УДК 004.725

# ИЗУЧЕНИЕ КЛАССИФИКАТОРОВ ТРАФИКА НА ОСНОВЕ ПРОГРАММИРУЕМОЙ ЛОГИКИ

**А. С. Коваль**

*Воронежский государственный университет*

**Поступила в редакцию 21.08.2015 г.**

**Аннотация.** Приводится опыт использования ПЛИС-платформы для прототипирования сетевого оборудования и проведения исследований в области классификации трафиков IP-сетей, способы размещения платформы в существующей инфраструктуре корпоративной сети и совместного использования.
**Ключевые слова:** IP-сети, трафик сетей, ПЛИС.
**Annotation.** We present experience of the use of FPGA-based platform for network equipment prototyping and research in the field of traffic classification, deployment options inside campus network infrastructure, and sharing among several users.
**Keywords:** IP networks, FPGA, network traffic.

## ВВЕДЕНИЕ

Классификации трафика – это процесс разделения сетевых пакетов или потоков на несколько категорий для применения к каждой категории разных методов обработки или сбора статистики для учета потребленного ресурса, либо анализа состояния сетевой инфраструктуры. Постоянное увеличение объемов и видов передаваемых данных требует всё большей производительности от систем классификации. Одним из способов увеличения пропускной способности классификаторов является аппаратная реализация алгоритмов поиска сигнатур и других методов оценивания потоков данных. Регулярное обновление сигнатур и других признаков классификации дополнительно требует возможности оперативной модификации классификаторов и программируемая логика является одним из подходящих решений. В данной работе используется аппаратная платформа NetFPGA, разработанная в Стэнфордском университете для прототипирования сетевого оборудования и проведения исследований в области трафиков IP-сетей.

## ПЛАТФОРМА ПРОГРАММИРУЕМОЙ ЛОГИКИ

Платформа NetFPGA представляет собой полноразмерную PCI-плату и содержит ПЛИС Xilinx Virtex-II-Pro-50, 4 порта 1000Base-T и параллельный PCI-интерфейс (33МГц, 32 разряда) [1].

*Рис. 1. Фотография репрограммируемой платформы NetFPGA*

ПЛИС Virtex-II Pro содержит 53,136 логических ячеек, 4,176 Кбит блочного ОЗУ, до 738 Кбит распределенной ОЗУ. Особенностью данной ПЛИС является размещение двух нерепрограммируемых вычислительных

ядер процессора PowerPC-405, что потенциально может повысить надежность решений с самомодификацией или с частичным репрограммированием ПЛИС. Пропускная способность – 8 Гбит/с позволяет реализовать неблокирующие режимы передачи. Отдельная оперативная память ZBT SRAM 2x18 Мбит может быть использована для хранения look-up таблиц, например, таблиц форвардинга, динамическая память DDR2 DRAM 64Мбайт – для буферизации потока пакетов с пиковой пропускной способностью 25.6 Гбит/с. К выводам ПЛИС подключены два разъема SATA.

Блок-схема NetFPGA (рис. 2) демонстрирует основные компоненты платформы и пути потоков данных: MAC уровень 1GE интерфейсов реализован на Virtex-II Pro, каждый имеет FIFO-буфер. Блок «Control, PCI Interface» (CPCI) реализован на отдельной ПЛИС – Xilinx Spartan II, что повышает надежность работы хост-системы (по сравнению с однокристальным решением) при ошибках программирования основной ПЛИС. Задача этого блока – организация интерфейса для загрузки конфигурации в основную ПЛИС (Virtex-II Pro) и регистрового ввода-вывода для обмена данными с платформой. Платформа имеет JTAG порт для отладки, например, с использованием среды Xilinx ChipScope.

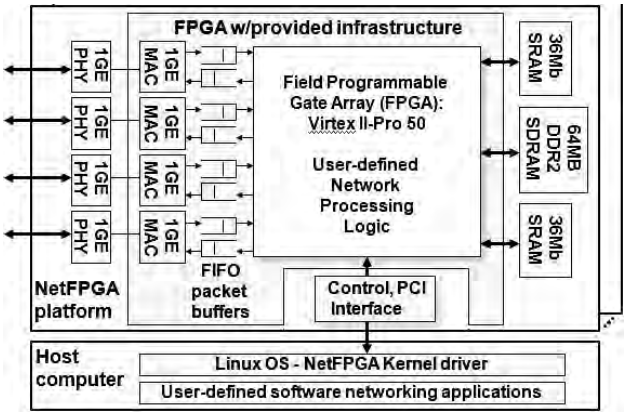


*Рис. 2. Блок-схема платформы NetFPGA [2]*

Последовательность разработки проектов на ПЛИС платформах существенно сложнее разработки ПО, т.к. нужно учитывать временные ограничения, использовать несколько средств разработки и языков описаний, и включает несколько дополнительных стадий: синтез, размещение на ИС, моделирование с учетом размещения на ИС. Базовое ПО (так называемый пакет) и среда разработки NetFPGA поддерживают конвейер модулей обработки потока пакетов, позволяющий внедрить свой модуль обработки потока, не разрабатывая полный проект устройства с регистрами-портами взаимодействия платформы и хост-компьютера, MAC-уровнем и буферами. Однако даже при пересборке проекта, не говоря об изменении состава проекта (добавление, удаление, изменение схемы подключения модулей) требуется среда проектирования ПЛИС Xilinx ISE/XDS. Возможны следующие модели разработки:

1. Модель «маршрутизатор с аппаратным ускорением», в которой отображается маршрутная таблица и ARP-кеш хост-системы на платформу, которая работает со скоростью интерфейса (4x1ГБ/с).

2. Модель «модуль-фильтр», модуль создается на Verilog.

3. Модель – создание проекта «с нуля».

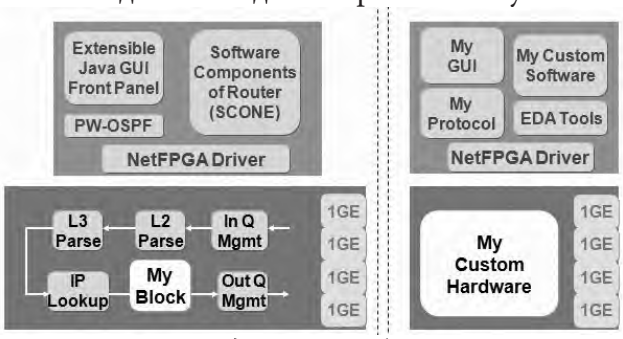


*Рис. 3. Модели разработки 2 и 3*

На рис. 3 представлены модели разработки 2 и 3, в которых, соответственно, блоки «My Block» и «My Custom Hardware» должны быть описаны на языке Verilog. Тем не менее, поскольку драйвер уже разработан, обеспечивает регистровый интерфейс и копирование кадров в режиме DMA между памятью ОС и платформы, разработка значительно упрощается.

В типичном проекте маршрутизатора, пакет в конвейере платформы (рис. 4) проходит 5 стадий: ввод в очередь со стороны сети (MAC) или со стороны компьютера (CPU), входной арбитраж, определение выходного порта и модификация пакета, выходная очередь, отсылка. Интерфейс взаимодействия

модулей конвейера ориентирован на передачу сетевых пакетов и состоит из двух шин: ***данных*** и ***управления*** и двух скалярных сигналов: ***готовности*** к приему пакета последующим модулем и ***сигнала записи***, который формирует предыдущий модуль. По 64-битной шине данных следующему модулю передаются последовательно заголовки модуля, содержащие размер пакета, номера входного и выходного портов, содержимое заголовков Ethernet, IP и данные (полезная нагрузка) пакета.

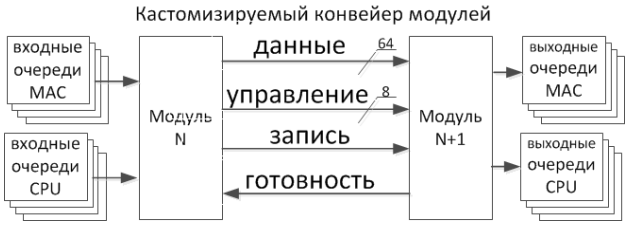


*Рис. 4. Конвейер модулей*

Проведено размещение ПЛИС-платформы в сети факультета компьютерных наук: сетевые интерфейсы NetFPGA подключены к портам коммутатора L3 с возможностью копирования трафика на так называемый «monitoring port», что позволяет проверять классификаторы и IDS на реальном трафике корпоративной сети: доступны трафики лабораторий, серверов, точки доступа беспроводной сети, аплинк в корпоративную университетскую сеть (рис. 5). Изучение классификаторов и работа с различными проектами на одном оборудовании сопряжены с известными проблемами, например, с требованиями разных версий базового пакета NetFPGA, разных версий скриптов, выполняющих сборку проекта и языков скриптов системы сборки (Perl, Python), версий средств разработки на Verilog/VHDL ISE Xilinx. Мы используем XEN-виртуализацию для полной изоляции проектов разных групп. Добавление новой группы возможно добавлением виртуальной машины. Единственное ограничение – версия CPCI (PCI Control FPGA) для которой разрабатываются проекты, определяется загруженной в данный момент в ПЛИС Spartan-II (микросхема 2s200fg456) конфигурацией. Еще одна из особенностей размещения платформы – развертывание контейнеров LXC. Вместо внешнего потока трафика возможна организация внутренней топологии сети на основе LXC с малым потреблением памяти.

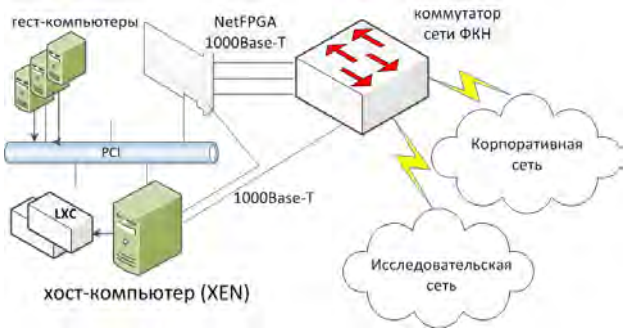


*Рис. 5. Размещение платформы ПЛИС в сети и виртуализация*

Как видно на рис. 5, NetFPGA включена в инфраструктуру сети как сенсор и позволяет лишь проводить анализ пакетов в качестве IDS. Если в проекте необходимо реализовать функцию защиты (IPS), требуется сформировать виртуальную топологию средствами LXC и сконфигурировать NETFPGA как межсетевой шлюз-экран.

## КЛАССИФИКАЦИЯ ТРАФИКА

Классификация трафика обычно реализуется непрерывным анализом заголовков и содержимого пакетов и необходима для решения трех основных задач:

1. Обеспечение качества обслуживания в сети (QoS);
2. Учёт особых платных сервисов (биллинг);
3. Решение проблем информационной безопасности.

Первая задача всегда требует последующего «окрашивания» трафика, установки меток в пакетах для их обработки с различным качеством. Например, часто требуется резервировать полосу для пакетов приложений реального времени: IP-телефонии, приложений видеоконференцсвязи и, наоборот, обрабатывать пакеты P2P файлового обмена по остаточному принципу. Причем анализа заголовков до 4 уровня часто недостаточно, т.к. не все приложения работают с зарегистрированными номерами портов, а в некоторых случаях, пользователи или владельцы ресур-

са сознательно изменяют их для преодоления правил межсетевых экранов. Поэтому даже для QoS-маркировки иногда приходится выполнять анализ вплоть до уровня приложения.

Вторая задача (учёт трафика) частично связана с QoS и стала особенно актуальной в последнее время по экономическим причинам. Из-за высокой конкуренции, основным источником прибыли провайдеров Интернет становятся услуги с высокой добавленной стоимостью, например, IP-телефония. И кроме обеспечения надлежащего качества с помощью QoS, провайдеры учитывают и устанавливают особые тарифы на данные услуги связи.

Третья задача (обеспечения информационной безопасности) чаще всего ассоциируется с классификаторами трафика, т.к. предъявляет особые требования к их производительности и надежности. Например, выявление вируса в потоке пакетов часто выполняется по базам с несколькими сотнями тысяч строковых сигнатур, что несравнимо с выявлением VoIP-трафика. Подобные системы на основе классификаторов называют Системами Обнаружения Вторжений (СОВ) или IDS/IPS (англ. Intrusion Detection/Protection System) [7]. В целом, работу с сетевым трафиком вплоть до уровня приложений называют DPI (англ. Deep Packet Inspection). Все три вышеприведенных класса задач решаются DPI-системам. В 2012 году ITU-T принял стандарт Y.2770 [3], который определяет архитектуру DPI для сетей нового поколения NGN (англ. Next Generation Networks). Планы управления, контроля и данных узла DPI приведены на рис. 6.

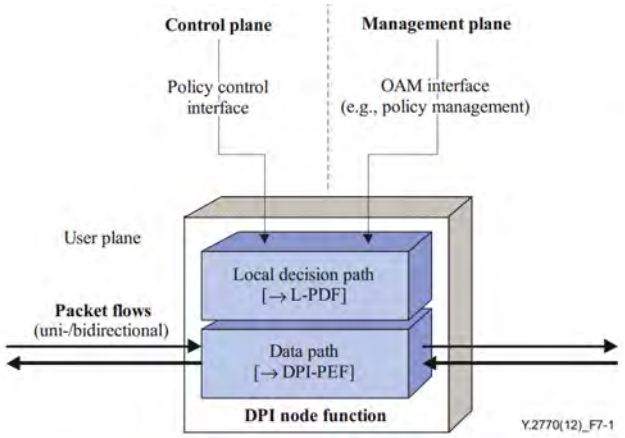


*Рис. 6. Планы управления, контроля и данных узла DPI согласно Y.2770*

## КЛАССИФИКАТОРЫ

Помимо классификаторов, DPI системы состоят и из буферов и механизмов форвардинга данных, интерфейсов систем управления и контроля, памяти хранения правил, политик, журналов. Но пропускная способность систем DPI определяется обычно именно алгоритмом и реализацией классификатора. Для небольших корпоративных сетей (количество компьютеров до 5000) допустимо применять программные решения, среди которых есть и проекты с открытым кодом: Hogwash , Snort, Bro, Suricata. Поставщики услуг связи чаще используют аппаратные решения, которые иногда являются продолжение программных. Например, коммерческие «аппаратные» версии SNORT – Cisco FirePOWER Appliance и Cisco FirePOWER Services.

Поскольку DPI имеет отношение и к информационной безопасности, актуальной является также проблема доверия к закрытым, как правило, аппаратным IPS/IDS решениям. Выход – самостоятельное проектирование и изготовление отечественных «доверенных» сетевых устройств. Прототипирование таких устройств разумно начинать с создания аппаратных классификаторов на открытой платформе, например, NetFPGA. Основные требования к аппаратным классификаторам: высокая пропускная способность (10–100Гб/с) и масштабируемость. Для учебных и исследовательских классификаторов важна возможность периодической актуализации баз данных правил и репрограммируемость. Всем этим требованиям удовлетворяет ПЛИС-платформа.

Прежде всего, заметим, что здесь рассматривается классификация – как задача сопоставления с образцом (англ. pattern matching). Сопоставления 3 уровня или «поиск наиболее специфичного префикса» (англ, Longest Prefix Matches, LPM) – задача решаемая любым маршрутизатором. Варианты решения для 3 и 4 уровней: медленный линейный по-

иск; поиск/выборка из таблицы – 4G записей при прямой реализации; префиксные деревья, бор – предопределенное ограниченное время поиска, может быть построен конвейер, требует организации мнопортовой памяти; фильтры Блума – наименьшие требования к объему памяти, но есть вероятность ложного определения; и, наконец, широко распространенная ассоциативная память, например, троичная (англ., Ternary CAM, TCAM) – занимает много места на кристалле и имеет высокое энергопотребление. Сопоставления более высоких уровней, требуют операций над восстановленным контентом. Наиболее распространены для реализации на ПЛИС: экстенсивный подбор (brute-force); Ахо – Корасик и его множественные модификации; хеш-таблицы и фильтры Блума; деревья принятия решений и их модификации (HyperCuts); КМП-алгоритм (Кнута – Морриса – Пратта).

Brute-force можно применять только для малого количества сигнатур, поскольку количество компараторов растет нелинейно, его сложность $O(m*n)$. Кроме количества компараторов, растет и количество связей, что может исчерпать ресурсы маршрутизации FPGA уже при сотнях правил.

Алгоритм Ахо – Корасик в реализации на ПЛИС требует довольно много памяти, кроме того, желательно использовать возможности параллелизма ПЛИС, организуя несколько независимых вычислителей и конвейерную

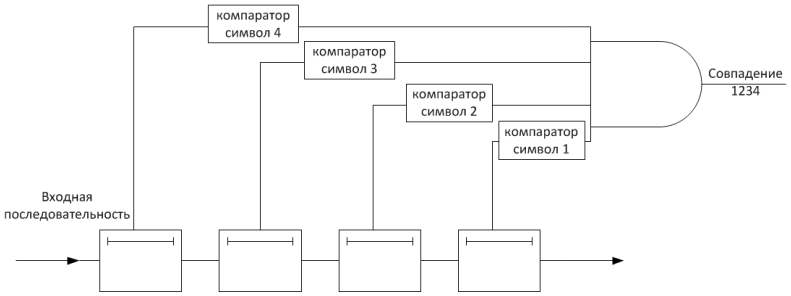


*Рис. 7. Brute-force сопоставление*

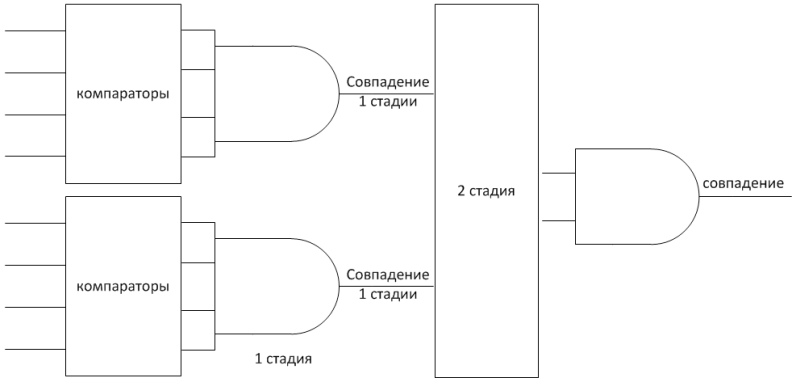


*Рис. 8 Конвейерное сопоставление*

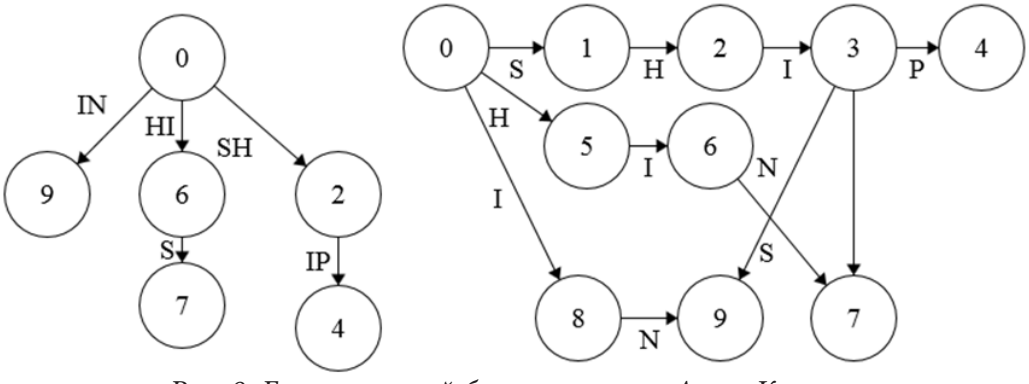


*Рис. 9. Бор и сжатый бор алгоритма Ахо – Корасик*

обработку. Исходная структура данных загружается из базы сигнатур и таким образом статична. В работе [4] было показано, что бор алгоритма Ахо – Корасик можно редуцировать (для конвейерной обработки) до префиксного дерева, которое будет содержать только прямые переходы.

Следует отметить, что в зависимости от приложения классификации, масштаб реального времени может быть разным. Для IDS систем иногда требуется лишь оповещение администратора и копирование подозрительных данных для дальнейшего изучения, что замедляет масштаб времени. В задачах QoS и IPS, большие задержки обычно недопустимы. Потоки данных через или к устройству DPI должны во всех случаях протекать со скоростью заданной интерфейсами оборудования (line-rate).

Постоянно актуализируемые базы данных сигнатур (rules-set) – важная составляющая процесса изучения и построения классификаторов. Обычно используются базы компании Emerging Threats [5], SNORT [5]. Размер баз и размеры сигнатур существенно влияют на возможность загрузки их после предварительной обработки в классификатор. В работе [4] были исследованы базы Snort и ClamAV.

Таблица 1

*Характеристики баз rules-set IDS и антивирусной программы*

| | Snort | ClamAV |
|---|---|---|
| Версия | 2.80 | 0.95.2 |
| Дата | 2009-04-21 | 2009-06-16 |
| Количество паттернов | 9033 | 42020 |
| Всего символов | 197298 | 3025497 |
| Средняя длина паттерна | 21.84 | 72.0 |
| Максимальная длина паттерна | 232 | 382 |
| Минимальная длина паттерна | 1 | 1 |

На октябрь 2015 года, размеры баз несколько изменились: Snort (вер. 2.9.7.6) – 9883, ClamAV (вер. 0.98.7) – 2424387, базы компании Emerging Threats (вер. 8130) для Snort и Suricata содержат, соответственно 18154 и 17424 записей, структурированых по целям атак.

## ЗАКЛЮЧЕНИЕ

В данной работе приводится опыт использования платформы NetFPGA для проведения исследований в области классификации трафиков IP-сетей, способы размещения платформы в существующей инфраструктуре корпоративной сети и совместного использования. Данная статья является продолжением работы [8] с платформой NetFPGA, размещенной в сети факультета компьютерных наук ВГУ для выполнения выпускных квалификационных работ и проведения исследований. Планируется дальнейшее тестирование и оценка алгоритмов и методов классификации, а также реализация самомодифицирующегося классификатора с использованием софт-процессоров или встроенных процессорных ядер PowerPC.

## СПИСОК ЛИТЕРАТУРЫ

1. *Gibb G.* NetFPGA: An open platform for teaching how to build gigabit-rate network switches and routers / G. Gibb, J. W. Lockwood, J. Naous, P. Hartke, N. McKeown // IEEE Transactions on Education, 2008. – Т. 51, № 3. – С. 364–369.

2. Сайт проекта NetFPGA. – (http://netfpga.org)

3. Recommendation ITU-T Y.2770 Requirements for deep packet inspection in next generation networks. – (http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11566)

4. *Jiang W.* Scalable multi-pipeline architecture for high performance multi-pattern string matching / W. Jiang , Y. E. Yang, V. K. Prasanna // IEEE Proceedings IPDPS, 2010. – С. 1–12.

5. Сайт компании Emerging Threats. – (http://www.emergingthreats.net)

6. Сайт проекта SNORT. – (http://www.snort.org)

7. *Коростиль Ю. М. Гильгурт С. Я.* Принципы построения сетевых систем обнару-

жения вторжений на базе ПЛИС / Ю. М. Коростиль, С. Я. Гильгурт // Моделювання та інформаційні технології — Киев: ИПМЭ им. Г. Е. Пухова НАНУ, 2010. – Вып. 57. – С. 87–94.

8. *Коваль А. С.* Проведение исследований в области трафиков IP- сетей на базе ПЛИС-платформы Вестник Воронежского ун-та. Серия: Системный анализ и информационные технологии. – 2014. – № 3. – С. 54–60.

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

**Koval Andrey Sergeevich** – senior lecturer of Information Systems Department, Computer Science Faculty, Voronezh State University.
Tel. (473) 2-20-87-24
E-mail: koval@cs.vsu.ru