиентов при вариации интенсивности трафика

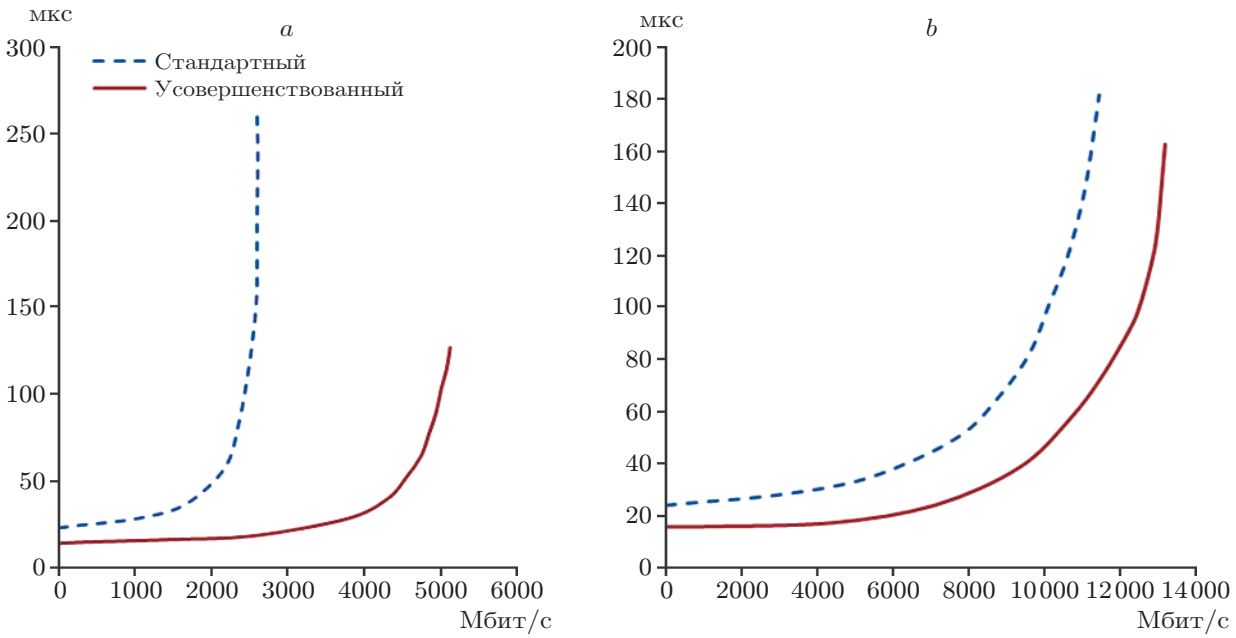

Рис. 5. Зависимости времени ожидания ответа Web-сервера от интенсивности виртуальных сетевых потоков в эмулируемой системе: при размере ответа Web-сервера 1 Кбайт (a) и 100 Кбайт (b)

и размеров ответов показали преимущество усовершенствованного алгоритма. Установлено уменьшение среднего времени ожидания ответа от Web-сервера при прохождении пользовательского запроса к Web-ресурсу через модель устройства фильтрации, которое использовало усовершенствованный алгоритм и работало в режиме постанализа запросов по сравнению с устройством, работающим в стандартном режиме предварительного анализа запросов. Снижение составило до 54 % на линейном участке графика при ответах сервера размером 1 Кбайт и 42 % при ответах сервера размером 100 Кбайт. При использовании усовершенствованного алгоритма пропускная способность фильтра повысилась. Её увеличение достигало 94 % при ответах сервера размером 1 Кбайт и 15 % при ответах сервера размером 100 Кбайт.

Моделирование работы фильтрующего устройства проведено при идеальных характеристиках среды передачи данных, созданных в рамках виртуального цифрового стенда в целях отработки алгоритмов фильтрации. Для реального прибора на той же самой процессорной платформе, снабжённого сетевыми адаптерами, при испытаниях на реальном стенде будут получены результаты, на которые негативно повлияют задержки при передаче пакетов по физическим линиям связи. Характеристики будут отражать реальные возможности фильтрующего устройства.

Заключение. В работе представлен метод имитационного компьютерного моделирования сетевого устройства с помощью построения цифровых эмуляторов прибора и окружающего его оборудования, предназначенного для генерации трафика. В процессе моделирования объекты взаимодействуют через виртуальные линии связи, в роли которых выступают общие области памяти. Данный подход позволяет минимизировать временны́е издержки на передачу пакетов и сосредоточить внимание на алгоритмах моделируемого устройства. К достоинствам метода можно отнести возможность построения функционирующей модели в то время, когда само устройство ещё не создано, а также низкие затраты на моделирование. Недостатком можно считать тот факт, что имитационное мо-

делирование даёт результат в условиях идеальной среды передачи данных, тогда как при стендовом получаются характеристики, которыми будет реально обладать проектируемое устройство.

Метод проиллюстрирован имитационным компьютерным моделированием работы информационной Web-системы, в которую встроено фильтрующее устройство. Исследование проведено путём создания цифрового эмулятора системы, моделирующего фильтрующее устройство, сетевые интерфейсы, линии связи, клиентов и Web-сервер. Это позволило исключить сетевую составляющую системы, поставив в центр исследования фильтрующее устройство. Таким образом, была получена возможность сопоставить различные алгоритмы фильтрации. В процессе моделирования проведено сравнение стандартного алгоритма фильтрации и усовершенствованного алгоритма с использованием постанализа запросов к Интернет-ресурсу. Подтверждены преимущество усовершенствованного алгоритма и перспектива его использования в устройствах фильтрации.

Финансирование. Работа выполнена при поддержке Министерства науки и высшего образования РФ (государственная регистрация № 121042900050-6).

СПИСОК ЛИТЕРАТУРЫ

1. **Rahman M. A., Pakštas A., Wang F. Zh.** Network modelling and simulation tools // Simulation Modelling Practice and Theory. 2009. **17**, N 6. P. 1011–1031. DOI: 10.1016/j.simpat.2009.02.005.
2. **Гудов А. М., Семехина М. В.** Имитационное моделирование процессов передачи трафика в вычислительных сетях // Управление большими системами. 2010. Вып. 31. С. 130–161. URL: http://www.mathnet.ru/links/d79bc2b16e1a1c3a7843188567b5c6db/ubs454.pdf (дата обращения: 12.04.2021).
3. **Liangxu S., Wu I., Zhang Yu., Yin H.** Comparison between physical devices and simulator software for Cisco network technology teaching // Proc. of the 8th Intern. Conference on Computer Science & Education (ICCSE 2013). Colombo, Sri Lanka, 26–28 April, 2013. DOI: 10.1109/ICCSE.2013.6554134.
4. **Золотухин М. С., Симонова Е. С.** Сетевые симуляторы и эмуляторы оборудования Cisco // Современные наукоёмкие технологии. 2020. № 7. С. 57–61. URL: https://top-technologies.ru/ru/article/view?id=38134 (дата обращения: 12.04.2021).
5. **Budnikov K. I., Kurochkin A. V., Lubkov A. A., Yakovlev A. V.** Experimental study of symmetric computer model of HttpFilter // Proc. of the 3rd Russian Pacific Conference on Computer Technology and Applications. Vladivostok, Russia, 18–25 Aug., 2018. DOI: 10.1109/RPC.2018.8482147.
6. **Лях Т. В., Зюбин В. Е., Гаранина Н. О.** Автоматическая верификация алгоритмов управления в киберфизических системах на программных имитаторах // Автометрия. 2019. **55**, № 2. С. 103–113. DOI: 10.15372/AUT20190211.
7. **Белоконь С. А., Золотухин Ю. Н., Филиппов М. Н.** Нечёткая кластеризация в задачах определения аэродинамических характеристик и моделирования динамики летательного аппарата // Автометрия. 2018. **54**, № 5. С. 99–107. DOI: 10.15372/AUT20180513.
8. **Апетьян С., Ковалев А., Файб А.** Фильтрация контента в Интернете. Анализ мировой практики // Фонд развития гражданского общества. 22 мая 2013. URL: http://civilfund.ru/Filtraciya_Kontenta_V_Internete_Analiz_Mirovoy_Praktiki.pdf (дата обращения: 12.04.2021).
9. **Pat. 20060064469 A1 US.** System and method for URL filtering in a firewall / J. Balasubrahmaniyan, K. Daftary, V. R. Yarlagadda, K. Kumar. Publ. 23.03.2006.

10. **Будников К. И., Курочкин А. В., Лубков А. А., Яковлев А. В.** Метод фильтрации HTTP-пакетов на основе постанализа запросов к web-ресурсу // Сибирский физический журнал. 2017. **12**, № 1. С. 13–18. DOI: 10.25205/2541-9447-2018-13-1-5-12.
11. **Budnikov K. I., Kurochkin A. V., Lubkov A. A., Yakovlev A. V.** Regulation of access to web-resource based on post-analysis of http-requests // Proc. of the Intern. Conference Information Technology and Nanotechnology (ITNT 2016). Samara, Russia, 17–19 May, 2016. Vol. 1638. P. 284–289. URL: http://ceur-ws.org/Vol-1638 (дата обращения: 12.04.2021).

Поступила в редакцию 31.03.2021
После доработки 20.04.2021
Принята к публикации 26.04.2021