УДК 004.725

ПРОВЕДЕНИЕ ИССЛЕДОВАНИЙ В ОБЛАСТИ ТРАФИКОВ IP-СЕТЕЙ НА БАЗЕ ПЛИС-ПЛАТФОРМЫ

А. С. Коваль

Воронежский государственный университет

Поступила в редакцию 31.10.2014 г.

Аннотация. Приводится опыт использования ПЛИС-платформы для проведения исследований в области трафиков IP-сетей, методики и варианты проведения пассивных и активных (с модификацией способов обработки трафика) исследований. Приведены оценки производительности исследо-вательской платформы.
Ключевые слова: IP-сети, трафик сетей, ПЛИС.
Annotation. IWe present experience of the use of FPGA-based platform for research in the field of IP-networks traffic, methods and options for passive and active (with modification of traffic processing) research. The research platform's performance was estimated and presented.
Keywords: IP networks, FPGA, network traffic.

ВВЕДЕНИЕ

Исследования трафиков реальных IP-сетей требует доступа к внутренним компонентам сетевого оборудования, как для получения текущих характеристик трафика, так и для управления основными элементами сетевого оборудования на уровне пересылки данных (data plane) и уровне управления (control plane). Поскольку управление во многих случаях требует режима реального времени, а потоки данных должны протекать со скоростью заданной интерфейсами оборудования (line-rate), требуются аппаратные решения. Необходимую скорость и возможность реконфигурации и управления на низком уровне можно получить на платформе, содержащей ПЛИС и сетевые интерфейсы с реализацией MAC-уровня как в виде специализированных ИС, так и внутри самой ПЛИС.

Существует довольно большое разнообразие таких платформ, ос-новные требования к которым: наличие достаточного количества сетевых интерфейсов, необходимая емкость ПЛИС для реализации алгоритмов обработки трафика, интеграция с хост системой для высокоскоростного обмена данными. Последнее требование реализуется лучше всего в случае, если платформа имеет интерфейс системной шины ПК, например, PCI или PCI-Express, а не USB. В данной работе используется платформа Net-FPGA [1].

Внедрение устройств – «сенсоров» в инфраструктуру сети, позво-лит лишь проводить пассивный анализ, при этом будет невозможно не только воздействие на параметры обработки пакетов в реальном времени, но и многие «внутренние» характеристики штатного оборудования, необходимые в исследованиях, будут недоступны. Поэтому тестовая платформа должна быть способна и исполнять роль штатного оборудования сети: маршрутизатора, коммутатора или экрана.

Последовательность разработки проектов на ПЛИС платформах существенно сложнее разработки ПО и включает несколько дополнительных стадий, таких как синтез, размещение, моделирование с учетом размещения на кристалле. Особенность используемой платформы Net-FPGA – в конвейере модулей обработки потока пакетов, позволяющем внедрить свой модуль обработки потока, не разрабатывая полный проект устройства с регистрами-портами взаимодействия платформы и хост-компьютера, MAC-уровнем, буферами. В основном использовался именно

этот метод, ориентированный на модификацию базовых проектов: референсных маршрутизатора, коммутатора и сетевого адаптера. Перечисленные проекты распространяются открыто по BSD-подобной лицензии. Ниже приводятся статистики работы референсного маршрутизатора, который является главной основой для модификаций (см. рис. 1).

ИССЛЕДОВАНИЯ В ОБЛАСТИ SDN

Дополнительные возможности в исследованиях можно получить, если и штатное оборудование будет поддерживать управление и способы форвардинга совместимые с тестовой платформой, например, протокол OpenFlow [2]. Это делает возможным синхронное изменение функциональности устройств, протоколов во всей исследуемой сети.

Конечные этапы разработки сетевых протоколов, исследования трафиков компьютерных сетей почти всегда требуется проводить не на моделях, а в реальных условиях: на действующих сетях с реальными пользователями и сетевыми приложениями. Это вызывает ряд проблем:

- действующие сети нельзя подвергать опасности вывода из строя из-за проводимых экспериментов, по крайней мере, следует минимизировать время восстановления рабочего состояния сети;
- часто производители сетевого оборудования защищают свою интеллектуальную собственность, know-how и поставляют устройства с закрытой системой управления не предоставляя API для управления потоком данных внутри устройства и, следовательно, не дают возможность проводить какие-либо изменения этого управления;
- применения одного-двух экспериментальных устройств не дает возможность проводить масштабные изменения в крупной сети, т.к. большая часть оборудования остается неизменяемой.

Одним из решений этих проблем может быть стандартизация управления сетевым оборудованием на уровне потоков данных, иначе – реализация идеи выделения уровня потоков данных (data plane) и уровня управления (control plane), которая обсуждается специалистами телекоммуникационных компаний (NEC, HP, CISCO) и известными школами в области коммуникаций (Stanford, MIT, Princeton) уже около десяти лет. Эта идея начала реализовываться в виде так называемой

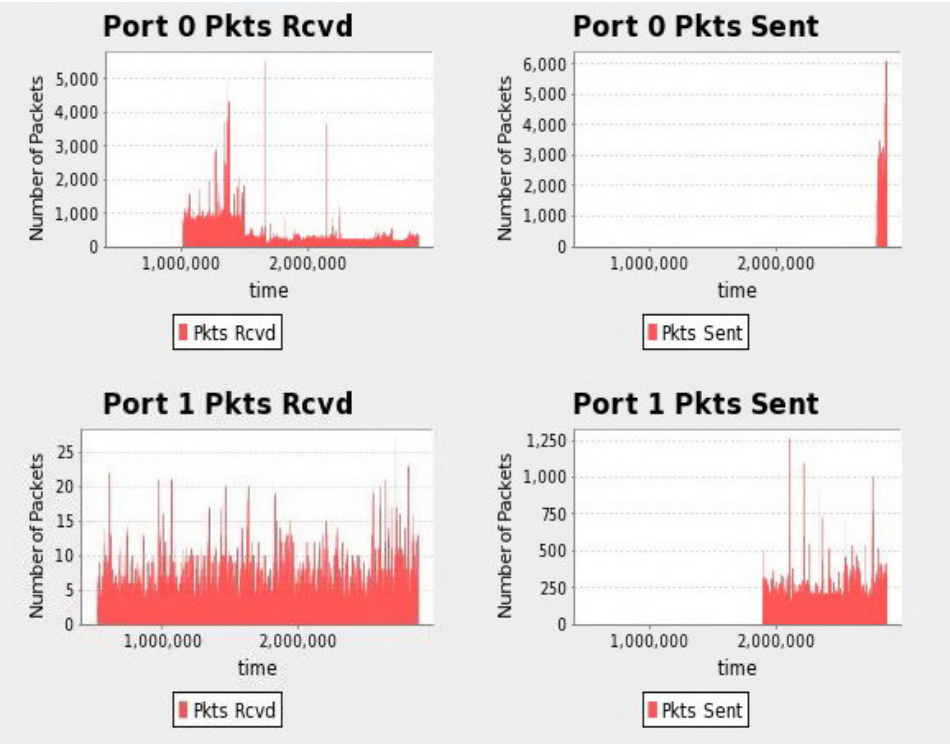

Рис. 1. Статистика портов референсного маршрутизатора

Software-Defined Network (SDN). В 2008 году ряд авторов [1] предложил подход к решению проблемы организации исследований на действующих сетях под названием OpenFlow. Фактически речь идет о программно-определяемой архитектуре сети, управляемой с помощью контроллера (управляющего ПО), которому оборудование, поддерживающее OpenFlow, предоставляет API. Контроллер может таким образом адаптировать сеть к требованиям используемых сетевых служб, в том числе и в режиме реального времени (см. рис. 2).

В узком смысле, OpenFlow – это протокол управления по которому сетевое оборудование управляется контроллером SDN-сети. Поток в OpenFlow определяется [2] как 15-элементный кортеж: *F(Вх_порт, Meta, Eth_S_адрес, Eth_D_адрес, Eth_тип, VID V_приоритет, MPLS_метка, MPLS_класс_трафика, IP_S_адрес, IP_D_адрес, IP_протокол, IP_ToS, S_порт, D_порт)*. Потоки регистрируются контроллером и вводятся в таблицу потоков (flow table) сетевого оборудования, где наряду с кортежем F, для каждой записи о потоке определены счетчики учета пакетов/байтов (counters) и набор действий (instructions). В общем случае, сетевое оборудование OpenFlow работает с несколькими таблицами потоков в режиме конвейера, последовательно сравнивая входной пакет с записями каждой таблицы и выполняя соответствующие инструкции при совпадении параметров пакета с записью о потоке в таблице (см. блок-схему).

Ряд производителей оборудования (CISCO, Juniper, HP) уже выпустил firmware с поддержкой OpenFlow в своих устройствах. Это связано с тем, что помимо исследований, SDN позволит решать и другие проблемы: построение сервис-ориентированных открытых сетей станет задачей программирования. Такая программно-заданная конфигурации сети не будет конфликтовать с уже существующей конфигурацией сетевого оборудования, т.к. сама будет определять эту конфигурацию исходно и полностью, учитывая требования приложений. Требования могут быть описаны на формальном языке и скомпилированы в конфигурацию сети OpenFlow.

В сети факультета компьютерных наук было проведено тестирование платформы на основе ПЛИС-платформы в качестве OpenFlow коммутатора, к портам которой подключены основные генераторы трафика: коммутаторы лабораторий, серверы, точки доступа, программные мосты гипервизора XEN и аплинк в корпоративную университетскую сеть (рис. 3).

Тестовая платформа содержит ПЛИС Xilinx Virtex-II-Pro-50 и па-раллельный PCI-интерфейс (33МГц, 32 разряда). Платформа позволяет, не прерывая обычный трафик, проводить исследования, выполнять обработку передаваемых данных на FPGA на скорости работы портов. В качестве прототипа был использован проект коммутатора Jad Naous [1], со следующей структурой (рис. 4).

Система управления представляет собой OpenFlow контроллер – ПК с процессором

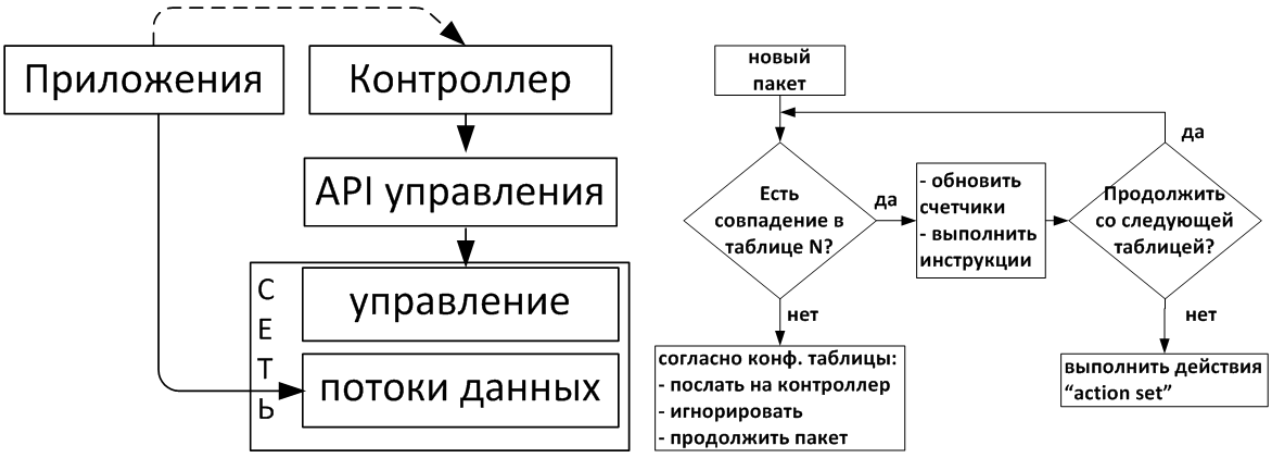

Рис. 2. SDN и OpenFlow

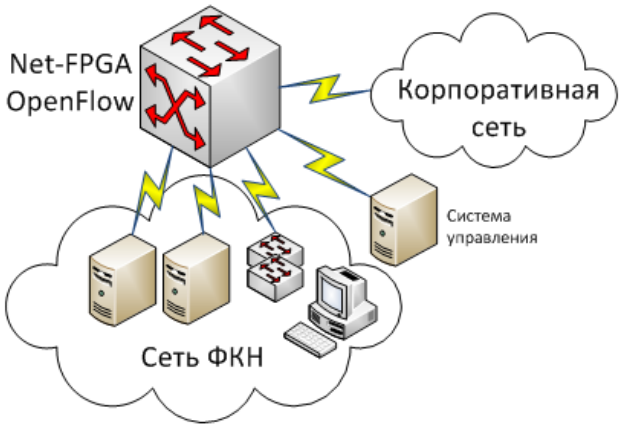

Рис. 3 Тестовая платформа

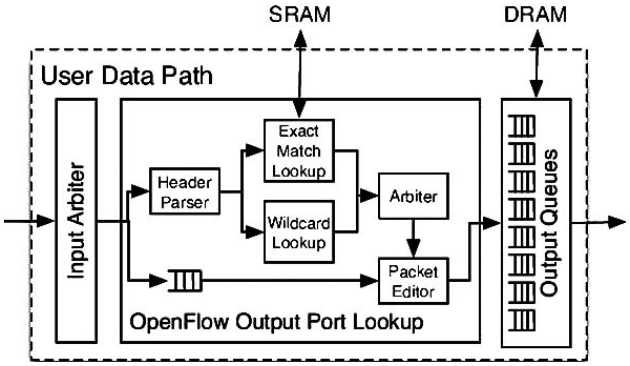

Рис. 4 Структура OpenFlow коммутатора

Intel Pentium-D 3.4ГГц и памятью 1ГБ, систему загрузки конфигураций в ПЛИС, программы для проведения тестов под управлением ОС GNU/Linux (Fedora). При исследованиях сигнатурных методов контекстной обработки, выявление сигнатуры позволяет через механизм управления OpenFlow-коммутатором, включать копирование потока на отдельный порт-монитор для архивирования на жесткий диск или перенаправления на порт, к которому подключен модификатор пакетов.

ТРЕБОВАНИЯ К ПРОИЗВОДИТЕЛЬНОСТИ ПЛАТФОРМЫ

Требования к производительности и пропускным способностям интерфейсов подобных платформ зависят от методов проведения исследований. Анализ трафиков корпоративной сети или сети оператора связи подразумевает получение *агрегированного* трафика. В нашем случае, интерфейсы оборудования сети – 100/1000Base-T, исследовательская платформа имеет 4 порта 1000Base-T. Определим требования к пропускным способностям для разных методов проведения исследований в области трафика:

a) получение и копирование трафика на хост-компьютер;

b) получение и копирование трафика на непосредственно подключенный к платформе диск;

c) получение, регистрация сигнатур на хост-компьютере;

d) пересылка трафика между портами;

e) получение, обработка, передача трафика через платформу.

Существуют две возможности взаимодействия платформы с уст-ройствами записи: через системную шину PCI и через порты SATA уста-новленные на плате платформы. В первом случае пропускная способ-ность в блочном режиме составит около 1 Гбит/с (полудуплекс, PCI) во втором случае – 1.5Гбит/с, 3Гбит/с или 6Гбит/с (полудуплекс, SATA). Нужно также учитывать, что системная шина PCI, в случае копирования трафика на жесткий диск хост-компьютера, будет дополнительно загружена ПДП-транзакциями контроллера жесткого диска (см. рис. 5).

Определим узкие места для вышеперечисленных методов проведения исследований: обозначим F – анализируемый поток в Мбит/с, Сд, Сш, Сп – пропускные способности устройства записи (жесткий диск), шины и исследовательской платформы соответственно. Тогда условия неблокирующей работы платформы будут для методов “a” – “e” иметь вид:

a) $F < \min \{ Сд, Сш/2, Сп\}$
b) $F < \min \{ Сд, Сп \}$
c) $F < \min \{ Сд, Сп \}$
d) $F < Сп$
e) $F < Сп$

На примере задачи регистрации (полного копирования) сетевого трафика можно заметить следующее. Копирование трафика на максимальной для данной платформы скорости потока F = 4 Гбит/с (четыре порта) предъявит сложно-выполнимые требования к пропускной способности ее системной шины (использование параллельной PCI v2.0 –

недостаточно). Непосредственное копирование трафика через SATA-интерфейс сделает возможной запись с большей скоростью, но также сформирует узкое место – Сд. Следует также учитывать, что пропускная способность устройства записи обычно зависит от размеров блоков записываемых данных и, как правило, значительно отличается от скорости интерфейса SATA. В таких случаях, может быть более эффективным метод «е», при котором платформа вы-полняет обработку. В случае регистрации трафика, это может быть сжатие без потерь, которое уменьшит объем копируемых данных (гарантировано за счет заголовков пакетов) и сделает возможной максимальную скорость потока F. Конкретные значения пропускной способности Сп (платформы) зависят от сложности обработки трафика на ПЛИС, которая полностью определяется исследователем. Приведенный пример задачи полного копирования трафика является частным случаем разнообразных задач регистраций трафика, в которых часто требуется идентификация сигнатуры с последующим копированием. Идентификация сигнатур в таких случаях выполняется на ПЛИС.

ПЛАТФОРМА СКЗИ

Замена корпоративных выделенных WAN-сетей Интернет-подключениями, а также, информатизация многих бизнес-процессов, привели к повышению требований к информационной безопасности передаваемых данных через Интернет. Для этого обычно (см. рис. 6) используют различные реализации туннелей VPN (Virtual Private Network), в том числе основанные на семействе протоколов IPsec (IP se-curity) [3].

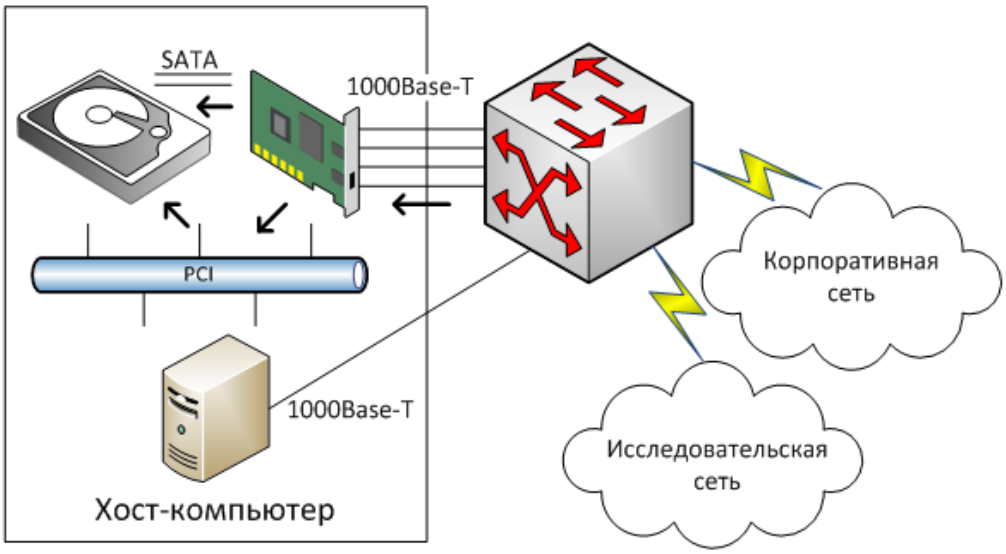

Рис. 5. Регистрация (копирование) трафика платформой

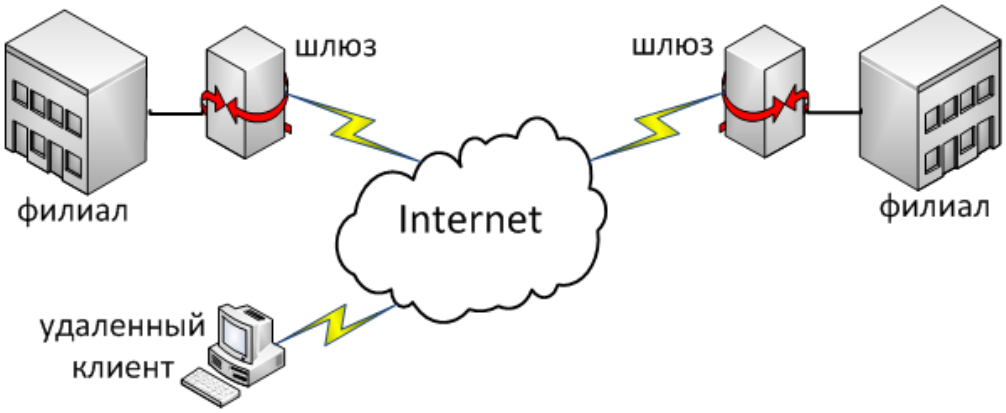

Рис. 6. Туннельные соединения

Кроме того, возросли объемы трафика и количество подключений через Интернет к ресурсам корпоративных сетей. Таким образом, требуется конфиденциально передавать все большие объемы данных, со все большей скоростью и иногда временными ограничениями, поддерживая множество аутентифицированных соединений с шифрацией данных. Существующие программные реализации VPN, например, основанные на IPsec, требуют существенных вычислительных мощностей уже при сотнях па-раллельных соединений с минимальной пропускной способностью (см. табл.)

Из-за больших вычислительных затрат, часто приходится выбирать какой именно трафик должен быть зашифрован и какие именно соединения должны быть гарантировано аутентифицированы. Такой селективный подход может приводить к ошибкам и компрометации данных и/или соединений, которые посчитали не очень важными. В тех случаях, когда требуется поддержка тысяч и более соединений, используют оборудование с аппаратными ускорителями алгоритмов, позволяющими разгрузить процессоры сетевого оборудования. Чаще всего это – ASIC (Application-Specific Integrated Circuit) сопроцессоры или SoC, ориентированные на выполнение одного или нескольких крипто-алгоритмов (обычно: DES, 3DES, AES, MD5, SHA-1/2). Перестройка такой микросхемы на другой алгоритм или даже изменение параметров алгоритма (длины ключа, временных па-раметров) за установленные пределы, невозможна. С другой стороны, сам набор протоколов IPsec позволяет использовать практически любые протоколы для подписи данных и для шифрации [4], в том числе, например, стандартные в СНГ ГОСТ Р 34.11-94 (вычисления хэш-функции, подпись) и ГОСТ 28147-89 (шифрация).

Группа методов, которыми можно решить проблему аппаратной реализации разнообразных крипто-алгоритмов, основана на возможностях некоторых ПЛИС допускать частичную реконфигурацию и, следовательно, возможность модернизировать реализацию крипто-алгоритма или заменить ее на другую. Аналогичные требования возникают на стадии разработки и изучения новых ускорителей крипто-алгоритмов, ASIC, где в качестве прототипа удобно использовать ПЛИС-платформу.

Возможны различные варианты архитектур репрограммируемых платформ, отличающихся взаимодействием универсального процессора и собственно репрораммируемого вычислителя (приводятся в порядке возрастания пропускной способности):

- взаимодействие через системную шину (PCI, PCI-X, …);
- взаимодействие через межпроцессорный интерфейс (HT, …);
- взаимодействие через сопроцессорный интерфейс ПЛИС с аппа-ратно реализованными внутри ПЛИС процессорами (APU);
- взаимодействие через созданный разработчиком интерфейс с Soft-процессором.

Вне зависимости от интерфейса взаимодействия, организация вы-числений может производиться как в однократно измененном репрограммируемом вычислителе, так и в масштабе-времени выполнения задачи. На блок-схеме (см. рис. 7) приводятся основные этапы реализации совмещенного программно-аппаратного выполнения задачи. Возможность режима реального времени зависит от времени репрограммирования и момента синтеза аппаратуры. Время репрограм-

Таблица

условие: 75 % загрузка процессора	Pentium4, 3GHz	Dual Xeon, 3GHz
кол-во IPsec туннелей (30 Kbps на соединение)	700	850
кол-во IPsec туннелей (500 Kbps на соединение)	40	50
максимальная пропускная способность в режиме сервера доступа, Mbps	10	13
пропускная способность в режиме шлюза, Mbps	18	30

мирования, с появлением новых устройств репрограммируемой логики, неуклонно уменьшается. Кроме того, для ряда устройств возможно частичное репрограммирование. Момент синтеза аппаратуры может приходиться как на этап обычной предвари-тельной компиляции, так выполнятся для фрагментов кода в масштабе-времени исполнения программы.

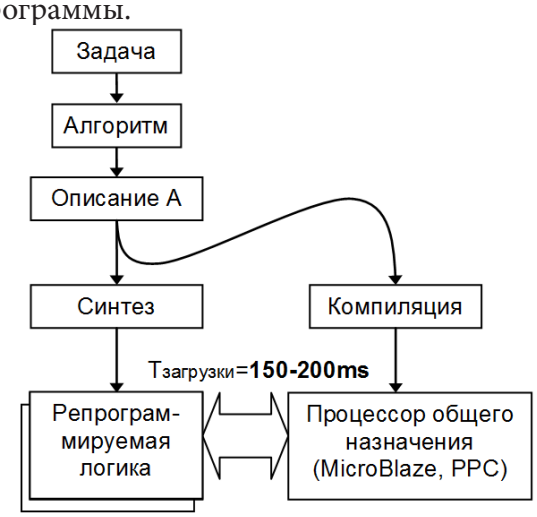

Рис. 7 Этапы программно-аппаратной реализации

Для определения IPsec-производительности сетевых устройств, существует стандартизированная IETF методика "Methodology for Benchmarking IPsec Devices" [5], определяющая следующие измеряемые параметры:

- максимально возможное количество туннелей и SA (параметр Capacity);
- максимальная пропускная способность без потерь (параметр Throughput), измеряемая отдельно для режимов ESP и AH и для различных видов трафика: UDP/IPv4, UDP/IPv6, TCP/IPv4, TCP/IPv6;
- минимальная, максимальная и средняя задержки (параметр Laten-cy).
- процент потерянных кадров (параметр Frame Loss Rate);
- время установки туннеля (параметр IPsec Tunnel Setup Rate).

Аппаратное ускорение крипто-алгоритмов прежде всего повлияет на параметры Capacity и Throughput. Следует отметить, что в современных поколениях процессоров Intel (Core i5, Xeon 5600), в систему команд добавлены 6 инструкций (AES-NI) для аппаратного ускорения AES (выполнение и генерация ключей раундов), однако поддержка других крипто-алгоритмов (в частности ГОСТ) отсутствует. Преимущество в этой связи ПЛИС решения с возможностью частичной реконфигурации – несомненно.

ЗАКЛЮЧЕНИЕ

В данной работе приводится опыт использования ПЛИС-платформы для проведения исследований в области трафиков IP-сетей, рассматриваются варианты проведения пассивных и активных (с модификацией характеристик обработки трафика) исследований. Будущие работы будут проводиться в направлении реализации DPI на данной платформе.

СПИСОК ЛИТЕРАТУРЫ

1. *Naous J.* Implementing an OpenFlow switch on the NetFPGA platform // ANCS '08 Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2008. – pp. 1–9.

2. *McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J.* OpenFlow: enabling innovation in campus networks // ACM Computer Communication Review, Vol. 38 Issue 2, 2008. – 69–74 pp.

3. IETF Network Working Group, RFC 4301: Security Architecture for the Internet Protocol. – 2005. – (http://tools.ietf.org/html/rfc4301)

4. IETF Network Working Group, RFC 4835: Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH). – 2007. – (http://tools.ietf.org/html/rfc4835)

5. *Van Herck T. Kaeo M.* Methodology for Benchmarking IPsec Devices: draft-ietf-bmwg-ipsec-meth-05. – 2009. – (http://tools.ietf.org/html/draft-ietf-bmwg-ipsec-meth-05)

Коваль Андрей Сергеевич – старший преподаватель кафедры информационных систем факультета компьютерных наук Воронежского государственного университета.
Тел.: (473) 2-20-87-24, E-mail: ko-val@cs.vsu.ru

Koval Andrey Sergeevich – senior lecturer of Information Systems De-partment, Computer Science Faculty, Voronezh State University.
Tel.: (473) 220-87-24, E-mail: koval@cs.vsu.ru