

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

На правах рукописи

Чемерицкий Евгений Викторович

**ИССЛЕДОВАНИЕ МЕТОДОВ
КОНТРОЛЯ ФУНКЦИОНИРОВАНИЯ
ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ**

05.13.11 – математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата физико-математических наук

МОСКВА – 2015

Работа выполнена на кафедре автоматизации систем вычислительных комплексов факультета вычислительной математики и кибернетики Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный университет имени М. В. Ломоносова».

Научный руководитель: доктор физико-математических наук, член-корреспондент РАН, заведующий Лабораторией вычислительных комплексов факультета вычислительной математики и кибернетики Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный университет имени М. В. Ломоносова», профессор
Смелянский Руслан Леонидович.

Официальные оппоненты: Доктор физико-математических наук, профессор департамента программной инженерии факультета компьютерных наук Федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»
Ломазова Ирина Александровна

Доктор технических наук, заведующий кафедрой Вычислительной техники института автоматики и вычислительной техники «Национальный исследовательский университет «Московский энергетический институт», профессор
Топорков Виктор Васильевич

Ведущая организация: Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Защита диссертации состоится “_____” _____ 2015 г. в ____ часов на заседании диссертационного совета Д 501.001.44 при Московском государственном университете имени М. В. Ломоносова по адресу: 119991, ГСП-1, Москва, Ленинские горы, МГУ, 2-й учебный корпус, факультет вычислительной математики и кибернетики, аудитория 685.

С диссертацией можно ознакомиться в Научной библиотеке Московского государственного университета имени М.В. Ломоносова по адресу: 119192, г. Москва, Ломоносовский проспект, д. 27 – а так же на официальном сайте факультета ВМК МГУ <http://www.cmc.msu.ru> в разделе «Диссертации».

Автореферат разослан “_____” _____ 20__ г.

Ученый секретарь
диссертационного совета Д 501.001.44
доктор физико-математических наук, доцент

О.В. Шестаков

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы диссертации

В данной работе рассматривается задача исследования методов анализа конфигураций и управления поведением программно-конфигурируемых сетей (ПКС) – нового класса компьютерных сетей, ключевыми особенностями которого являются строгое разделение контура передачи данных и контура управления логикой передачи, а так же управление коммутационным оборудованием из логически единого центра – сетевой операционной системы (далее ПКС контроллера, или просто контроллера).

Коммутаторы ПКС представляют собой очень простые устройства в сравнении с роутерами и коммутаторами традиционных сетей. Они лишь исполняют инструкции по обработке проходящих через них пакетов данных, которые контроллер динамически формирует и загружает в них. Они не содержат программного обеспечения, реализующего стек протоколов TCP/IP. Тем самым контроллер получает возможность гранулярной настройки поведения каждого ПКС коммутатора, а сам коммутатор избавляется от множества служебных протоколов, предназначенных, прежде всего, для согласования работы коммутационного оборудования.

Работа посвящена исследованию ПКС, в которых взаимодействие между контроллером и подключёнными к нему коммутаторами обеспечивается протоколом OpenFlow. Низкоуровневые абстракции этого протокола позволяют контроллеру устанавливать и удалять правила обслуживания пакетов непосредственно. Тем самым, OpenFlow освобождает сетевых инженеров от необходимости распутывать нетривиальные зависимости между настроенными параметрами и поведением оборудования, что существенно упрощает администрирование сети.

Перечисленные особенности ПКС и OpenFlow позволили описывать поведение коммутационных устройств набором установленных в них правил обслуживания пакетов и одновременно ограничили изменения конфигурации сети воздействиями контроллера. В результате у исследователей появилась удобная возможность для наблюдения за глобальным состоянием сети, что стимулировало развитие и применение формальных методов для анализа компьютерных сетей.

Настоящая работа посвящена исследованиям в области одного из наиболее многообещающих и востребованных на практике направлений использования формальных методов в контексте ПКС – проверке соответствия поведения ПКС требованиям политик маршрутизации. В работе рассматривается два разных класса требований и соответствующих им свойств ПКС: логические свойства правил маршрутизации пакетов в масштабах сети, а так же максимальное время передачи пакетов по заданному маршруту с учётом взаимного влияния потоков друг на друга.

Цель работы

Целью данной работы является исследование и разработка методов и средств для оценки свойств ПКС по известной конфигурации её компонентов и проверке соответствия этих свойств предъявляемым к ней требованиям политик маршрутизации.

Для достижения поставленной цели необходимо решить следующие задачи:

- Предложить формальный язык для спецификации политик маршрутизации, выражающих свойства достижимости в пространстве состояний пакетов;
- Построить математическую модель ПКС и алгоритм для проверки модельного представления ПКС на соответствие спецификациям предложенного языка;
- Разработать математическую модель ПКС и алгоритм оценки для максимальной сквозной задержки при передаче пакетов через инфраструктуру сети;
- Выполнить программную реализацию и провести экспериментальное исследование предложенных алгоритмов.

Основные результаты работы

Основные *результаты* диссертационной работы следующие:

1. Разработаны язык спецификации политик маршрутизации и реляционная модель поведения ПКС, в рамках которой строго описана семантика этого языка. Впервые было показано, что для задания статических свойств ПКС достаточно логики первого порядка с оператором транзитивного замыкания;

2. Построен алгоритм проверки соответствия конфигурации ПКС заданному множеству политик маршрутизации, для которого определена структура данных, позволяющая эффективно представлять отношения, моделирующие поведение ПКС;
3. Построена математическая модель функционирования ПКС с временем, в рамках которой поставлена задача и построены алгоритмы вычисления верхней оценки задержки передачи пакетов через последовательность коммутаторов с учётом текущей нагрузки на сеть;
4. Все разработанные алгоритмы были реализованы и прошли экспериментальную апробацию.

Научная новизна

Впервые формализм логики первого порядка, дополненной оператором транзитивного замыкания, был использован в качестве языка для спецификации свойств достижимости в ПКС. Для данного языка были разработаны оригинальные методы проверки соответствия конфигурации ПКС его произвольным спецификациям;

Для исследуемого класса сетей впервые был предложен алгоритм вычисления достижимой верхней оценки для задержки передачи пакета, обладающий полиномиальной сложностью.

Практическая ценность

Результаты настоящей работы могут быть использованы в качестве базы для разработки новых методов и средств исследования разнообразных свойств компьютерных сетей по их конфигурациям. Распространение инструментальных средств указанного вида способно снизить затраты на обслуживания сетей и повысить продуктивность сетевых администраторов, наделив их возможностью заблаговременно выявлять несоответствия между настройками сетевого оборудования и требованиями, предъявляемыми к поведению сети. Полученные в работе результаты могут быть рекомендованы к использованию при проектировании и модернизации вычислительных систем реального времени.

Методы исследования

При получении основных результатов работы использовались методы математической логики, формальной верификации, теории сетевого исчисления (network calculus), а так же линейного программирования.

Апробация работы

Результаты работы докладывались на научно-исследовательском семинаре кафедры автоматизации систем вычислительных комплексов (АСВК) факультета ВМК МГУ, на научном семинаре лаборатории вычислительных комплексов кафедры АСВК, а также на следующих конференциях:

1. IV международном семинаре «Программные семантики, спецификации и верификация (PSSV-2013)» (Екатеринбург, 2013);
2. Международной научной конференции «Управление и виртуализация в современных сетях (Сети 2014: SDN & NFV)» (Москва, 2014);
3. Международной научно-практической конференции «Инструменты и методы анализа программ (ТМРА-2014)» (Кострома, 2014);
4. Международном семинаре «Formal Foundations for Networking (Dagstuhl Seminar 15071)» (Дагштуль, 2015).

Публикации

По теме диссертации опубликовано 11 печатных работ, в том числе 3 работы в журналах «Моделирование и анализ информационных систем» (МАИС) и «Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление», входящих в перечень ведущих рецензируемых научных журналов ВАК РФ, а также 5 работ, индексируемых системой Scopus. Список работ приведён в конце автореферата.

Личный вклад

Личный вклад автора заключается в получении основных теоретических результатов, программной реализации и экспериментальной апробации. Вклад соавторов заключается в следующем. В работах [1-2,10] вклад В. А. Захарова и Р.Л. Смелянского ограничен постановкой задачи и обсуждением результатов её решения. В работах [3,7,9] В.А. Захаров предложил адаптировать разработанную автором диссертации модель для решения задачи согласованного обновления конфигурации ПКС. В работах [4,8,11] В.А. Захаров предложил использовать предложенный в рамках диссертации метод верификации ПКС для проверки

конфигурации в оперативном режиме; вклад В.С. Алтухова и В.В. Подымова ограничивается разработкой некоторых программных модулей представленной в этой работе системы. В работе [5] В.С. Алтухов использовал разработанную автором диссертации модель сети для измерения сквозных задержек соединения. В работе [6] вклад Р.Л. Смелянского ограничивается постановкой задачи.

Структура и объём диссертации

Диссертация состоит из введения, двух глав, заключения, трёх приложений и списка литературы. Объём работы – 200 страниц. Список литературы содержит 131 наименование.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность диссертационной работы и дано краткое описание задачи.

В **первой** главе представлены результаты исследований методов и средств проверки соответствия правил передачи пакетов в сети требованиям политик маршрутизации. В этой главе описано устройство ПКС сети, Open Flow протокол и взаимодействие контроллера с ПКС коммутаторами; дана классификация свойств ПКС и приведены примеры типичных политик маршрутизации. Свойства ПКС подразделяются на локальные и глобальные, статические и динамические. Среди динамических свойств ПКС особо выделяются свойства дискретного и реального времени.

Приведен обзор формальных языков, применяемых для описания свойств распределённых вычислительных систем: темпоральные логики CTL, LTL и CTL*, а так же пропозициональное μ -исчисление. Дополнительно рассмотрена логика первого порядка с оператором транзитивного замыкания FO[TC], который оказался удобным для описания разнообразных свойств достижимости. По результатам обзора сделан вывод о перспективности логики транзитивного замыкания FO[TC] для описания статических свойств ПКС, и темпоральной логики линейного времени LTL – для описания динамических свойств ПКС.

Проведён обзор формализмов, подходящих для построения модели ПКС. По итогам проведённого обзора простейшие компоненты ПКС, пакеты и шаблоны применяющихся к ним правил обслуживания – было предложено представлять булевыми наборами. Правила обработки пакетов, реализованных текущей конфигурацией ПКС – отношениями преобразования булевых наборов,

выражающих состояния пакетов в сети. В качестве наиболее перспективного представления для таких отношений были выбраны упорядоченные двоичные решающие диаграммы OBDD. Для описания поведения ПКС построенная реляционная модель конфигурации сети дополнена конечным автоматом, описывающим поведение контроллера.

По результатам проведённых обзоров было разработано две математических модели ПКС: *статической* и *динамической*.

Статическая модель ПКС предназначена для описания сети коммутаторов с заданными таблицами правил OpenFlow. Эта модель не охватывает контроллер и не учитывает возможность изменения таблиц коммутации.

В рамках статической модели ПКС для описания заголовков пакетов, номеров портов и уникальных имен коммутаторов используются двоичные векторы вида $\mathbf{h} = (h_1, h_2, \dots, h_N)$, $\mathbf{p} = (p_0, p_1, \dots, p_k)$ и $\mathbf{w} = (w_1, w_2, \dots, w_m)$ соответственно. Для обозначения отдельных элементов векторов используются квадратные скобки, например, $\mathbf{h}[i] = h_i$.

Если начальный бит номера коммутатора равен единице $\mathbf{p}[0] = 1$, то такой порт называется входным, иначе – выходным портом коммутатора. Множества всех заголовков, входных, выходных и всех портов, а так же уникальных имён коммутаторов обозначаются записями \mathcal{H} , \mathcal{IP} , \mathcal{OP} , \mathcal{P} и \mathcal{W} соответственно.

Выходной порт $\mathbf{p} = (0, 0, \dots, 0)$ считается портом для сброса пакетов и обозначать записью *drop*. Выходной порт $\mathbf{p} = (0, 1, 1, \dots, 1)$ считается выходным портом управления и обозначается записью *octr* – через этот порт коммутатор направляет сообщения контроллеру. Порт $\mathbf{p} = (1, 1, 1, \dots, 1)$ – это входной порт канала управления. Он обозначается записью *ictr*, и через этот порт коммутатор получает команды от контроллера.

Пары $\langle \mathbf{h}, \mathbf{p} \rangle$, $\mathbf{h} \in \mathcal{H}$, $\mathbf{p} \in \mathcal{P}$ называются *локальными состояниями пакетов*, а пары вида $\langle \mathbf{p}, \mathbf{w} \rangle$, $\mathbf{p} \in \mathcal{P}$, $\mathbf{w} \in \mathcal{W}$ – *точками сети*. Тройки $\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle$, $\mathbf{h} \in \mathcal{H}$, $\mathbf{p} \in \mathcal{P}$, $\mathbf{w} \in \mathcal{W}$ называются (*глобальными*) *состояниями пакетов*. Для обозначения множества глобальных состояний пакетов используется символ \mathcal{S} .

Шаблон заголовка называется троичный вектор $\mathbf{z} = (\sigma_1, \sigma_2, \dots, \sigma_N)$, где $\sigma_i \in \{0, 1, *\}$, $1 \leq i \leq N$, а *шаблоном порта* – подобного же рода троичный вектор $\mathbf{y} = (\delta_1, \delta_2, \dots, \delta_k)$. Шаблоны используются как для выбора правил коммутации пакетов, так и для модификации заголовков пакетов.

Предполагается, что при обслуживании пакетов коммутаторы выполняют над ними два типа действий: *действие коммутации* $OUTPUT(\mathbf{y})$, где \mathbf{y} – шаблон порта коммутатора, и *действие модификации заголовка* $SET_FIELD(\mathbf{z})$, где \mathbf{z} – шаблон заголовка пакета.

Конечная последовательность действий называется *инструкцией* обработки пакетов. *Правилом коммутации пакетов* называется тройка $\mathbf{r} = \langle \mathbf{z}, \mathbf{y}, \alpha \rangle$, где \mathbf{z} и \mathbf{y} – это шаблоны заголовка и порта, а α – инструкция.

Топология сети и функциональность коммутаторов описываются бинарными отношениями на множествах точек и локальных состояний пакетов; эти отношения задаются посредством квантифицированных булевых формул. В указанных формулах используются две параметризованные вспомогательные функции $U_\sigma(u, v)$ и $E_\sigma(u)$, где $\sigma \in \{0, 1, *\}$, а u, v – двоичные векторы:

- если $\sigma = *$, то $U_\sigma(u, v) = u \equiv v$ и $E_\sigma(u) = 1$;
- если $\sigma \in \{0, 1\}$, то $U_\sigma(u, v) = u \equiv \sigma$ и $E_\sigma(u) = u \equiv \sigma$.

Действие $a = OUTPUT(\mathbf{y})$ отправляет пакеты без изменения их заголовков во все выходные порты, имена которых подпадают под шаблон $\mathbf{y} = (\delta_1, \delta_2, \dots, \delta_k)$. Действие $b = SET_FIELD(\mathbf{z})$ изменяет заголовки пакетов в соответствии с шаблоном $\mathbf{z} = (\sigma_1, \sigma_2, \dots, \sigma_N)$: бит заголовка пакета h_i остается неизменным, если соответствующий ему символ шаблона σ_i равен «*»; в противном случае этот бит принимает значение σ_i . Семантика обоих действий описывается бинарными отношениями на множестве локальных состояний пакетов $\mathcal{H} \times \mathcal{P}$:

$$R_a(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = \bigwedge_{i=1}^N (\mathbf{h}[i] \equiv \mathbf{h}'[i]) \wedge \bigwedge_{i=0}^k U_{\delta_i}(\mathbf{p}'[i], \mathbf{p}[i])$$

$$R_b(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = \bigwedge_{i=1}^N U_{\sigma_i}(\mathbf{h}'[i], \mathbf{h}[i]) \wedge \bigwedge_{i=0}^k (\mathbf{p}[i] \equiv \mathbf{p}'[i])$$

Инструкция α вычисляет последовательную композицию составляющих ее действий. Если α – пустая последовательность, то пакет сбрасывается.

Семантика инструкции α описывается бинарным отношением R_α , которое определяется так:

- если инструкция α – пустая, то $R_\alpha = \text{false}$;
- если $\alpha = a, \beta$, то отношение R_α задается следующими формулами в зависимости от действия a :
 - если a – это действие коммутации пакета, то
$$R_\alpha(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = R_a(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) \vee R_\beta(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle)$$
 - если a – это действие модификации заголовка пакета, то

$$R_\alpha(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = \exists \mathbf{h}'' (R_\alpha(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}'', \mathbf{p}' \rangle) \wedge R_\beta(\langle \mathbf{h}'', \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle))$$

Правило коммутации пакетов $r = \langle \mathbf{z}, \mathbf{y}, \alpha \rangle$ применяет инструкцию α ко всем тем пакетам, локальные состояния которых подпадают под соответствующие шаблоны заголовка \mathbf{y} и порта \mathbf{z} .

Семантика правила определяется бинарным отношением R_r на множестве локальных состояний пакетов $\mathcal{H} \times \mathcal{P}$:

$$R_r(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = precondition_r(\langle \mathbf{h}, \mathbf{p} \rangle) \wedge R_\alpha(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle)$$

$$precondition_r(\langle \mathbf{h}, \mathbf{p} \rangle) = \bigwedge_{i=1}^N E_{\sigma_i}(\mathbf{h}[i]) \wedge \bigwedge_{i=0}^k E_{\sigma_i}(\mathbf{p}[i])$$

Предикат $precondition_r$ называется *предохранителем* правила r .

Таблицей коммутации называется пара $\langle D, \beta \rangle$, где $D = \{r_1, r_2, \dots, r_n\}$ – множество правил коммутации, а β – *инструкция по умолчанию*. Коммутатор применяет правила таблицы ко всем пакетам, поступающим на его входные порты. Если ни одно из правил множества D нельзя применить к пакету, то этот пакет обрабатывается инструкцией β .

Семантика таблицы коммутации tab задается бинарным отношением:

$$R_{tab}(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) = \bigvee_{i=1}^n R_{r_i}(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) \vee \left(\neg \left(\bigwedge_{i=1}^n precondition_{r_i}(\langle \mathbf{h}, \mathbf{p} \rangle) \right) \wedge R_\beta(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) \right)$$

Для обозначения всевозможных таблиц коммутации для заданной сети используется запись Tab . Показано, что топология сети полностью определяется отношением пересылки пакетов $T \subseteq (OP \times \mathcal{W}) \times (IP \times \mathcal{W})$. Точки, вовлеченные в отношение T , называются внутренними точками сети; все прочие точки сети считаются её внешними точками. Для обозначения множеств внешних входных и внешних выходных точек сети используются записи In и Out .

Предполагается, что внешние точки подключены к внешним устройствам (контроллерам, серверам, сетевым шлюзам и так далее), которые находятся вне сферы влияния контроллера ПКС.

Для заданного множества коммутаторов W и топологии T *конфигурацией сети* называется всюду определенная функция $Net: W \rightarrow Tab$, ассоциирующая с каждым коммутатором сети некоторую таблицу коммутации пакетов. Семантика сети с заданной конфигурацией Net определяется отношением на множестве состояний пакетов $\mathcal{S} = \mathcal{H} \times \mathcal{P} \times \mathcal{W}$:

$$R_{Net}(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle) = (C_{Net}(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle) \wedge Out(\langle \mathbf{p}', \mathbf{w}' \rangle)) \vee \exists \mathbf{p}'' (C_{Net}(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}', \mathbf{p}'', \mathbf{w}' \rangle) \wedge T(\langle \mathbf{p}'', \mathbf{w} \rangle, \langle \mathbf{p}', \mathbf{w}' \rangle))$$

$$C_{Net}(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle) = \bigvee_{w \in W} R_{Net(w)}(\langle \mathbf{h}, \mathbf{p} \rangle, \langle \mathbf{h}', \mathbf{p}' \rangle) \wedge \bigwedge_{i=1}^m (\mathbf{w}[i] = \mathbf{w}'[i])$$

Если отношение $R_{Net}(\mathbf{s}, \mathbf{s}')$ выполняется для пары состояний пакетов $\mathbf{s} = \langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle$ и $\mathbf{s}' = \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle$, то пакет с заголовком \mathbf{h} , поступающий на порт \mathbf{p} коммутатора \mathbf{w} , может быть за один скачок (hop) отправлен либо на входной порт \mathbf{p}' коммутатора \mathbf{w}' , либо на устройство, подключенное к внешнему выходному порту \mathbf{p}' коммутатора \mathbf{w} .

Семантика статической модели ПКС Net определяется отношением одношаговой маршрутизации пакетов $R_{Net}(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle)$, которое составляет реализуемые сетью правила коммутации пакетов, а также свойствами $Out(\langle \mathbf{p}, \mathbf{w} \rangle)$ и $Out(\langle \mathbf{p}, \mathbf{w} \rangle)$, описывающими множество внешних точек сети. Располагая этими отношениями можно проследить маршруты всех потоков, проходящих через сеть при заданной ее конфигурации.

Динамическая модель ПКС предназначена для описания поведения контроллера. В этой модели контроллер представлен в виде автомата, получающего сообщения от коммутаторов на входе и генерирующего последовательности команд для их реконфигурации на выходе. Множество допустимых сообщений в модели контроллера ограничено лишь заголовками тех пакетов, которые могут достичь контроллера, будучи отправленными из какой-либо внешней точки сети.

В качестве формальной модели контроллера ПКС построен автомат $A = (\mathcal{S}, \mathcal{C}, Q, q_0, \Delta)$, где

- \mathcal{S} и \mathcal{C} – входной и выходной алфавиты соответственно,
- Q – множество внутренних состояний управления контроллера,
- $q_0, q_0 \subseteq Q$ – начальное состояние управления, и
- $\Delta, \Delta \subseteq Q \times \mathcal{S} \times \mathcal{C}^* \times Q$ – отношение переходов.

Для каждой конфигурации сети Net контроллер A может получать только такие сообщения, которые были индуцированы пакетами, поступившими на внешние точки сети. Рефлексивно-транзитивное замыкание R_{Net}^* отношения одношаговой коммутации R_{Net} позволяет определить множество сообщений $Event(Net)$, допустимых в конфигурации Net следующим образом:

$$Event(Net) = \{ \langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle : Visible(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle) \}$$

$$Visible(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle) = \exists \mathbf{h}', \mathbf{p}', \mathbf{w}', \mathbf{h}'' :$$

$$\left(In(\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle) \wedge R_{Net}^* (\langle \mathbf{h}' , \mathbf{p}' , \mathbf{w}' \rangle, \langle \mathbf{h}' ' , \mathbf{p}, \mathbf{w} \rangle) \wedge \right. \\ \left. R_{Net} (\langle \mathbf{h}' ' , \mathbf{p}, \mathbf{w} \rangle, \langle \mathbf{h}, octr, \mathbf{w} \rangle) \right)$$

Динамическая модель ПКС полностью определяется множествами $\mathcal{W}, \mathcal{P}, \mathcal{H}$ коммутаторов, портов и заголовков пакетов, отношением пересылки пакетов T и контроллером A .

Частичным прогоном ПКС $M = (W, P, H, T, A)$ называется (конечная или бесконечная) последовательность вида:

$$run = (Net_0, q_0) \xrightarrow{s_1} (Net_1, q_1) \xrightarrow{s_2} \dots \xrightarrow{s_i} (Net_i, q_i) \xrightarrow{s_{i+1}} (Net_{i+1}, q_{i+1}) \xrightarrow{s_{i+2}} \dots$$

где для каждого i , $0 \leq i$:

- Net_i – конфигурация ПКС, q_i и s_i – состояния контроллера и пакета;
- $s_{i+1} \in Event(Net_i)$;
- отношение переходов контроллера A содержит четверку $(q_i, s_{i+1}, \omega_i, q_{i+1})$, для которой конфигурация Net_{i+1} получается из Net_i путём применения к ней последовательности команд обновления конфигурации ω_i .

Полный прогон ПКС – это частичный прогон, который либо является бесконечным, либо завершается в состоянии ПКС (Net_i, q_i) , для которого выполнено соотношение $Event(Net_i) = \emptyset$. Для заданной ПКС M и конфигурации сети Net_0 запись $Run(M, Net_0)$ обозначает множество полных прогонов M , начинающихся парой (Net_0, q_0) .

Для спецификации политик маршрутизации построена 3-х уровневая иерархия языков, в которой выражения языка нижележащего уровня выступают в роли атомарных высказываний для языка вышележащего уровня:

Язык первого уровня \mathcal{L}_0 предназначен для описания свойств простейших элементов сети – отдельных состояний пакетов и их множеств – например, шаблонов правил коммутации. Поскольку семантика этих компонентов сети определяется двоичными векторами, математическую основу этого языка составляет булева алгебра.

Рассмотрим множество переменных $Var = \{X_1, X_2, \dots\}$, принимающих значения из множества состояний пакетов $\mathcal{S} = \mathcal{H} \times \mathcal{P} \times \mathcal{W} = \{0, 1\}^{N+k+m}$. Спецификацией состояния пакета называется всякая булева формула ϕ над множеством булевых переменных $\{X_i[j] : X_i \in \{Var\}, 1 \leq j \leq N + k + m\}$ и связок \neg, \wedge . Множество всех таких формул обозначается записью \mathcal{L}_0 . Формулы из \mathcal{L}_0 позволяют выразить любую булеву функцию на множестве

состояний пакетов или, что то же самое, выделить любое подмножество указанных состояний. Язык \mathcal{L}_0 был использован при задании отношений, специфицирующих поведение коммутирующих элементов сети.

Язык второго уровня \mathcal{L}_1 описывает глобальные статические требования, значительная часть которых представляет собой ограничения на отношения коммутации и маршрутизации. Математической основой языка \mathcal{L}_1 является логика первого порядка, расширенная оператором транзитивного замыкания.

В языке \mathcal{L}_1 используются единственный двухместный предикатный символ R и два одноместных предикатных символа I и O ; это наименьший язык, удовлетворяющий следующим требованиям:

- если $\phi \in \mathcal{L}_0$, то $\phi \in \mathcal{L}_1$;
- если $X, Y \in Var$, то атомарные выражения $R(X, Y)$, $I(X)$, $O(Y)$ принадлежат \mathcal{L}_1 ;
- если ψ_1 и ψ_2 – формулы из \mathcal{L}_1 , и $X \in Var$, то формулы $(\neg \psi_1)$, $(\psi_1 \wedge \psi_2)$ и $(\exists X \psi_1)$ так же принадлежат \mathcal{L}_1 ;
- если $\psi(X, Y)$ – формула из \mathcal{L}_1 , содержащая не более двух свободных переменных, то $TC(\psi(X, Y)) \in \mathcal{L}_1$.

Семантика языка \mathcal{L}_1 определяется с помощью отношений из статической модели сети. Пусть Net – некоторая конфигурация сети, и $\mathbf{s} = \langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle$ и $\mathbf{s}' = \langle \mathbf{h}', \mathbf{p}', \mathbf{w}' \rangle$ – пара состояний пакетов. Тогда

$Net \models I(X)[\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle]$ тогда и только тогда, когда $\langle \mathbf{p}, \mathbf{w} \rangle \in In$;

$Net \models O(X)[\langle \mathbf{h}, \mathbf{p}, \mathbf{w} \rangle]$ тогда и только тогда, когда $\langle \mathbf{p}, \mathbf{w} \rangle \in Out$;

$Net \models R(X, Y)[\mathbf{s}, \mathbf{s}']$ тогда и только тогда, когда $(\mathbf{s}, \mathbf{s}') \in R_{Net}$;

$Net \models TC(\psi(X, Y))[\mathbf{s}, \mathbf{s}']$ тогда и только тогда, когда существует конечная последовательность состояний пакетов $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_l$, где $0 < l$, что $\mathbf{s}_0 = \mathbf{s}$ и $\mathbf{s}_l = \mathbf{s}'$, и для каждого i , $1 \leq i \leq l$ выполнено условие $Net \models \psi(X, Y)[\mathbf{s}_{i-1}, \mathbf{s}_i]$.

Отношение выполнимости для прочих формул \mathcal{L}_1 определяется стандартным образом.

Поскольку язык \mathcal{L}_1 построен на базе логики транзитивного замыкания $FO[TC]$, то он обладает большей выразительной мощностью, чем темпоральные логики LTL, CTL, CTL* и ω -исчисления. Что касается алгоритмической сложности его использования – в работе доказана следующая

Теорема 1. Задача верификации модели сети $Net \models \psi$ относительно замкнутых формул языка \mathcal{L}_1 является PSPACE-полной.

Язык третьего уровня \mathcal{L}_2 предназначен для описания динамических требований к поведению ПКС. Эти требования описывают желаемую реакцию контроллера на сообщения, поступающие от коммутаторов. В качестве математической основы языка \mathcal{L}_2 предлагается использовать фрагмент языка темпоральной логики линейного времени FO[TC], атомарными высказываниями которого выступают замкнутые формулы языка \mathcal{L}_1 .

Синтаксис языка \mathcal{L}_2 определяется следующим образом. Пусть $\mathcal{L}_0(X)$ и $\mathcal{L}_1(X)$ – множества всех тех формул языков \mathcal{L}_0 и \mathcal{L}_1 соответственно, в которых единственная свободная переменная X . Тогда $LTL(\mathcal{L}_1)$ – это наименьший язык, удовлетворяющий следующим требованиям:

- если $\psi(X) \in \mathcal{L}_1(X)$, то $\psi(X) \in LTL(\mathcal{L}_1)$;
- если $\Phi(X), \Psi(X) \in LTL(\mathcal{L}_1)$, то формулы $(\neg\Phi(X))$, $(\Phi(X) \wedge \Psi(X))$, $(\mathbf{X} \Phi(X))$ и $(\Phi(X) \mathbf{U} \Psi(X))$ принадлежат $LTL(\mathcal{L}_1)$.

Оценка истинности формул из $LTL(\mathcal{L}_1)$ производится на бесконечных последовательностях сетевых конфигураций $\{Net_i\}_{i=1}^{\infty}$ для заданного состояния пакета \mathbf{s} :

- если $\psi(X) \in \mathcal{L}_1(X)$, то $\{Net_i\}_{i=1}^{\infty} \models \psi(\mathbf{s})$ тогда и только тогда, когда $Net_1 \models \psi(\mathbf{s})$,
- $\{Net_i\}_{i=1}^{\infty} \models \mathbf{X} \Phi(\mathbf{s})$ тогда и только тогда, когда $\{Net_i\}_{i=2}^{\infty} \models \Phi(\mathbf{s})$,
- $\{Net_i\}_{i=1}^{\infty} \models \Phi(\mathbf{s}) \mathbf{U} \Psi(\mathbf{s})$ тогда и только тогда, когда $\{Net_i\}_{i=k}^{\infty} \models \Psi(\mathbf{s})$ для некоторого $k, 1 \leq k$, и $Net_j \models \Phi(\mathbf{s})$ для каждого $j, 1 \leq j < k$,
- семантика связок \neg и \wedge определяется обычным образом.

Язык спецификации политик маршрутизации \mathcal{L}_2 состоит из множества выражений вида $\phi(X) \Rightarrow \Phi(X)$, где $\phi(X) \in \mathcal{L}_0(X)$, а $\Phi(X)$ – темпоральная формула языка $LTL(\mathcal{L}_1)$. Семантика этих выражений определяется посредством отношения выполнимости на прогонах формальных моделей ПКС.

Для заданного прогона run модели ПКС M и выражения $\phi(X) \Rightarrow \Phi(X)$ языка \mathcal{L}_2 отношение $run \models \phi(X) \Rightarrow \Phi(X)$ имеет место тогда и только тогда, когда $\{Net_i\}_{i=n}^{\infty} \models \Phi(\mathbf{s}_n)$ для каждого $n, 1 \leq n$, удовлетворяющего условию $\phi(\mathbf{s}_n) = 1$.

Политика маршрутизации PFP задаётся ограничением ψ на множество допустимых начальных конфигураций сети, представляющим собой замкнутую формулу языка \mathcal{L}_1 , и конечным множеством $\{\phi_1(X) \Rightarrow \Phi_1(X), \dots, \phi_n(X) \Rightarrow \Phi_n(X)\}$ выражений языка \mathcal{L}_2 . Считается, что ПКС M реализует политику

маршрутизации PFP в том случае, когда для любой сетевой конфигурации Net_0 , удовлетворяющей условию $Net_0 \models \psi$, каждый прогон run из множества $Run(M, Net_0)$ удовлетворяет соотношению $\phi_i(X) \Rightarrow \Phi_i(X)$, $1 \leq i \leq n$. Таким образом, задача верификации моделей ПКС состоит в том, чтобы проверить для заданной формальной модели ПКС M , удовлетворяет ли она спецификации политики коммутации $\$PFP\$$.

Для случая, когда контроллер ПКС представляет собой конечный автомат, сформулированная задача верификации разрешима. Показано, что указанная задача сводится к хорошо известной проблеме верификации конечных моделей программ для темпоральной логики LTL . На основе проведенных теоретических исследований было построено программное средство верификации сетевых конфигураций относительно их спецификаций на языке \mathcal{L}_1 , состоящее из трех компонентов:

1. Конструктор моделей строит множество двоичных решающих диаграмм OBDD; описывающих конфигурацию ПКС.
2. Анализатор спецификаций предназначен для разбора выражений языка спецификации политик маршрутизации \mathcal{L}_1 ;
3. Верификатор ПКС позволяет убедиться в том, что свойства заданной конфигурации ПКС удовлетворяют множеству предъявляемых к ней требований, или, наоборот, обнаружить факт их нарушения.

Наиболее интересной частью разработанной программной реализации является применение аппарата двоичных диаграмм OBDD для компактного и эффективного представления модели. В частности, в разделе исследован вопрос выбора наилучшего порядка двоичных переменных и проведения над диаграммами некоторых специфических для поставленной задачи операций. Дано строгое описание грамматики разработанного языка \mathcal{L}_1 и примеры спецификаций для некоторых наиболее распространённых политик маршрутизации.

В работе показано, что задача проверки соответствия конфигурации ПКС заданным требованиям политик маршрутизации сводится к задаче проверки выполнимости описывающих эти требования формул языка \mathcal{L}_1 на множестве отношений, моделирующих ПКС. Этот факт был использован при разработке верификатора: он преобразует значения отношений, полученных на этапе

анализа конфигурации сети, в деревья вычислений, полученные от анализатора спецификаций, и производит поочерёдное их вычисление:

- Если хотя бы одна формула выдаёт ложное значение, конфигурация сети нарушает требования политик маршрутизации и считается некорректной;
- Иначе, если все формулы выполняются – конфигурация корректна.

В конце этой главы представлены методики проведения и результаты экспериментальных исследований построенных средств. Результаты проведённых экспериментов показали, что они могут успешно применяться на практике. Кроме того, построенные средства позволили обнаружить в конфигурации сети ошибки, которые не удалось выявить другими методами.

Во **второй главе** рассматривается задача получения верхней оценки сквозной задержки передачи данных в сети. Показано что данная задача относится к проблеме управления качеством сервиса в сети.

Приводится краткое описание основных методов управления качеством сервиса, с которым обслуживаются передающиеся через сеть потоки данных, а так же соответствующих им моделям распределения ресурсов. В частности, в этом разделе рассматриваются *модель интегрированных сервисов* и *модель дифференцированных сервисов*.

В модели интегрированных сервисов резервирование ресурсов происходит в процессе установки соединения и не может изменяться до момента его завершения. Поэтому для обслуживания каждого соединения выделяется такое количество ресурсов, которое было бы достаточным для удовлетворения требований использующего это соединение приложения на пике нагрузки. Такой подход чреват фрагментацией ресурсов: сеть не справляется с нагрузкой, не смотря на то, что часть её ресурсов не используется.

Основная идея модели дифференцированных ресурсов – введение приоритетов на множестве соединений. Коммутаторы заранее определяют несколько классов обслуживания, а передающиеся через сеть соединения выбирают наиболее подходящие из предложенных классов. Объём сетевых ресурсов, который выделяется для обслуживания каждого конкретного класса обслуживания, не фиксирован. Гарантируется лишь то, что более высоким классам обслуживания будет предоставлено больше ресурсов, чем менее приоритетным. Указанные особенности модели дифференцированных сервисов позволяют избежать простаивания оборудования, что выгодно отличает её от

модели интегрированных ресурсов. Поэтому в качестве наиболее актуальной модели распределения ресурсов была выбрана модель дифференцированных сервисов.

Далее в этой главе приводится описание внутреннего устройства коммутатора, а так же рассматриваются способы реализации моделей интегрированных и дифференцированных сервисов в коммутационном оборудовании, анализируются разные методы буферизации: буферизация на выходе (OQ), буферизация на входе (IQ), виртуальная буферизация на выходе (VOQ), множественная виртуализация на выходе (MOQ) и комбинированная буферизация (CIOQ).

Для постановки задачи получения верхней оценки сквозной задержки и построения модели передачи пакета через сеть было выбрано *сетевое исчисление (network calculus)*¹. Сетевое исчисление изучают зависимости между производительностью элементов сети и качеством сервиса, с которым обрабатываются передаваемые через эту сеть потоки данных. В основе теории лежит идея получения детерминированных оценок путём рассмотрения наилучших и наихудших, с точки зрения заданной характеристики, сценариев работы отдельных частей сети и эффективного комбинирования этих сценариев между собой.

Модель ТСИ описывает коммутационную сеть следующим образом:

- Сеть разбивается на множество *обработчиков*. Разные обработчики могут моделировать элементы сети различного масштаба, начиная от физической линии или очереди пакетов внутри сетевого устройства и заканчивая коммутаторами и даже целыми подсетями;
- Поведение обработчика определяется *кривой сервиса*, которая отражает его производительность и используемые им правила распределения ресурсов между проходящими через него потоками;
- Для каждого потока данных задаётся маршрут их передачи через обработчики и *кривая нагрузки*, которая описывает характерные для этого потока ограничения на скорость поступления данных в сеть.

¹ J.-Y. Le Boudec, P. Thiran. Network Calculus: a theory of deterministic queuing systems for the internet // Lecture Notes in Computer Science (LNCS), volume 2050, Berlin, Germany: Springer Verlag, 2001.

В качестве кривых сервиса и кривых нагрузки используется класс \mathcal{F} накопительных функции времени, которые выражают общее количество данных, переданных этими модельными сущностями с момента начала отсчёта.

$$\mathcal{F} \stackrel{\text{def}}{=} \{f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\} \mid \text{LeftContinuous}(f) \wedge \text{NonDecreasing}(f) \wedge \text{Causal}(f)\}$$

$$\text{NonDecreasing}(f) \stackrel{\text{def}}{=} \forall t_1 \leq t_2: f(t_1) \leq f(t_2)$$

$$\text{LeftContinuous}(f) \stackrel{\text{def}}{=} \forall t_0 \in \mathbb{R}: \lim_{t \rightarrow t_0 - 0} f(t) = f(t_0)$$

$$\text{Causal}(f) \stackrel{\text{def}}{=} \forall t = 0 \rightarrow f(t) = 0$$

Функцией *прибытия (отправки)* потока данных для обработчика S называется такая накопительная функция $A \in \mathcal{F}$ ($D \in \mathcal{F}$), которая описывает зависимость суммарного количества данных потока, поступивших на этот обработчик (переданных этим обработчиком), от времени.

Периодом отставания называется временной интервал, в течение которого отставание $b(t)$ обработчика строго положительно. Если момент $t \in \mathbb{R}$ находится внутри периода отставания $[\text{SBP}(t); \text{EBP}(t)]$, то его границы задаются формулами:

$$\text{SBP}(t) = \sup\{u \leq t \mid A(u) = D(u)\}$$

$$\text{EBP}(t) = \inf\{u \geq t \mid A(u) = D(u)\}$$

После получения данных обработчики способны временно удерживать их внутри себя. Зависимость объёма хранящихся в обработчике S данных от времени называется его *отставанием* $b(t)$ и представляется в виде разности функций прибытия A и отправки D обслуживаемого им потока данных:

$$b(t) = A(t) - D(t)$$

Задержкой $d(t)$ обработчика S называется время, которое порция данных, поступившая в S в момент времени t , пробудет внутри этого обработчика.

Для произвольного обработчика S и любой пары соответствующих этому обработчику функций прибытия A и отправки D , $\langle A, D \rangle \in S$, его отставание $b(t)$ ограничено сверху вертикальным отклонением $v(A, D)$ этих потоков. Более того, если обработчик S обслуживает данные по дисциплине FIFO, его задержка $d(t)$ ограничена сверху их горизонтальным отклонением $h(A, D)$:

$$\forall t \in \mathbb{R}: b(t) \leq v(A, D) = \sup_t \{A(t) - D(t)\}$$

$$\forall t \in \mathbb{R}: d(t) \leq h(A, D) = \sup_t \{ \inf \{ \tau \geq 0 | A(t) \leq D(t + \tau) \} \}$$

Каждый обработчик S может быть описан перечислением соответствующих ему пар вида $\langle A, D \rangle$ и задан отношением вида $S \subset \mathcal{F} \times \mathcal{F}$.

На практике для описания обработчика используют кривые сервиса: (нестрогой) *кривой сервиса* называется такая функция $\beta \in \mathcal{F}$, что при обслуживании потока с функцией прибытия $A \in \mathcal{F}$ для его функции отправки $D \in \mathcal{F}$ выполняется:

$$D(t) \geq \inf_{s \leq t} \{ A(s) + \beta(t - s) \}$$

Строгой кривой сервиса обработчика S называется всякая такая функция $\beta_s \in \mathcal{F}$, что в процессе обслуживания потока с функцией прибытия A и функцией отправки D , $\langle A, D \rangle \in S$, в течение каждого периода отставания $[s; t]$ элемент S обрабатывает, по меньшей мере, $\beta_s(t - s)$ данных:

$$D(t) - D(s) \geq \beta_s(t - s)$$

Как и в случае функции производительности P обработчика, построение сколь-нибудь точной функции прибытия $A \in \mathcal{F}$ для заданного потока данных на практике может вызвать затруднения. Поэтому для описания потоков поступления данных используется тот же приём, что и для описания обработчиков: явная зависимость функции поступления A от времени заменяются зависимостью от длительности интервала передачи данных.

Кривой нагрузки для потока с функцией прибытия $A \in \mathcal{F}$ называется такая функция $\alpha \in \mathcal{F}$, что для каждого временного интервала длины τ количество переданных в течение него данных этого потока не превышает величины $\alpha(\tau)$:

$$\forall t: \forall \tau: A(t + \tau) - A(t) \leq \alpha(\tau)$$

Теория сетевого исчисления определяет над множеством накопительных функций операции поточечного сложения и минимума. Построенная с их помощью Min-Plus алгебра используется для выражения зависимостей между потоками данных, поступающими на входы обработчиков, и потоками данных, передаваемыми через их выходы. При этом используются операции прямой и обратной Min-Plus свёртки, обозначенные символами \otimes и \oslash соответственно.

$$(f \otimes g)(t) \stackrel{\text{def}}{=} \inf_{0 \leq s \leq t} \{ f(s) + g(t - s) \}$$

$$(w \oslash g)(t) \stackrel{\text{def}}{=} \sup_{t \geq u \geq 0} \{ w(t + u) - g(u) \}$$

Указанные операции позволяют компактно сформулировать основные результаты теории сетевого исчисления: Если обработчик с кривой сервиса $\beta \in \mathcal{F}$ обслуживает поток данных, ограниченный кривой нагрузки $\alpha \in \mathcal{F}$, то:

1. Для отставания $b(t)$ выполнена формула:

$$b(t) \leq (\alpha \oslash \beta)(0)$$

2. При дисциплине FIFO для задержки выполняется неравенство:

$$d(t) \leq \inf\{\tau \geq 0 | (\alpha \oslash \beta)(-\tau) \leq 0\}$$

3. Поток обработанных данных ограничен кривой прибытия $\alpha'(t)$:

$$\alpha'(t) = (\alpha \oslash \beta)(t)$$

Одна из основных сложностей применения теории сетевого исчисления на практике обусловлена высокой сложностью вычисления прямой и обратной свёртки. Однако известны быстрые алгоритмы для вычисления этих операций при рассмотрении более узких классов функций. Например, в литературе часто рассматриваются кривые нагрузки и кривые сервиса образованные парой линейных сегментов. Одним из известных результатов является следующая

Теорема. График функции свёртки для двух выпуклых кусочно-линейных функции $f \in \mathcal{F}$ и $g \in \mathcal{F}$ может быть построен путём последовательной композиции составляющих эти функции отрезков в порядке увеличения значений их угловых коэффициентов.

Одним из результатов настоящей работы является изложенное в разделе доказательство, в некоторой степени, обратного для неё утверждения:

Теорема 2. Пусть задана пара кусочно-линейных функций: вогнутая функция $w \in \mathcal{F}$ и выпуклая функция $g \in \mathcal{F}$ – пересекающихся ровно в двух точках: 0 и $t_0 = \sup\{w(t) = g(t)\} > 0$. Тогда график их обратной свёртки $f = w \oslash g$ может быть построен из точки $\langle t_0, -g(t_0) \rangle$ последовательным соединением отрезков, составляющих графики функций $w(t)$ и $\hat{g}(t) = \min(g(t), g(t_0))$, в порядке уменьшения их угловых коэффициентов.

Эта теорема позволила обобщить два существующих алгоритма оценки времени передачи пакетов на более широкий класс функций описания кривых нагрузки и сервиса. Ранее эти алгоритмы предполагали, что кривые нагрузки и сервиса – кусочно-линейные функции из двух сегментов. Благодаря доказанной теореме те же алгоритмы можно применять и в том случае, когда указанные функций состоят из произвольного количества сегментов.

Показано, что оба указанных алгоритма не в состоянии предоставить достижимые оценки для времени передачи пакетов и дают лишь грубые оценки сверху. Одним из путей получения более точных оценок является отказ от использования свёртки, и описание всей сети в виде системы неравенств, отражающей зависимость объёма переданных через сеть данных от количества данных, поступивших на её входы, во времени. Далее в работе приводится описание способа использования систем линейных неравенств для вычисления достижимой верхней оценки времени передачи пакетов через сеть.

Топология сети обработчиков задана ориентированным ациклическим графом $G = \langle V, E \rangle$ без кратных дуг, вершины которого пронумерованы натуральными числами $V = \{1, \dots, m\}$. Вершине с номером v сопоставлен обработчик S_v , который обслуживает совокупность проходящих через него потоков данных в соответствии с вогнутой кусочно-линейной строгой кривой сервиса $\beta_v \in \mathcal{F}$. Обработчики обслуживают каждый отдельный поток по дисциплине FIFO, однако правила мультиплексирования этих потоков между собой, вообще говоря, произвольные – модель построена в ограничениях *слепого мультиплексирования*.

Каждый путь π длины $l = |\pi|$ через граф G однозначно описывается неповторяющейся последовательностью $\pi = \{v_k\}_{k=1}^l$ вершин. Для обозначения i -й вершины маршрута π используется запись $\pi[k]$. При этом вне зависимости от маршрута π значение выражения $\pi[0] \equiv 0$.

Через граф G пролегают маршруты π_1, \dots, π_n передачи данных множества пронумерованных натуральными числами потоков $F = \{F_1, \dots, F_n\}$. При этом потоку с номером i сопоставлена собственная вогнутая кусочно-линейная кривая нагрузки $\alpha_i \in \mathcal{F}$.

Кривые нагрузки потоков и кривые сервиса обработчиков согласованы между собой так, что суммарный объём данных, пребывающих на каждый из обработчиков сети, асимптотически не превышает того объёма, который этот обработчик в состоянии обслужить:

$$\forall v \in V, t \rightarrow \infty: \sum_{i: v \in \pi_i} \alpha_i(t) \leq \beta_v(t)$$

Функция поступления данных потока $F_i \in \mathcal{F}$ в сеть обозначается записью F_i^0 . Для обозначения функции передачи данных, которая для каждого момента

времени $t \geq 0$ возвращает суммарное количество данных потока F_i , сумевших преодолеть обработчик с номером v за время $[0; t]$, используется запись F_i^v .

Таким образом, интенсивность передачи данных потока F_i через сеть характеризуется множеством функций $\bar{F}_i = \{F_i^{\pi_i[k]} \mid 0 \leq k \leq |\pi_i|\}$.

Сценарием работы сети называется всякое множество $\cup \bar{F}_i$ из $\sum |\pi_i| + n$ функций поступления данных, удовлетворяющее следующим ограничениям:

- Каждая из указанных функций должна принадлежать классу \mathcal{F} ;
- В каждый момент времени объём данных, преодолевших произвольный обработчик сети, не превышает объёма данных, которые на него поступили: $\forall F_i, t \geq 0, 1 \leq k \leq |\pi_i|: F_i^{\pi_i[k-1]}(t) \geq F_i^{\pi_i[k]}(t)$
- Функции, непосредственно описывающие поступление потоков в сеть, удовлетворяют ограничениям соответствующих кривых нагрузки;
- Функции, проходящие через один и тот же обработчик, удовлетворяют соответствующим ему кривым сервиса;

Поскольку перечисленные ограничения определяют функции поступления данных неоднозначно, то одна и та же сеть способна обслуживать проходящие через неё потоки по разным сценариям. Таким образом, для построения достижимой верхней оценки задержки передачи заданного потока данных через сеть достаточно рассмотреть множество допустимых сценариев её работы и выбрать среди них один из тех сценариев, при котором некоторая порция данных этого потока передаётся наибольшее время.

Предлагается описывать семейство допустимых сценариев работы сети системой линейных неравенств. В качестве переменных в построенных линейных системах используются значения отдельных функций сценария в некоторые моменты времени. Сами неравенства получаются из ограничений сценария работы сети. Преимущество указанного подхода заключается в том, что выбор такого сценария работы сети, на котором достигается максимальная задержка передачи данных заданного потока, может быть проведён с помощью линейного программирования.

Так как не все ограничения на сценарий имеют линейную природу, то вычисление верхней оценки для времени передачи данных через сеть сводится к решению нескольких систем линейных неравенств, каждая из которых соответствует некоторому семейству допустимых сценариев работы сети. Наибольшее среди их решений является искомой верхней оценкой.

Хотя использование систем линейных уравнений для моделирования сети обработчиков в предположениях задачи и не является новым, в работе предложен оригинальный алгоритм разбиения всех допустимых сценариев работы сети на семейства, а так же генерации соответствующих им неравенств. Указанный алгоритм является конструктивным доказательством для следующей теоремы:

Теорема 3. Если топология сети обработчиков может быть представлена неориентированным ациклическим графом, то задача построения достижимой верхней оценки передачи данных по маршруту из n обработчиков сводится к решению $n-1$ системы линейных неравенств и может быть решена за полиномиальное время.

Ранее полиномиальное время решения задачи построения верхней оценки для задержки было получено лишь для сетей с линейной (tandem) топологией.

В конце этой главы приводится описание разработанных программных средств построения верхних оценок для сквозной задержки передачи данных, а так же результаты экспериментальных апробаций указанных средств.

Полученные с помощью разработанных алгоритмов оценки времени передачи данных сравниваются с показателями специально разработанной имитационной модели сети. На основе экспериментальных данных показано, что время передачи пакетов в имитационной модели сети никогда не превосходят полученных теоретически верхних оценок.

В **заключении** сформулированы основные результаты диссертационной работы и перечислены некоторые из возможных путей их развития.

В **Приложении А** приведено описание синтаксиса и семантики некоторых перспективных формальных языков для спецификации политик маршрутизации.

В **Приложении Б** даны описания некоторых перспективных формальных моделей ПКС.

В **приложении В** содержатся доказательства некоторых вспомогательных теорем сетевого исчисления.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Полученные автором результаты изложены в следующих работах:

1. Chemeritskiy E., Smelyansky R., Zakharov V. A formal model and verification problems for software defined networks // Proceedings of the 4-th International

- Workshop "Program Semantics, Specification and Verification: Theory and Applications". — Yekaterinburg, Russia, 2013. — P. 21–30.
2. Захаров В., Смелянский Р., Чемерицкий Е. Формальная модель и задачи верификации программно-конфигурируемых сетей // Моделирование и анализ информационных систем. — 2013. — Т. 20, № 6. — С. 33–48.
 3. Chemeritskiy E., Zakharov V. On the network update problem for software defined networks // Proceedings of the 5th Workshop "Program Semantics, Specification and Verification: Theory and Applications". — Издательский дом ГУ ВШЭ Москва, 2014. — P. 26–37.
 4. V. Altukhov, E. Chemeritskiy, V. Podymov, V. Zakharov Vermont - a toolset for checking SDN packet forwarding policies on-line // Proceedings of the 2014 international science and technology conference "Modern Networking Technologies (MoNeTec)". — SDN & NFV modern networking technologies. — МАКС Пресс Москва, Россия, 2014. — P. 7–12.
 5. Altukhov V., Chemeritskiy E. On real-time delay monitoring in software-defined networks // Proceedings of the 2014 international science and technology conference "Modern Networking Technologies (MoNeTec)". — SDN & NFV modern networking technologies. — Макс Пресс Москва, Россия, 2014. — P. 1–6.
 6. Smeliansky R., Chemeritsky E. On QoS management in SDN by multipath routing // Proceedings of the 2014 international science and technology conference "Modern Networking Technologies (MoNeTec)". — SDN & NFV modern networking technologies. — МАКС Пресс Москва, Россия, 2014. — P. 41–46.
 7. Chemeritskiy E., Zakharov V. Consistent network update without tagging // Proceedings of the 2014 international science and technology conference "Modern Networking Technologies (MoNeTec)". — SDN & NFV modern networking technologies — МАКС Пресс Москва, Россия, 2014. — P. 47–52.
 8. V. Altukhov, E. Chemeritskiy, V. Podymov, V. Zakharov A runtime verification system for software defined networks // Материалы Международной научно-практической конференции: Tools & Methods of Program Analysis, ТМРА-2014. — Кострома: КГТУ, 2014. — P. 19–28.
 9. Захаров В., Чемерицкий Е. О некоторых задачах реконфигурирования программно-конфигурируемых сетей // Моделирование и анализ информационных систем. — 2014. — Т. 21, № 6. — С. 57–69.

10. Chemeritsky E., Smeliansky R., Zakharov V. A formal model and verification problems for software defined networks // Automatic Control and Computer Sciences. — 2014. — Vol. 48, no. 7. — P. 398–406.
11. В. Алтухов, В. Захаров, В. Подымов, Е. Чемерицкий Vermont - средство верификации программно-конфигурируемых сетей / // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — 2015. — Т. 212, № 1. — С. 74–87.