Анализ сетевого трафика в режиме реального времени: обзор прикладных задач, подходов и решений*

А.И. Гетьман <thorin@ispras.ru>
Е.Ф. Евстропов <john0606@yandex.ru>
Ю.В. Маркин <ustas@ispras.ru>
ИСПРАН, 109004, Россия, г. Москва,
ул. А. Солженицына, дом 25

Аннотация. В данной работе дается обзор научных исследований в области анализа сетевого трафика в режиме реального времени, а также рассматриваются конкретные программно–аппаратные решения. В работе выделены основные направления развития технологий анализа и показано как они использовались для решения прикладных задач. На основе обширного обзора различных прикладных задач производится выделение общих инфраструктурных компонент и алгоритмов, которые, в тех или иных комбинациях, используются практически во всех готовых решениях. Выделенные компоненты последовательно рассматриваются с указанием подходов, которые применяются при их реализации, возникающих подзадач и ограничений по применимости. На базе выделенных компонент формулируется обобщённая схема анализа сетевого трафика. Исходя из устройства предложенной схемы, выводится список требований, предъявляемый, как к отдельным компонентам, так и ко всей схеме в целом. Рассматриваются основные классы существующих систем анализа, с точки зрения особенностей их подключения к сетям обмена данными и используемой программно-аппаратной базы. Рассматривается вопрос масштабирования систем, при увеличении пропускной способности контролируемого канала.

Ключевые слова: анализ сетевого трафика; глубокий анализ пакетов; перехват пакетов на высокой скорости; классификация трафика**.**

1. Введение

Анализ сетевого трафика на сегодняшний день — очень обширная тема. Под «анализом сетевого трафика» мы будем понимать совокупное название технологий и их реализаций, позволяющих проводить накопление, обработку, классификацию, контроль и модификацию сетевых пакетов в зависимости от их содержимого в реальном времени. Одним из осложняющих факторов, при рассмотрении данного вопроса, является двойственность развития средств

* Работа поддержана грантом РФФИ 14-07-00606 А

анализа сетевого трафика: с одной стороны — это развитие алгоритмов и подходов к анализу, с другой — развитие программно-аппаратных средств для эффективного решения этой задачи. В свою очередь, это приводит как к путанице в терминологии, так и к сознательному манипулированию фактами и цифрами в маркетинговых целях. В данной работе сделана попытка отразить, как историческое развитие, так и текущее состояние данной области с научной и прикладной точек зрения. Также делается попытка систематизировать сведения о совокупности технологий, содержащиеся в публикациях. Для достижения этой цели предполагается выполнение следующих шагов:

1. Провести ретроспективный обзор развития прикладной сферы анализа сетевого трафика для понимания исторического пути этой технологии.
2. Рассмотреть историческое развитие применяемых технологий методов и схем анализа, для обеспечения вновь возникающих прикладных задач.
3. Исследовать необходимость и оптимальность применения тех или иных подходов и алгоритмов для решения конкретных прикладных задач.
4. Выделить общую схему анализа, использующуюся в подавляющем числе конкретных систем анализа сетевого трафика.
5. Для каждого этапа этой общей схемы:
   - указать к какому именно объекту применяется анализ на этом этапе,
   - какие данные можно получить из этого объекта, путём дополнительной обработки.
   - какие подзадачи требуется решать на этих этапах и проблемы, возникающие при решении этих подзадач.
6. Отразить список устройств, решений и оригинальных разработок, как имеющих непосредственное отношение к данной технологии, так и тех, которые, не будучи связанными с ней напрямую, используются во многих практических схемах реализации данной технологии.
7. Провести обзор текущего уровня технологий и аспектов их прикладной реализации.

2. История развития средств анализа сетевого трафика

Зарождение технологий анализа сетевого трафика можно отнести к началу 90х годов прошлого века. Потребности в их возникновении появились примерно в одно время в нескольких областях.

Усложнение схем сетей и многообразие сетевых устройств привели к усложнению их настройки и поддержки сети в работоспособном состоянии — необходим был инструмент позволяющий, с одной стороны локализовать проблему, а с другой предоставить как можно более исчерпывающую

информацию о природе проблемы. Собственно объектом, который содержит в себе всю необходимую информацию и является сетевой трафик. Одним из инструментов, изначально предназначенным для решения именно этой проблемы стал сетевой сниффер/анализатор[1] Wireshark [2](ранее Ethereal), созданный инженером Джеральдом Комбом (Gerald Comb) в 1997 году. Wireshark продолжает активно развиваться и является стандартом в определённой области сетевого анализа.

В это же время начинает применяться технология трансляции адресов NAT [3], предназначенной как для того, чтобы сэкономить IP адреса, так и для того, чтобы скрыть от внешнего наблюдателя устройство и ресурсы внутренней локальной сети. Для реализации этой технологии требовался инструмент — аппаратный или программный транслятор адресов. Данный функционал в результате был внедрён в качестве составной части в большинство маршрутизаторов. Существуют и программные реализации, как в составе серверных операционных систем, так и в виде отдельных приложений [4].

К этому же времени относятся первые упоминания о вирусах и DoS/DDoS [5] атаках, в основном типа Syn flood — первое упоминание о DDoS относится к 1996 году. Для защиты от этих угроз требовался инструмент, анализирующий и фильтрующий пакеты до их попадания на основной сервер. Одним из видов таких защит стали межсетевые экраны (firewall). Первое поколение данных решений относилось к типу пакетных фильтров (packet filters), которые обрабатывали пакеты по одному (не учитывая предысторию) и анализировали только уровни L1-L3 модели OSI и (для протоколов TCP/UDP) номера портов из транспортного уровня L4 (см. рис. 1). Для определения типа трафика (web, email и т.д.) использовался список фиксированных номера портов из каталога IANA [6]. Процесс анализа заключался в сравнении данных, извлечённых из пакета, с набором заданных правил и, в зависимости от результата — блокировка или пропуск пакета в сеть с занесением события в журнал и опциональным уведомлением источника пакета о ситуации. Например, правило «Блокировка Telnet трафика» выглядело, как правило, описывающее пакеты, транспортный протокол которых — TCP, номер целевого порта — 23, а действие при выявлении такого пакета — блокировка. Одним из первых подобных решений был продукт DEC SEAL.

Ближе к концу 90х — началу 2000х годов, в связи с ростом сетевых потоков данных, актуальными стали ещё две задачи, требовавшие сетевого анализа: балансировка нагрузки между серверами и ускорение работы отдельных видов сетевых приложений. К сетевым приложениям, требовавшим ускорения, относились, прежде всего, приложения, использующие протоколы HTTP, DNS, SSL [7]. Для решения второй проблемы использовались, т.н. прокси-сервера, осуществляющие кэширование поступающих данных, минимизируя, так образом, обмены по сети.

Устройства, разработанные для решения обеих этих задач (инкапсулирующие, в частности, функционал прокси-серверов) носили название контроллеры

доставки приложений (Application delivery controllers, ADC). Такие решения в частности были разработаны компаниями Alteon, Radware, F5, Brocade, Cisco.

В первой половине 2000х годов сетевые технологии получили бурное развитие — появились средства голосового обмена по сети (VoIP) и обмена данными в одноранговых сетях P2P (Napster, KaZaA), что, в частности, привело к очередному резкому скачку объёмов передаваемых по сети данных. Для развивающихся сетей крупных корпораций потребовалось объединять в единую локальную сеть территориально разнесённые площадки. Более частыми и сложными стали сетевые атаки, что требовало более развитых средств защиты.

Для реализации передачи управляющих сигналов и данных VoIP с использованием таких протоколов как SIP[8] и RTP[9] между различными провайдерами, как телефонной связи, так и интернета требовались специальные устройства – пограничные контроллеры сессий (session border controllers, SBC) [10], которым требовалось выделять соответствующий трафик из общего потока. Данные устройства производились в таких компаниях как Acme Packet, Audiocodes, Cisco, Genband.

Для решения проблемы эффективного обмена данными между разными сегментами распределённой сети, соединёнными каналом ограниченной пропускной способности (данная проблема имеет название Channel optimization) был разработан целый спектр техник под общим названием Wan Optimizations [11]. Среди этих техник можно указать:

- Дедупликация (Deduplication)– уменьшение повторной передачи данных за счёт сохранения на обоих концах обмена повторяющихся элементов данных и последующей передачи ссылок на эти данные вместо самих данных. Может осуществляться на разных уровнях сетевого стека (в частности, TCP и IP)
- Сжатие (Compression) – передача данных по каналу в сжатом виде с последующим разжатием на другой стороне.
- Оптимизация латентности - упреждающая отправка сетевых пакетов-подтверждений TCP.
- Кэширование получаемого содержимого. Реализовывалось с помощью прокси-серверов, наиболее распространёнными из которых были Web-прокси, кэшировавшие содержимое сайтов. Примерами такого ПО являются Squid и NetCache.
- Объединение нескольких пакетов интенсивных сетевых протоколов, таких как CIFS [12], в один (protocol spoofing).

Данные техники впоследствии реализовывались как в виде отдельных сетевых устройств (Middleboxes [13]), так и программно, на мощных серверах (Network appliances). Одним из первых производителей стала компания Riverbed, впоследствии купившая анализатор Wireshark и интегрировавшая его в свои продукты.

В сфере сетевой безопасности в этот период также произошли значительные изменения. Усложнение сетевых атак привело к тому, что их стало затруднительно с достаточной точностью определять по отдельным пакетам, а скорость появления новых атак — к необходимости реагирования на ещё неизвестные их виды. В совокупности это привело к появлению методов защиты на основе анализа поведения сетевых потоков (tcp session behaviour analysis). В то же время стали появляться вредоносные сайты, заражающие их посетителей, а также методы внедрения вредоносного функционала в не заражённые сайты. Для защиты от таких атак потребовалось внедрение обновляемых чёрных списков сайтов и необходимость фильтрации и блокировки по URL. Среди производителей средств защиты можно указать Arbor, BlueCoat, SonicWall.

Наиболее полное развитие технология анализа сетевого трафика получила, начиная со второй половины 2000х годов, в связи с несколькими факторами:

- Непрекращающийся рост объёмов передаваемых данных.
- Рост ширины каналов, обеспечивающих возможности для передачи этих объёмов.
- Увеличение количества разнообразия передаваемых данных, в частности тех, которые могут использоваться для составления различных профилей, как отдельных пользователей, так и различных групп.
- Рост как разнообразия сетевых угроз и атак, так и их количественные характеристик.

Эти факторы привели к росту потребностей со стороны провайдеров интернета (internet service providers, ISP) и различных компаний. Интересы этих групп различны, но, в тоже время, имеют значительные пересечения.

Так, например, общей областью интересов является защита сетевых ресурсов, которая, в свою очередь, делится на ряд направлений:

- Антивирусные решения (AV).
- Развитые межсетевые экраны Next Generation Firewalls (NGFW).
- Системы обнаружения и предотвращения сетевых атак Intrusion detection/prevention systems IDS/IPS.
- Системы защиты от DDoS-атак.

В то же время, специфичной областью интересов провайдеров интернета является [14, 15]:

- Обеспечение качество связи в часы наибольшей нагрузки (ЧНН) с учётом экономии на расширении арендуемых каналов связи.
- Получение конкурентного преимущества за счёт возможности предлагать более выгодные индивидуальные тарифы с учётом индивидуального профиля пользования сетевым каналом.
- Регулирование полосы пропускания для некоторых видов трафика. Одной из основных проблем является P2P трафик, который, может

занимать значимую часть арендуемого провайдером канала (до 60-80%[16]), приводя к тому, что чтобы обеспечить необходимое качество сервиса (quality of service, QoS) провайдеру приходится ускоренными (по сравнению с прогнозами роста абонентской базы и пользовательских потребностей) темпами расширять данный канал.

Основной областью интересов компаний, предлагающих свои товары и услуги с использованием Интернета, являются «профили» пользователей с точки зрения их интересов и предпочтений. Подобные профили можно опосредованно выявить, в частности, с помощью списка сайтов, которые пользователь посещает, набора его поисковых запросов, сетевых приложений, которые он использует.

К другой группе относятся компании, предоставляющие различные интернет сервисы, например, с помощью технологии виртуализации сетевых функций (Network Function Virtualization, NFV). К таким сервисам можно отнести:

- облачные сервисы,
- сервисы защиты,
- хранения и др.

Для этих компаний, специфичным является вопрос управления большими объёмами входящего трафика — требуется балансировка и интеллектуальное управление.

В соответствии с приведённым выше историческим развитием потребностей в области сетевых сервисов происходило развитие технологий анализа сетевого трафика, ложащихся в основу аппаратных, программных и гибридных решений.

3. Направления развития технологий анализа сетевого трафика

Можно выделить два основных направлений развития.

- Рост «глубины» анализа для отдельного сетевого пакета, то есть увеличение уровня модели OSI, данные которого подвергаются анализу.
- Полнота учёта состояния потока, к которому относится пакет, а также других потоков, связанных с данным.

В следующих разделах будут рассмотрены оба этих направления развития.

3.1 Глубина анализа сетевых пакетов

По этой «оси» технологии анализа трафика развивались последовательно, каждая последующая наследовала часть предыдущих механизмов и добавляла свои. Можно выделить три уровня развития технологии, которые приведены на рис. 1.

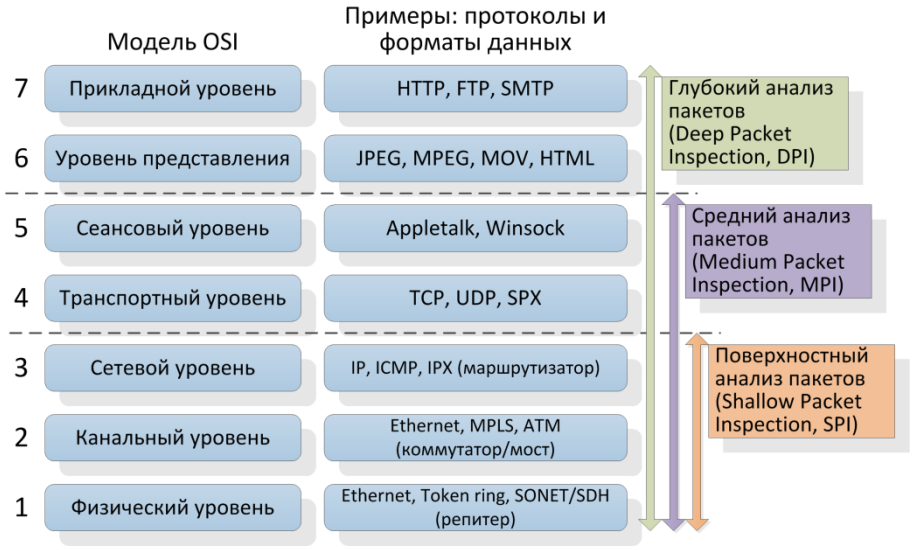

Рис. 1 – Уровни развития технологии анализа сетевого трафика по «глубине».

Рассмотрим эти уровни более детально.

3.1.1 Поверхностный анализ пакетов (SPI)

Технология анализа трафика, основывающаяся исключительно на заголовках пакета уровней L1-L3 по модели OSI. Предъявляет низкие требования к вычислительным ресурсам, что позволяет анализировать большие объёмы трафика. Технология широко распространена, на её основе работает большинство межсетевых экранов операционных систем (в частности в ОС Windows XP/Vista и OS X), маршрутизаторов и других сетевых устройств. На её основе реализованы сетевые списки контроля доступа на уровне IP адресов и портов (Access Control List, ACL). Таким образом, данная технология хорошо подходит для разграничения доступа извне к отдельным компьютерам (IP) и сервисам (порты) внутренней сети.

3.1.2 Средний анализ пакетов (MPI)

Технология анализа трафика, основывающаяся на инспектировании сессий и сеансов связи, инициированных приложением, но устанавливаемых шлюзом-посредником (см. рис. 2). Также применяется термин «прокси приложений» (application proxy). В рамках данной технологии содержимое пакетов анализируется частично и по предопределенным правилам. Не используются сложные методы анализа типа сигнатурного. Устройства, реализующие данный функционал размещаются между провайдером интернета и конечным

пользователем. Данные устройства разбирают заголовки вплоть до транспортного уровня и небольшую часть данных пакета для сопоставления разобранной части с некоторым списком разбора (parse list), с последующей реакцией в случае их обнаружения. Данные списки обычно короче списков ACL и предоставляют более широкий диапазон действий в отличие от «разрешить/запретить» в случае ACL. Эти списки также более выразительны, так как позволяют привязываться не к IP-адресам, а к формату данных пакетов и данным некоторых протоколов уровня приложения, например, URL-адресам в случае протокола HTTP. С помощью MPI можно, например, заблокировать возможность получения flash-файлов или картинок с определённых интернет сервисов (на уровне представления OSI) или заблокировать часть команд (на уровне приложения OSI) в отдельных протоколах. Набор протоколов, как правило, очень ограничен. Например, в первых версиях CheckPoint FireWall-1 (CheckPoint FW-1) поддерживались протоколы Telnet, FTP, HTTP, а в Cisco Private Internet Exchange (Cisco PIX) - FTP, HTTP, H.323, RSH, SMTP и SQLNET. Впоследствии данные наборы незначительно расширялись. Также известно, что данная технология используется в продуктах компаний McAfee и Symantec. Межсетевые экраны, использующие данную технологию, относятся ко второму поколению [17].

Данная технология более гибкая в сравнении с SPI и, помимо разграничения доступа, подходит для большего числа задач — кэширование содержимого, анализ сжатого/шифрованного трафика, ограничение функционала отдельных протоколов путём запрета отдельных команд. Благодаря подключению в режиме прокси, может служить в качестве Wan Optimizer'a (см. выше).

Основной недостаток MPI — плохая масштабируемость: каждая команда и протокол требуют отдельного «шлюза» (входной-выходной порты). Кроме того, работа в режиме прокси сильно снижает скорость обработки. Для снижения нагрузки на прокси-сервер был разработан протокол ICAP [18], позволяющий прокси-серверам отправлять проходящие через них данные для проведения анализа сторонним серверам на предмет безопасности или анализа содержимого. Эта схема реализована в антивирусном продукте ClamAV, который может подключаться к прокси-серверам Squid и NetCache, упомянутым выше.

Эти факторы сильно ограничивают применение данной технологии на уровне провайдеров интернета вследствие необходимости анализа большого числа протоколов и команд на широких каналах связи.

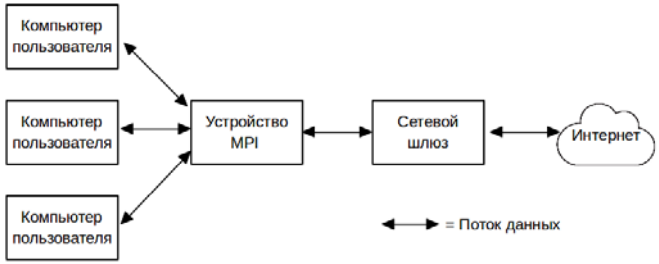

Рис. 2 - Схема применения устройств анализа на основе технологии MPI.

3.1.3 Глубокий анализ пакетов (DPI)

Иногда употребляют более узкий термин — DPP (Deep Packet Processing), который подразумевает такие действия над пакетами, как модификация, фильтрация или перенаправление. Сегодня оба термина часто используются как взаимозаменяемые [19]. Данная технология является логичным развитием MPI. В рамках данного подхода анализатор просматривает содержимое каждого пакета полностью. Одним из важных отличий от предыдущих технологий является то, что системы на базе DPI могут принимать решение не только по содержимому пакетов, но и по косвенным признакам, присущим каким-то определённым сетевым программам и протоколам. Для этого может использоваться статистический анализ. Например, анализ частоты встречи определённых символов, длин пакетов, расстояние между метками времени последовательных пакетов и т.д. Также, по сравнению с предыдущими подходами, значительно расширен список применений технологии: классификация, ограничение полосы, приоритезация, маркировка, кэширование и т. д. Технология DPI получила развитие, прежде всего, из-за стремительного роста вычислительных способностей процессоров, их быстродействия и, соответственно, возможностей для более полного и точного анализа сетевых данных.

В отличие от MPI, данная технология изначально разрабатывалась для высокоскоростной обработки и идентификации большого числа приложений в реальном времени. Таким образом, решения на основе DPI хорошо масштабируются как по ширине сетевого канала (известны решения, работающие на каналах порядка 100 Гбит/сек), так и по числу идентифицируемых приложений (в существующих решениях — порядка нескольких тысяч). С точки зрения реализации, основной компонент любого решения DPI - модуль классификации, отвечающий за классификацию сетевых потоков. При этом в зависимости от целей применения DPI, классификация может выполняться с различной точностью:

- тип протокола или приложения (например, Web, P2P, VoIP)
- конкретный протокол уровня приложения (HTTP, BitTorrent, SIP)
- приложение, использующее протокол (Google Chrome, µTorrent, Skype)

Важно отметить, что соответствие между классами различных уровней точности не однозначно, что показано на рис. 3.

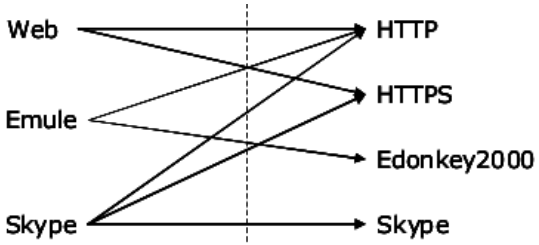

Рис. 3 - Различие между идентификацией приложений (слева) и протоколов (справа).

Технология DPI на данный момент является текущим стандартом де-факто для средств анализа сетевого трафика и относится к области критически важных технологий необходимых для обеспечения, как сетевой безопасности, так и требований законодательства. Вследствие этого в последнее время на международном уровне был принят ряд стандартов, требований и рекомендаций по особенностям реализации, внутреннему устройству и набору функций соответствующих средств [20, 21]. Эта технология редко применяется в межсетевых экранах — это скорее область IDS/IPS систем, в качестве исключений можно указать экраны Hogwash и Shield. Однако межсетевые экраны, относящиеся к четвёртому поколению [17] могут учитывать данные IDS/IPS систем в процессе анализа.

3.2 Учёт состояния потока при анализе сетевого трафика

Вторым направлением развития технологии анализа можно назвать учёт состояния протокола (потока) в процессе анализа — т.н. stateless/statefull виды анализа. Данное направление актуально только для протоколов, использующих транспортный протокол с установлением соединения (connection-oriented). Это означает, что перед любым обменом командами и данными происходит процесс «установления соединения», в ходе которого стороны обмениваются фиксированной последовательностью пакетов, которая часто называется «рукопожатием» (handshake), а после завершения обмена происходит аналогичный процесс «закрытия соединения». К connection-oriented протоколам, в частности, относится протокол TCP, но не UDP. Однако следует учесть, что поверх UDP может быть реализован другой транспортный протокол, с установлением соединения. В качестве примера можно привести протокол Quick UDP Internet Connections (QUIC) [22] — протокол транспортного уровня с установлением соединения, использующий UDP. Из этого следует, что, в общем случае, нельзя полностью исключить statefull анализ для UDP пакетов.

Для описания различий описанных подходов требуется дать определение понятию «поток пакетов». Известны различные определения данного понятия. Часть из наиболее широко используемых приведена на сайте Center for Applied Internet Data Analysis (CAIDA)[23]. В данной работе мы будем использовать «односторонний поток транспортного уровня» — последовательность пакетов передающихся с заданного IP-адреса и TCP/UDP порта на данный IP-адрес и TCP/UDP порт, с указанием протокола транспортного уровня (TCP/UDP). Таким образом, поток задаётся пятёркой <srcIP, srcPort, dstIP, dstPort, protocol>. С учётом данного определения, можно сформулировать отличие statefull от stateless подхода. Оно состоит в том, что в случае statefull подхода учитывается тот факт, к какому именно потоку относится анализируемый пакет, и результат (состояние) анализа предыдущих пакетов этого же потока, если данный пакет не первый. В случае если пакет первый — проверяется, что он является корректным пакетом установления соединения. Следует также отметить, что понятие «statefull» не вполне чёткое и может иметь разные градации с различным «состоянием», что приводит к различному балансу точность анализа/ресурсоёмкость/скорость работы [24, 25]. Один из вариантов градации можно видеть на рис. 4. Список уровней учёта состояния потока, который там отражён – следующий:

- Анализ отдельных пакетов без учёта потоков и состояний (Packet Based No State, PBNS).
- Анализ пакетов в рамках потоков (Packet Based Per Flow State, PBFS).
- Анализ сообщений в рамках потока (Message Based Per Flow State, MBFS), т.е. произведена сборка IP-фрагментов в IP-пакеты (IP-нормализация) и сборка TCP-сегментов в TCP-сеансы (TCP-нормализация).
- Анализ сообщений в рамках протокола (Message Based Per Protocol State, MBPS), т.е. учитывается состояние автомата протокола (возможность принимать тот или иной тип сообщений). Пример автомата состояний протокола HTTP приведён на рис. 5. Вершины соответствуют состояниям, рёбра — условиям перехода, к которым могут относиться приём/отправка сообщения, результаты обработки сообщений, истечение таймаута.

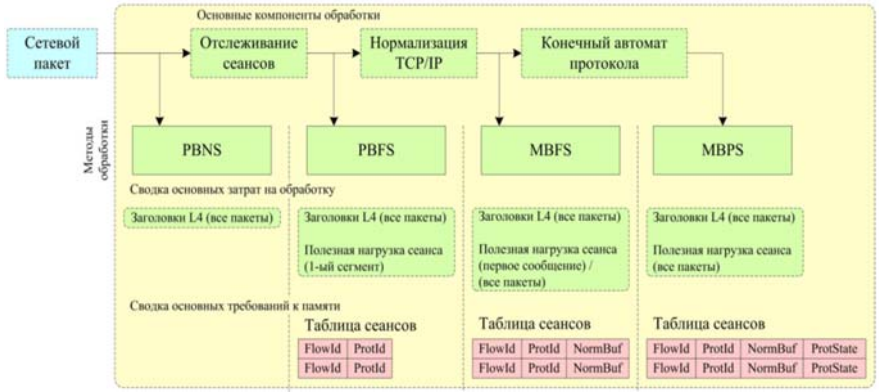

Рис. 4 - Градации полноты учёта состояния потока.

Базовые реализации технологии DPI часто относятся к stateless-анализу, то есть анализ выполняется на уровне отдельных пакетов, состояние между анализом нескольких пакетов одного сетевого потока не сохраняется. Этого уровня точности хватает для многих практических приложений и позволяет значительно экономить ресурсы (см. рис. 4). В то же время, существуют задачи, для которых такого уровня точности не достаточно. В качестве примеров можно привести две технологии, использующие statefull подход — инспекция пакетов с хранением состояния (statefull packet inspection, SPI) и глубокий анализ содержимого (deep content inspection, DCI).

3.2.1 Анализ сетевых пакетов с учётом состояния потоков

В рамках SPI подхода, программа или устройство, которое его реализует, в момент открытия нового соединения проверяет его на соответствие заданной политике безопасности и до закрытия хранит параметры этого соединения в памяти. С помощью таких решений, в частности, осуществляется проверка корректности соединения, например отсутствие пакетов на открытом сетевом порте после завершения соединения. Реализации SPI содержатся в большинстве современных маршрутизаторов в виде SPI-брандмауэров. Также эта технология используется в программных межсетевых экранах, учитывающих состояние (stateful firewalls), компании CheckPoint и ряде IDS/IPS систем. Межсетевые экраны, использующие эту технологию, относят к третьему поколению [17]. При данном подходе отслеживаются не только входящие и исходящие пакеты, но и состояние отдельных соединений, которое хранится в динамических таблицах. Благодаря этому при анализе очередного пакета могут учитываться не только заданные правила и политики по отношению к адресам и содержимому пакетов, но и состояние соединения, к которому относится пакет и предыдущих пакетов, которые к нему относятся, а также и других, связанных с данным, соединений. Классический пример преимущества межсетевого экрана поддерживающего состояние потока по сравнению с межсетевыми

экранами без такой поддержки — обработка FTP протокола. Данный протокол открывает новый поток передачи данных на каждую соответствующую команду, причём поток открывается на случайном порте, большем 1024. Так как межсетевой экран не имеет возможности узнать, что новый поток относится к допустимому FTP протоколу — этот поток будет заблокирован. В случае наличия поддержки состояний потоков — адресная
информация нового потока будет добавлена в таблицу легитимных потоков и сессия будет пропущена в сеть.

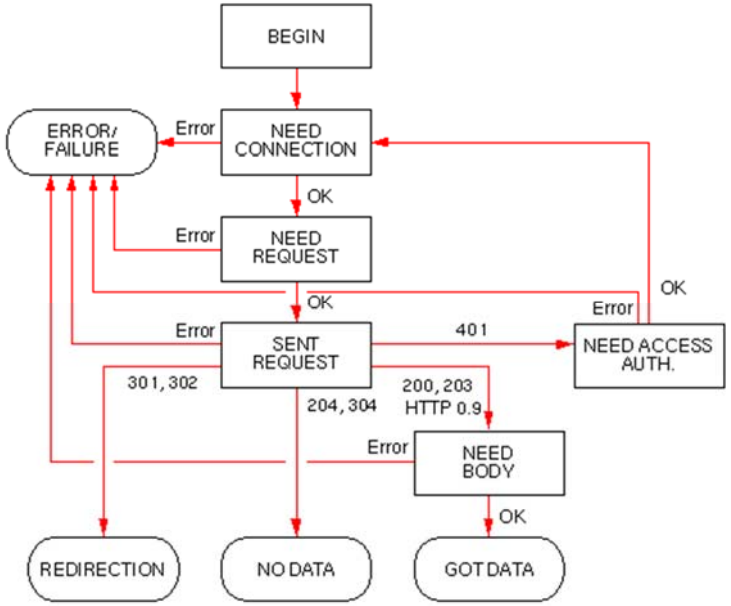

Рис. 5 - Пример автомата состояний протокола HTTP.

3.2.2 Анализ содержимого сетевых протоколов прикладного уровня

В рамках технологии DCI выполняется не только идентификация протокола конкретного сетевого потока, но и группировка потоков в группы, отвечающие за предоставление некоторого сервиса, например сигнального протокола, например, SIP и протокола передачи данных, например, RTP, в случае VoIP. Также в процессе применения DCI, анализ не останавливается на идентификации протокола, например, HTTP, но также делается попытка определить приложение, которое его использует (например, Gmail) и собрать контент этого приложения в том виде, в каком оно было передано приложением для отправки по сети (электронное письмо). Примером использования данной технологии может служить функция прослушивания VoIP звонков по перехваченному трафику в анализаторе WireShark [26].

С точки зрения функционала, основной вклад DCI в дополнение к модулю классификации (основной функционал DPI) — набор модулей разбора для различных протоколов прикладного уровня и различных видов данных в различных кодировках (например, MIME [27]), которые они содержат. Функции модулей разбора, сводятся к двум основным:

1. Разбор буфера данных (сетевого пакета или собранной сессии), в соответствии с форматом сообщений протокола, описанным, как правило, на одном из специальных языков типа ASN.1 [28] и P4 [29].
2. Сборка сессий для протоколов с установлением соединения и их последующий разбор (пункт 1).

Одной из тенденций последнего времени в развитии средств DPI/DCI является универсализация и централизация анализа. Данная концепция может быть обозначена как «DPI как сервис» - под этим названием она была приведена в работе [30]. Суть концепции заключается в том, что если в сети используется большое число различных средств, реализующих тот или иной анализ трафика (межсетевые экраны, системы IDS, оптимизаторы трафика и др.), то имеет смысл вынести весь анализ в отдельное устройство. Это устройство будет выполнять полный разбор сетевых данных и рассылать результаты анализа всем устройствам в зависимости от их потребностей, а те, в свою очередь, реализовывать только реакцию на поступающие данные. Переход к этой концепции в чём-то аналогичен переходу к программно-конфигурируемым сетям (Software Defined Networks, SDN) [31] в вопросах управления трафиком, при котором все решения по используемым алгоритмам маршрутизации и уровне её выполнения переходят от конкретных маршрутизаторов к выделенным устройствам - SDN-контроллерам. Такие подходы упрощают масштабирование систем и позволяют эффективно расширять функционал без дополнительных работ по интеграции и перенастройке оборудования.

Концепция «DPI как сервис» может быть эффективно реализована в рамках систем унифицированного управления угрозами (Unified threat management, UTM) и унифицированного управления безопасностью (Unified security management, USM). Эти системы также являются отражением тенденции централизации в виде объединения функционала межсетевых экранов, сетевых систем IDS/IPS, антивирусов, VPN-серверов, фильтров содержимого, балансировки нагрузки и предотвращения утечек данных в рамках единой системы.

Демонстрацией этих тенденций является выделение функционала распознавания протоколов и извлечения метаданных в виде отдельных модулей. Причём эти модули могут быть, как чисто программными, так и привязываться к некоторой аппаратуре. Примерами программных реализаций являются Qosmos Intelligence Engine [32], ipoque PACE [33], Windriver Content Inspection Engine [34], Procera PacketLogic Content Intelligence [35]. Среди привязанных к аппаратуре модулей можно указать Cisco Network Based Application Recognition (NBAR) [36] и Junos OS Next-Generation Application

Identification [37]. Использование этих модулей в виде составной части систем контроля и управления трафиком позволяет формулировать политики безопасности и другие виды политик в гораздо более высокоуровневых терминах, например в терминах URL, имён приложений, отдельных функциональностей в рамках этих приложений (например, блокирование передачи голоса в рамках Skype, при сохранении возможности обмена текстовыми сообщениями). По сути, набор функций данных модулей аналогичен расширению функционала технологии MPI на произвольное множество протоколов, их команд и данных, которое поддерживаются конкретным модулем распознавания протоколов. Типичная схема использование такого решения [14] приведена на рис. 6, где «Внешний интерфейс» — решение типа «DPI как сервис», PCRF - Policy and Charging Rules Function — устройство, хранящее политики и правила, применяемые к трафику, «Внутренний интерфейс» — устройство хранящее статистику, журналы, результаты применения правил к трафику, и т.д.

Рис. 6 - Схема использования системы DPI для применения политик к сетевому трафику.

Концепция «DPI как сервис» может также рассматриваться как отделение инфраструктурной части анализа сетевого трафика от бизнес-логики в рамках отдельных прикладных задач (сбор статистики, межсетевой экран, IDS/IPS системы и др.). В следующем разделе будет рассмотрена схема работы именно инфраструктурной части анализа, так как она является, с небольшими вариациями, идентичной в различных решениях для анализа сетевого трафика. В частности, будут выделены отдельные этапы анализа с кратким описанием их особенностей, а в последующих разделах каждый этап будет рассмотрен более подробно.

4. Общая схема инфраструктурных алгоритмов анализа сетевого трафика

Общая схема анализа сетевого трафика состоит из следующей последовательности шагов, каждый из которых приводит к повышению уровня представления объекта анализа.

1. Захват пакетов, проходящих через контролируемое сетевое соединение. Результатом данного шага является получение объекта анализа в виде сетевых пакетов. В зависимости от необходимой точности и скорости последующего анализа, а также доступных вычислительных мощностей могут использоваться различные подходы.

   - Слайсинг (slicing), при котором анализу подвергаются не всё содержимое пакетов, а только некоторый префикс (n первых байт). В ряде исследований (например, [38]) показано, что этот подход хорошо работает для последующей классификации трафика по протоколам. В частном случае, если перехватываемый размер равен суммарному размеру сетевых заголовков (L1-L3) является реализацией технологии SPI.
   - Самплинг (sampling), при котором перехватываются не все пакеты, а только их часть, которая может выбираться по различным условиям, в зависимости от потребностей. В процессе развития технологии было предложено большое число стратегий отбора [39]. Например, для задач мониторинга типов трафика подходит вариант с выбором каждого n-го пакета (uniform sampling), где n может выбираться в зависимости от соотношения ширины канала и пропускной способности системы анализа. Задача получения информации о полном состоянии сети по результатам самплинга известна как inversion problem [40], в частности, при применении uniform sampling происходит недооценка среднего размера пакетов, так как чаще будут отбираться пакеты меньшего размера [41]. Для передачи перехваченных данных используется протокол PSAMP [42].
   - Наконец, для задач, в которых требуется максимально точный анализ трафика, например для систем обеспечения сетевой безопасности, требуется перехватывать все данные всего поступающего трафика без потерь — для обозначения этого подхода используется термин lossless capture или deep packet capture (DPC).
2. Агрегирование пакетов в потоки по некоторым адресным признакам (flow generaion [43]), получение нового объекта для анализа — сетевого потока. Если при этом данные пакетов в дальнейшем анализе не учитываются, то такой вид анализа называется «анализ потоков» - flow

based analysis (в отличие от packet-based анализа, при котором анализируются данные пакетов). На рис. 7 показаны различия типичных схем packet и flow-based анализа. Flow-based анализ широко используется в силу значительно меньших требований к мощности вычислителя и пропускной способности, за счёт значительного снижения объёма данных для обработки. Такой вид анализа может выполняться как локально [43], так и удалённо от точки сбора данных [44]. Для передачи собранных данных от точки сбора до точки анализа используется большое число протоколов, часть из которых стандартизирована в виде IPFIX [45], а часть разработана отдельными производителями — Cisco NetFlow, Juniper Jflow. В рамках подхода записи, описывающие поток могут содержать разный набор данных. Наиболее общим набором таких данных является следующий:

- IP адреса источника и адресата,
- протокол транспортного уровня,
- в случае протоколов TCP/UDP — номера портов источника/адресата,
- набор счётчиков: количество переданных пакетов и байт, время создания и завершения потока.

Следует отметить, что хотя данный метод действительно значительно снижает требования к анализатору, тем не менее, он не является достаточно гибким, так как в отличие от слайсинга и самплинга не позволяет варьировать количество поступающих данных (оно зависит от входных данных). Более того в большинстве реальных задач количество потоков незначительно меньше количества пакетов (примерно на порядок) из-за большого числа очень коротких потоков, состоящих из нескольких пакетов — flash flows [46]. Для решения этой проблемы было предложено использовать самплинг для потоков [41]. Другой особенностью данного метода является то, что, вследствие ограниченности памяти, устройство, осуществляющее агрегацию пакетов, не может отслеживать один поток на протяжении произвольного промежутка времени. Для решения этой проблемы в конкретном решении обычно присутствует настройка, ограничивающая максимальную продолжительность потока (5 минут, в случае Cisco NetFlow [40]). По истечении этого времени считается, что поток завершился, и информация о последующих пакетах агрегируется в рамках «нового» потока. Исследование точности flow-based подхода и влияния этого эффекта на точность анализа содержится в работе [47]. Также в этой публикации описан инструмент FLOW-REDUCE, осуществляющий «сборку» полной информации о потоке из фрагментов, на которые она была разбита из-за ограничений по времени.

3. Выполнение классификации по протоколу прикладного уровня или конкретному сетевому приложению. Результатом данной операции является получение нового объекта для анализа — сетевого потока конкретного протокола или приложения (в этом случае связанных потоков может быть несколько, например, в случае VoIP приложения это потоки SIP и RTP). После выполнения данной операции возможна следующая дополнительная обработка полученного объекта, конкретный вид которой зависит от решаемой прикладной задачи:
   - разбор полей протокола (protocol parsing),
   - сборка сессии протокола для протоколов с установлением соединения,
   - извлечение данных приложения (content extraction) — страниц сайтов (HTML), файлов различных типов (исполняемые, изображения, текстовые документы, и т.д.), электронных писем, аудио-видео потоков и т. д.,
   - разбор данных приложения (application content parsing).

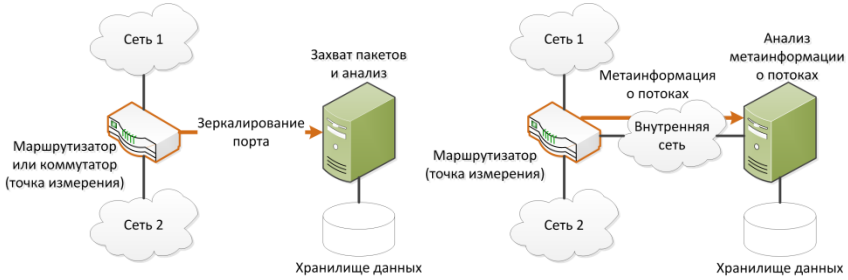

Рис. 7 - Различия типичных схем packet (слева) и flow-based (справа) анализа.

Для полноты картины, следует сказать, что помимо указанных выше packet-based и flow-based подходов существует ещё один источник данных о сетевом трафике — т.н. база управляющей информации (Manage Information Base, MIB) [48] – виртуальная база данных, используемая для управления объектами в сети связи.

Модули для накопления, хранения и обмена данными в формате MIB реализованы в большинстве устройств. Передача данных осуществляется по протоколу SNMP [49]. Данные получаемые таким путём имеют низкий объём и неспецифичны для протоколов. Например, в рамках данного подхода, можно получить сведения об общем количестве пакетов и байт прошедших через конкретный сетевой интерфейс конкретного сетевого устройства.

Следует сказать, что одной из причин развития MIB и flow-based подходов, несмотря на их сравнительно низкую точность, послужила до сих пор идущая глобальная дискуссия [50] о законности и допустимости глубокого анализа

трафика с точки зрения нарушения безопасности, прав на частную жизнь и т. д. На данный момент одним из следствий данной дискуссии является, в частности, то, что в научных работах, трафик, который подвергается глубокому анализу предварительно проходит процедуру «анонимизации» с помощью специальных средств [51].

Далее будут более подробно рассмотрены отдельные шаги из приведённой общей схемы анализа сетевого трафика, методы, алгоритмы и подходы, а также их особенности и ограничения применимости.

4.1 Захват сетевых пакетов

Программные и аппаратные средства, осуществляющие захват трафика относятся к классу сниферов (sniffers). Для решения задачи захвата трафика могут использоваться как стандартные серверные сетевые карты, так и специализированные сетевые карты, предназначенные для перехвата трафика на предельных скоростях без потерь. Специализированные карты, как правило, реализованы на базе FPGA или ASIC и имеют встроенные средства для проставления временных меток, аппаратной фильтрации, снятия некоторых заголовков низкоуровневых протоколов, балансировки нагрузки между процессорами на многопроцессорных компьютерах с учётом IP-потоков, выявления ошибочных и дублирующихся пакетов. При этом вся обработка (в том числе и копирование данных в память компьютера из памяти сетевой карты) осуществляется без привлечения ресурсов ЦПУ. По мере развития технологий многие из описанных свойств реализуются и на базе стандартных сетевых карт. Технология реализации таких дополнительных функций носит название TCP Offload Engine (TOE). Она включает в себя следующие различные технологии, базовыми из которых являются следующие:

- Large Segment Offload (LSO) или Giant send offload (GSO)— сегментация больших TCP-пакетов при отправке
- Large Receive Offload (LRO) — сборка приходящих отдельных сетевых пакетов в большие сегменты
- Checksum Offload — проверка контрольных сумм в заголовках IPv4, IPv6, TCP и UDP
- IP Security (IPSec) Offload — шифрование/дешифрование трафика протокола IPSec

Основной проблемой для стандартных сетевых адаптеров является не скорость передачи данных, как таковая, а количество пакетов в единицу времени. Это обусловлено особенностями внутренней реализации обработчиков пакетов на сетевых картах, драйверов сетевых карт и программных сетевых стеков ОС. Вследствие этого, стандартные сетевые карты без специализированных драйверов и сетевых стеков не обеспечивают перехват трафика без существенных потерь на скоростях более 3 Mpps (миллионов пакетов в секунду). Причины такого ограничения будут рассмотрены ниже. Ещё одной проблемой является точное проставление временных меток.

Проблемы, возникающие при переходе к сетевым соединениям, поддерживающим более высокие скорости передачи данных, связаны в основном с несколькими факторами:

- Ограниченной пропускной способностью аппаратуры.
- Архитектурными ограничениями при взаимодействии аппаратуры с ОС и ОС с пользовательскими приложениями.
- Объёмом памяти, необходимым для хранения получаемых данных.

Большинство распространённых систем анализа трафика работают, используя библиотеки Libpcap (ОС Linux) и WinPcap (ОС Windows). Данные библиотеки работают в пользовательском режиме. Для обеспечения своей работы со стороны ОС они используют драйверы уровня ядра Berkeley Packet Filter (BPF) и Netgroup Packet Filter (NPF) соответственно. Основная разница между этими драйверами заключается в схеме их работы с буферами памяти, использующимися для временного хранения пакетов, получаемых от сетевой карты. Драйвер BPF использует схему с двойной буферизацией, в то время как драйвер NPF использует кольцевой буфер [52].

Среди проблем этих решений, приводящих к снижению производительности можно выделить.

- Двойное копирование данных пакета (из карты в память ядра, из памяти ядра в память пользовательского процесса).
- Большое число прерываний от сетевой карты (на каждый пакет, чтобы он был скопирован в буфер ядра).
- Большое число переключений между режимами ядра и пользователя (на каждый пакет при его копировании в память пользовательского процесса).
- Недостаточное использование параллелизма на уровне отдельных ядер и процессоров (по умолчанию все прерывания обрабатываются одним ядром).
- Проблемы с синхронизацией при доступе к данным из нескольких потоков выполнения. В случае, если полученные данные должны обрабатываться в несколько потоков между этими потоками возникает ситуация соревнования за ресурсы.

В зависимости от количества копирований данных пакетов, которые выполняются в процессе перехвата, решения разделяются следующим образом.

- **0-copy (zero-copy).** Для реализации подхода с нулевым копированием требуется аппаратная поддержка со стороны сетевой карты – она должна содержать собственный DMA контроллер, копирующий данные с карты в память программы пользователя, без дополнительного копирования через память ядра. Примером может служить библиотека PF_RING ZC в связке с сетевыми картами Intel или Napatech [53]
- **1-copy.** Для реализации этого подхода возможны несколько

вариантов — разработка анализатора на уровне ядра, что является весьма сложной задачей или прямое отображение памяти ядра в память пользовательского процесса.

- **2-copy.** Стандартное решение на базе LibPcap или WinPcap.

Для решения перечисленных проблем было реализовано некоторое количество специализированных драйверов и сетевых стеков, к которым относятся, например, коммерческое решение Sniffer10G от Emulex и Myricom, а также открытая разработка PF_RING компании Ntop. Эти решения используют схему с кольцевым буфером, как более эффективную, а также оптимизированы для многопроцессорных и многоядерных компьютеров. В частности они реализуют следующий функционал:

- Обработка перехвата пакетов с использованием большого числа нитей исполнения (одна нить на входную очередь).
- Балансировка нагрузки между ядрами (одно ядро – одна входная очередь).
- Пакетная фильтрация внутри сетевой карты.

Для реализации этих функций используется как аппаратная поддержка со стороны архитектуры, так и поддержка со стороны ОС (специализированное API). Среди используемых технологий можно выделить следующие.

- Набор близких технологий Interrupt Moderation, Adaptive Interrupt Moderation, Interrupt Coalescing, Interrupt Blanking, Interrupt Throttling, позволяющих управлять задержкой доставки прерываний за счёт настраиваемого таймера и обрабатывать получение/отправку множества пакетов за одно прерывание.
- MSI-X — распределение I/O прерываний по нескольким процессорам и ядрам.
- New API (NAPI) — интерфейс уровня ядра ОС Linux, позволяющий применять технику уменьшения количества прерываний (interrupt mitigation) со стороны сетевых устройств.
- Receive-side Scaling (RSS) — технология, предоставляющая возможность динамической балансировки нагрузки входящих сетевых пакетов по нескольким ядрам и процессорам (прерывания поступают на разные процессоры). Существуют реализации для масштабирования на случаи более 64 процессоров. Данная технология поддерживается в семействе ОС Windows с появлением Scalable Networking Pack. В ОС Linux аналог этой технологии называется Linux Scalable I/O.

Также существует ряд аппаратных технологий от различных производителей процессоров, предназначенных для ускорения ввода/вывода.

- Intel Integrated I/O - технология прямого подключения шины PCI Express 3.0 к процессору (без отдельного PCI-контроллера), реализованная в семействе Intel Xeon E5.

- Direct Cache Access (DCA) – предоставление устройствам ввода/вывода, таким как сетевые адаптеры, возможности помещения данных напрямую в кеш процессора Intel.

4.2 Группировка сетевых пакетов в потоки

Группировка пакетов в потоки — достаточно стандартная и простая операция. Основное отличие разных реализаций данного функционала связано с тем, какие именно поля адресной информации и как использовать для идентификации потока. Наиболее употребляемое определение потока было дано ранее. Так как оно использует 5-ку полей как ключевую информацию для определения принадлежности конкретного пакета к конкретному потоку, то для его обозначения и обычно используют термин 5-tuple. Также иногда используются двусторонние потоки, симметричные к перестановке пар <srcIP, srcPort> и <dstIP, dstPort>. Модуль, отвечающий за группировку пакета обычно называют генератором потоков (flow generator). В процессе работы данный модуль хранит в памяти отображение соответствующей ключевой информации на данные конкретных потоков. При появлении нового пакета, с ним производятся следующие операции.

1. Из пакета извлекается ключевая информация, позволяющая идентифицировать, к какому потоку он принадлежит.
2. Производится поиск по текущему множеству потоков.
3. Если поток найден – в данных потока увеличиваются соответствующие счётчики – как правило, к ним относятся время жизни потока, количество пакетов и байт в потоке. Если поток не найден - создаётся новая запись потока и в неё добавляется информация о текущем пакете.

В работе [38] проведена оценка вычислительных ресурсов, необходимых для выполнения первых двух операций, а также для операции классификации (в случае использования детерминированных конечных автоматов). Результаты оценки приведены в Табл. 1. Абсолютные цифры, приведённые на рисунке, на данный момент могут быть не вполне актуальны, но их ценность, прежде всего, в относительной стоимости операций.

Таблица 1 - Оценка скорости выполнения основных операций при анализе трафика

Операция	Стоимость (такты процессора)
Извлечение идентификатора потока	78
Поиск/добавление идентификатора потока	49
Поиск сигнатуры с помощью детерминированного конечного автомата (мин., ср., макс.)	13-4331-8900

Описанная выше базовая схема, хоть и является корректной, но неполной. Она содержит существенный недостаток — предполагается, что модуль располагает бесконечной памятью, так как отсутствует определение условий завершения потока и поэтому непонятно, когда следует удалять запись о потоке из отображения. В случае транспортного протокола с установкой соединения (например, TCP) в этом протоколе предусмотрена явная процедура завершения соединения (обмен FIN-ACK пакетами или посылка RST пакета). В случае протоколов без установления соединения (например, UDP) такой подход не работает, поэтому, как правило, используется один из вариантов, основанных на использовании таймера — например, обрыв соединения через 5 мин (такой вариант используется в коммутаторах Cisco NetFlow). Этот же подход используется для слишком долгих TCP-потоков [43].

4.3 Классификация сетевого трафика

Тема классификации сетевого трафика сама по себе является очень обширной. Прежде чем переходить к методам, которыми она осуществляется, перечислим варианты классификации по её результатам, то есть объектам, которые получаются на выходе данного алгоритма, их свойствам и возможностям их дальнейшей обработки. По этому критерию, можно выделить три основных варианта классификации. Далее они перечислены в порядке увеличения «точности» классификации:

- **Тип трафика** не является достаточно содержательным способом классификации и, как правило, или не подвергается дальнейшему анализу, или подвергается достаточно простой дополнительной уточняющей классификации. В зависимости от сферы применения, типы могут быть различными. Среди примеров, можно указать:
  - P2P, видео-стриминг, веб-трафик — в случае систем сбора статистики и мониторинга,
  - трафик сетевой атаки/нормальный трафик — в случае систем защиты от сетевых атак,
  - трафик, содержащий/не содержащий объекты копирайта, в случае систем контроля копирайта.
- **Использующийся протокол прикладного уровня (protocol identification)** является достаточно содержательным и может, как использоваться непосредственно — например, в системах сбора статистики и мониторинга для повышения уровня точности. Основным способом дальнейшей обработки является разбор протокола, включающий два основных функции — сборка сессии прикладного уровня, в случае необходимости извлечение данных протокола из отдельных его полей (метаинформация уровня протокола).

- **Приложение, передающее данные (application identification),** дает максимально детализированный уровень классификации. На этом уровне могут осуществляться те же виды обработки, что и на уровне протокола прикладного уровня, а также извлекаться и интерпретироваться данные (метаинформация) конкретного приложения, что соответствует более высокому уровню их представления. Например, поле типа «строка», определённое на уровне протокола, может соответствовать «имени пользователя» на уровне приложения.

В различных прикладных задачах результаты идентификации протоколов и приложений могут интерпретироваться и, соответственно подвергаться различной последующей обработке (как и в случае идентификации типа трафика).

Например, в случае системы защиты от вредоносного кода, под протоколом может пониматься командный (command-and-control, C&C) протокол ботнета, а под приложением — конкретный вирус. Соответственно, извлекаемая метаинформация — команды ботнета, передаваемые им данные, а цель анализа — выяснение его функционала, оценка распространённости и исследование возможностей его деактивации.

В случае системы составления профиля пользователя для последующей демонстрации таргетированной рекламы (например, iMarker) в роли протокола может выступать HTTP, в роли приложения — браузер, а объектом анализа является запрос пользователя к поисковой системе, который подвергается дальнейшему текстовому анализу для извлечения ключевых слов.

Выбор конкретной прикладной задачи может значительно влиять как на выбор алгоритма классификации, так и на его параметры и производительность. В качестве примера можно рассмотреть следующее сравнение. В случае системы статистики, алгоритм классификации обычно работает последовательно на пакетах каждого потока «до первого срабатывания». Схема такой классификации приведена на рис. 8.

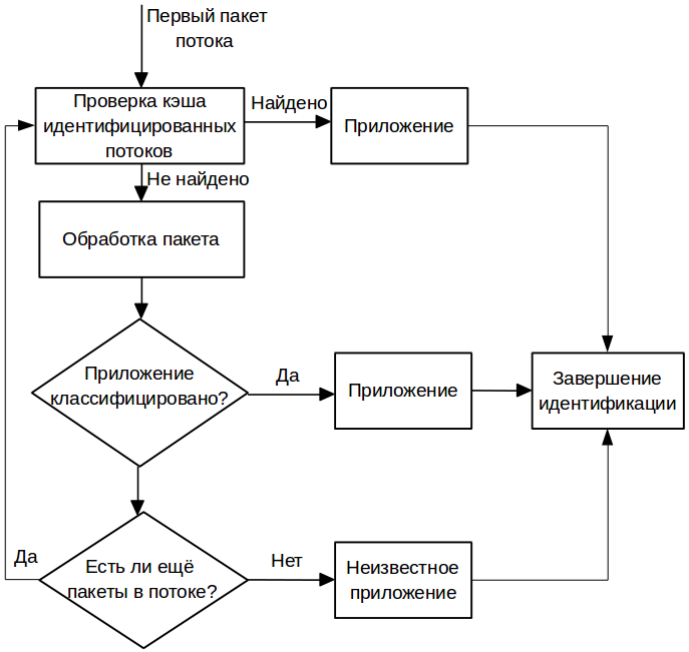

Рис. 8 - Схема классификации «до первого срабатывания».

В случае систем фильтрации по ключевым словам такой метод не подходит, так как в одном и том же сетевом потоке, в различных пакетах могут встретиться различные слова и, с точки зрения системы классификации, в этом случае данный поток попадёт сразу в несколько классов.

В общем случае, очевидно, что первый подход гораздо производительнее, так как приходится анализировать значительно меньшие объёмы данных. Кроме того, в ряде подходов, для дополнительного ускорения, анализируют не всё содержимое пакета, а только некоторый его префикс (по аналогии со слайсингом). Например, в работе [54], для идентификации потоков, содержащих шифрованные и сжатые данные, используются только первые 16 байт пакетов.

В работе [38] проведена оценка влияния размера анализируемого префикса пакета на точность классификации по протоколам и скорость работы классификатора на трёх снятых сетевых трассах Unibs-GT, Polito, Polito-GT. Результаты приведены на рис. 9, где на левом графике ошибки классификации обозначены как misclassified, а трафик, который не удалось классифицировать, как unknown.

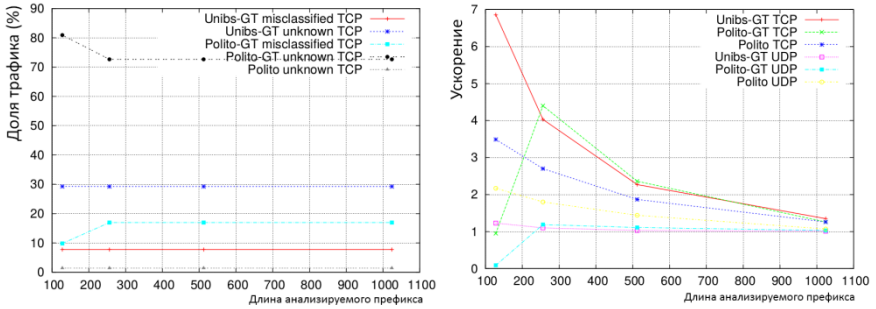

Рис. 9 - Оценка влияния длины префикса на точность классификации (слева) и скорость (справа).

На основе этих исследований, в частности делается вывод об избыточности проведения IP-дефрагментации и TCP-нормализации при решении данной задачи, так как данные алгоритмы (особенно второй) достаточно ресурсоёмки и практически не влияют на точность. Это происходит из-за того, что для классификации, как правило, используется не более 256 байт пакетов, а минимальный размер фрагмента обычно не меньше 576 байт. То есть, для данной задачи PBFS подход более предпочтителен, чем подход MBFS (см. рис. 4).

Рассмотрев виды классификации по получаемым результатам и подходы в разных прикладных задачах, перейдём к рассмотрению конкретных алгоритмов классификации.

Классическим подходом к классификации является анализ содержимого пакетов (payload-based). При этом, как правило, выполняется поиск т.н. «сигнатур» (signature-based подходы) - характерных признаков, которые заранее создаются для каждого приложения или их групп. Классификация может выполняться как на уровне отдельных пакетов (stateless анализ), или может учитываться состояние потока (statefull анализ). Для повышения точности распознавания часть подходов использует уточнённые «сигнатуры» на основе автоматов состояний протоколов (см. рис. 5). При таком подходе, получаемые сообщения, после их классификации, сопоставляются с переходами в различных автоматах протоколов, и оценивается корректность последовательностей таких переходов. Эта группа подходов называется Stateful Protocol Analysis Detection [55].

Как было показано на рис. 9, классификация является наиболее нагруженным алгоритмом анализа сетевых пакетов. Исторически, из-за нехватки вычислительных мощностей, предпринимались попытки достижения увеличения производительности алгоритма за счёт выбора источника данных, используемых алгоритмом в процессе классификации, таким образом, чтобы обрабатываемые данные, будучи не менее информативными, чем содержимое пакетов, были бы более компактны. Эта группа подходов (в отличие от «сигнатурного») относится к классу «основанных на выводе» (inference-based).

Одним из важных преимуществ inference-based подходов является то, что качество анализа не зависит от представления данных в сетевых пакетах, в частности, отсутствуют ограничения при анализе сжатого/шифрованного трафика. Далее будут рассмотрены основные подходы к решению задачи классификации, их особенности и ограничения применимости.

4.3.1 Подходы на основе вывода

Все подходы на основе вывода можно разделить на группы по двум основным параметрам:

- используемые для вывода данные,
- используемый для их анализа алгоритм.

Все виды данных, в свою очередь, можно разделить на:

- характеристики отдельных пакетов в рамках отдельного потока (packet based),
- характеристики потоков в целом (flow based).

К первой группе относятся подходы, использующие такие характеристики как: временные промежутки между пакетами, последовательности размеров пакетов [56], и др.

Ко второй группе относятся два основных подхода.

- Подход на основе анализа портов (port-based) при котором идентификация происходит по одному из номеров портов потока, на основе базы данных о характерных статичных портах, которые используют зарегистрированные в IANA протоколы (регистрировать можно любой номер порта, а не только первые 1024). Этот метод считается малоэффективным, так как на данный момент существует большое число протоколов с динамическими номерами портов. В частности, к таким протоколам относятся практически все реализации P2P. Кроме того, часто используются схемы, при которых трафик некоторого протокола (например, HTTP) передаётся по нехарактерному для него номеру порта (не 80 в случае HTTP).
- Подходы на основе статистической информация об активности отдельных хостов в сети: в скольких и каких именно обменах данными (потоках) участвовал данный хост, сколько данных, и в какую сторону передавалось и т.д. Эти данные сопоставлялись с набором заранее созданных шаблонов различных видов серверов. Один из таких подходов описан в работе [57].

Алгоритмы анализа данных делятся на два основных направления:

- сравнение с тем или иным видом заранее созданного шаблона,
- подход на основе машинного обучения и последующего распознавания.

Методы на основе машинного обучения в последнее время получили бурное развития. Одной из причин этого развития является доступность большого числа разнообразных данных для обучения (социальные сети, крупные БД, результаты поисковиков и т.д.). Эта группа методов на данный момент представлена большим числом алгоритмов: байесовские сети, деревья принятия решений, методы опорных векторов, методы k-средних и др. Данные методы, в свою очередь делятся на группы по методу обучения [58], который применяется для их конфигурирования:

- классификация (обучение с учителем),
- кластеризация (обучение без учителя),
- ассоциирование (association),
- численное предсказание (numeric prediction).

4.3.2 Методы на основе сигнатур

Недостатком этих методов является их высокая ресурсоёмкость, связанная с необходимостью просмотра больших объёмов данных. Однако в настоящее время вычислительные мощности позволяют использовать более точные, чем основанные на выводе, сигнатурные методы, которые, в свою очередь, делятся на две большие группы:

- поиск строк (string matching)
- поиск регулярных выражений (regexp matching).

Сигнатуры на основе строк.

В процессе развития, для поиска строк применялось большое число различных алгоритмов поиска строк, обладающих различными преимуществами и недостатками, что определяло область их применения [59,60]. Наиболее известными алгоритмами являются: прямой перебор (brute force, BF), Кнут-Морис-Пратт (KMP), Бойер-Мур (BM), Ахо-Корасик (AC), AC-BM (использующийся в Snort), Wu-Manber, Commentz Walter (CW), фильтры Блума (вероятностная структура на основе хеша).

В работе [61] проводится обзор и сравнение большого числа методов поиска строк по тому как реализован алгоритм сравнения с имеющимися сигнатурами. Выделено 4 группы методов.

- Последовательное сравнение со всеми сигнатурами (Exhaustive Search).
- Дерево сравнений (Decision Tree).
- Декомпозиция (Decomposition), при которой отдельные части сигнатура обрабатываются независимо, с последующим объединением результатов.
- Ассоциативный доступ (Tuple Space), при котором сигнатуры разбиваются на группы бит, с которыми проводятся операции сравнения.

На рис. 10 приведено распределение большого числа алгоритмов по данным группам. Алгоритмы, лежащие на границах, используют гибридные подходы.

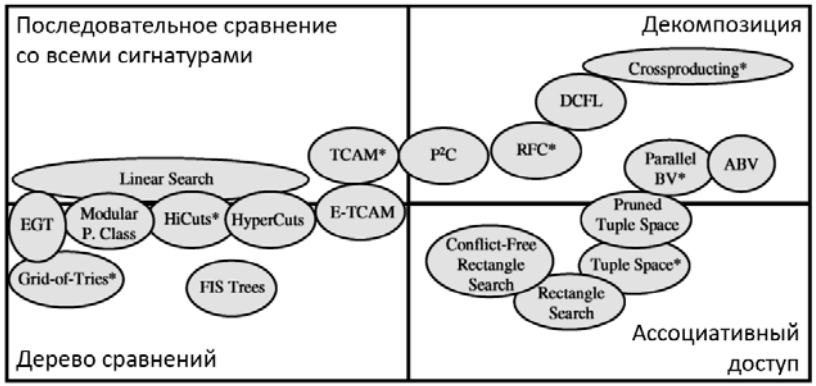

Рис. 10 - Распределение алгоритмов поиска строковых сигнатур по данным группам.

Сигнатуры на основе регулярных выражений.

С ростом числа протоколов и их сложности строковое представление было признано недостаточно выразительным, в связи с чем, для описания сигнатур стали использовать регулярные языки в виде грамматик и регулярных выражений. Для эффективного поиска сигнатур регулярный язык, описывающий сигнатуру, представляются в форме конечного автомата. Выделяют два основных вида автоматов - детерминированные или недетерминированные. Оба эти представления имеют свои достоинства и недостатки.

Одна из открытых баз сигнатур такого вида используется в открытом приложении для классификации l7-filter[62]. Кроме того, такие подходы могут не срабатывать в случае, если сигнатура была разделена на несколько пакетов на уровне IP или TCP. Для решения этой проблемы, перед поиском сигнатуры необходимо выполнить IP-дефрагментацию и TCP-нормализацию соответственно.

Основным достоинством недетерминированных конечных автоматов (НКА, NFA) является их компактность: объем занимаемой памяти пропорционален числу символов, входящих в регулярные выражения. Однако для обработки каждого символа входных данных недетерминированным конечным автоматам может потребоваться до O(N) обращений к памяти, где N – число состояний автомата [63]. По этой причине возможности применения НКА в высоконагруженных системах ограничены.

В свою очередь, детерминированные конечные автоматы (ДКА, DFA) требуют для каждого входного символа совершить единственное обращение к памяти. Их использование может представлять трудности в связи с их большим

размером: число состояний ДКА может экспоненциально расти («экспоненциальный взрыв»), и ограничено $O(2^l)$, где l – суммарная длина регулярных выражений в каноническом представлении. В работе [38] было проведено исследование влияния разных типов регулярных выражений на рост размеров автомата. Результаты показаны на рис. 11. Было выделено 3 типа регулярных выражений, с точки зрения их влияния на размер автомата:

- выражения, привязанные к началу пакета (поиск осуществляется только в начале пакета);
- выражения, привязанные к началу пакета и содержащие звёздочку Клини (*);
- выражения, не привязанные к началу и содержащие звёздочку Клини (*).

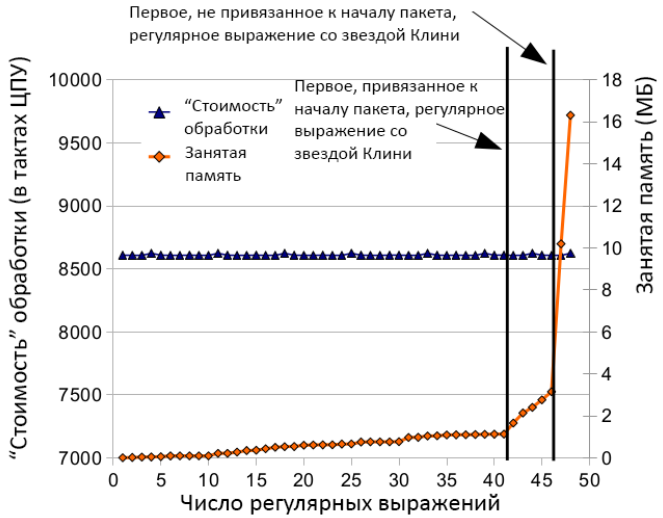

Рис. 11 - Экспоненциальный взрыв размера DFA при добавлении регулярных выражений со звёздочкой Клини.

Для снижения размеров автоматов часто применяют различные виды сжатия. Такие автоматы носят название сжатые ДКА (Compressed DFA, cDFA). В табл. 2 приведено сравнение трёх основных видов автоматов по размеру и производительности поиска, взятое из работы [38]. Данные автоматы были построены по регулярным выражениям классификатора L7[63]. Для предотвращения экспоненциального роста размера, детерминированные автоматы были разделены на 4 части.

Таблица 2 - Сравнение размеров и скорости работы основных видов конечных автоматов.

Алгоритм	Стоимость в тактах ЦПУ (мин, ср., макс.)	Количество автоматов	Размер автомата
НКА	$2.2*10^4$, $4.1*10^7$, $8.9*10^7$	1	509 Кб
ДКА	52, $2.5*10^4$, $3.6*10^4$	4	230 Мб
Сжатый ДКА	268, $1.2*10^5$, $1.7*10^5$	4	53 Мб

Несмотря на проблемы с требованиями к памяти, детерминированные конечные автоматы (и их модификации) получили намного большее распространение в высокоскоростных системах анализа.

Современные системы анализа трафика предъявляют высокие требования, как к скорости обработки данных, так и к количеству регулярных выражений, задействованных в обработке и, соответственно, размеру итогового автомата. Так как ни ДКА, ни НКА не могут удовлетворить одновременно требования и по скорости, и по размеру памяти, в настоящее время ведется большое количество исследований по разработке гибридных представлений. С точки зрения реализации, автоматы представляют собой таблицы из состояний, в каждой ячейке которых находится список возможных переходов из этого состояния в другое. Поэтому два основных направления работ сосредоточены на уменьшении числа состояний и переходов соответственно. В качестве примеров, можно привести представления D^2FA [64] и δFA [65], реализующие сжатие переходов и группу представлений MDFA [66], H-FA [67], XFA [68] и Dual FA [69], реализующих различные подходы к сокращению числа состояний.

4.3.3 Анализ данных в разных представлениях

Одну из важных проблем для классификаторов на основе содержимого представляет тот факт, что одни и те же данные (например, строка) могут быть при передаче по сети быть закодированы по-разному, в зависимости, например, от используемого протокола. В частности, под «различными представлениями» в данном разделе имеются следующие аспекты.

- Различные методы кодировки, в частности для текстовых данных — ASCII и Unicode кодировки, а для бинарных данных — различные транспортные кодировки, например представление в виде текста (binary-to-text), примером которых является Base64.
- Сжатия данных для уменьшения загруженности каналов передачи данных, например использования gzip и deflate алгоритмов для сжатия содержимого HTTP-сообщении.
- Шифрование данных для обеспечения безопасности, например использование криптографических алгоритмов RC4 и AES в протоколах SSL/TLS.

По данным различных исследований, сжатый и шифрованный трафик (иногда используется общий термин «непрозрачный», opaque) составляет всё большую долю от всех сетевых потоков данных [54]. Это является следствием большого числа факторов, таких как:

- рост популярности онлайн видеосервисов, использующие сжатие видеопотоков,
- распространённость P2P-сервисов, которые в большинстве своём используют шифрование,
- использование шифрованного соединения (HTTPS) по-умолчанию на многих популярных сайтах,
- внедрение сжатия в HTTP протоколе на многих Web-серверах.

Проблема классификации этих видов трафика имеет несколько аспектов.

- Для корректной классификации такого трафика требуется дополнительный функционал.
- Попытка классифицировать такой трафик «в лоб» существенно снижает общую производительность классификатора, так как приходится просматривать все данные пакетов, проходя по большой части автомата и при этом результат почти наверняка будет отрицательным. То есть такой трафик представляет собой «худший случай», характеристики работы на котором алгоритмов классификации существенно хуже средних (см. табл.2).

Для решения первого аспекта проблемы используются несколько подходов:

- Генерация копий сигнатур, которые подвергаются различным видам сжатия и кодирования. Данный метод ограничен только некоторыми алгоритмами сжатия и кодирования, а также плохо масштабируется с учётом роста количества алгоритмов сжатия и их количества их параметров.
- Использование модулей, осуществляющих разжатие/перекодировку данных перед их классификацией. Этот метод имеет такие же ограничения, как и предыдущий и также плохо масштабируется. Кроме того этот метод увеличивает уязвимость системы к атакам типа zip bomb [70], при которых размер разжимаемых данных превосходит размер сжатых на несколько порядков.
- Установка системы анализа на месте или после средства, осуществляющего разжатие/расшифрование данных. Пример такого средства - прокси-сервер.

Для устранения второго аспекта, требуется подавать на модуль классификации трафика только «прозрачный» трафик, для чего из всего трафика требуется предварительно отфильтровать «непрозрачную» его часть. Для решения этой задачи разработаны алгоритмы, большая часть которых использует характерное свойство «непрозрачного» трафика — повышенную энтропию

значений его отдельных байт. Примеры таких алгоритмов приведены в работах [71].

4.3.4 Классификация угроз

В реализациях DPI, связанных с безопасностью (например, IDS/IPS), где классификация применяется не для идентификации протоколов, а для классификации атак и угроз, разработаны подходы, специализированные под соответствующие задачи. Одним из таких подходов является статистическое выявление аномалий, когда вначале производится обучение системы на трафике, не содержащем атак («нормальном»), а затем на реальном трафике детектируются отклонения от «нормальной» картины. Такие подходы называют «статистическое детектирование аномалий» (statistical anomaly-based detection). На основе этой техники работают многие IDS и защиты от DDoS атак.

5. Требования, предъявляемые к современным средствам анализа содержимого сетевого трафика

Принимая во внимание приведённый обзор основных алгоритмов и схем анализа сетевого трафика можно сформулировать ряд функциональных и нефункциональных требований, предъявляемых к современным системам анализа сетевого трафика. Все требования можно разделить по подсистемам, к которым они применяются и отдельно выделить те, которые относятся ко всей системе в целом:

1. Система в целом.
   - Поддержка масштабирования по пропускной способности анализируемого канала передачи данных.
   - Минимизация числа перестановок пакетов в рамках отдельных потоков.
   - Возможность встраивания дополнительных средств предобработки сетевых пакетов перед их передачей подсистеме классификации (перекодировка, разжатие).
2. Подсистема перехвата данных.
   - Поддержка разбора всех протоколов ниже сетевого уровня, встречающихся в контролируемом канале (MPLS, VLAN и т.д.) Это необходимо, для обеспечения попадания всех пакетов одного потока в одну очередь обработки при выполнении балансировки нагрузки (хеширование должно выполняться на уровне IP-пакета).
   - Использование кольцевого буфера для хранения обрабатываемых пакетов и режима zero-copy, при наличии поддержки со стороны сетевой карты, или 1-copy, при отсутствии такой поддержки для экономии ресурсов центрального процессора.

   - Для эффективного использования ресурсов многопроцессорных и многоядерных машин требуется поддержка того или иного вида RSS-технологии (управления прерываниями и их распределения по разным ядрам).
3. Подсистема агрегации пакетов в потоки.
   - Поддержка возможности задания типа ключевой информации, по которой определяется принадлежность пакета к потоку, для обеспечения гибкости при использовании подсистемы для решения различных прикладных задач.
   - Максимизация количества одновременно обрабатываемых потоков и времени жизни каждого отдельного потока в условиях ограниченных ресурсов памяти.
   - Для обработки сжатых данных необходима возможность одновременного отслеживания потока, который представлен как в сжатом, так и в разжатом виде.
   - Встроенная защита от атак типа «zip-бомба».
   - Отслеживание факта связанности потоков (например, потока управления и потока данных в случае FTP), в частности, для уточнения классификации.
4. Подсистема классификации.
   - Сложность алгоритма поиска сигнатур должна быть не хуже чем линейной по входным данным «в среднем», а желательно и «в худшем» случае для устойчивости системы к целенаправленным атакам.
   - Расширяемый набор «сигнатур» для поддержки новых протоколов, их групп и сетевых приложений.
   - Хорошая масштабируемость по памяти при росте количества «сигнатур».
   - Возможность предварительного разделения трафика на «прозрачный» и «непрозрачный» с целью снижения нагрузки на данную подсистему.
   - Возможность анализа данных, представленных в различных кодировках.

Прежде чем перейти к рассмотрению конкретных реализаций отдельных компонентов систем анализа и того, насколько они удовлетворяют перечисленным выше требованиям, будут рассмотрены существующие способы подключения этих систем к сетям передачи данных и влияние конкретных видов подключения на точность работы систем анализа и ограничение применимости этих систем.

6. Классификация систем анализа по способу подключения к сети передачи данных

Одна из важных характеристик систем анализа сетевого трафика — способ получения и конкретный вид данных для анализа, то есть вопрос подключения к некоторому «каналу» связи. С точки зрения особенностей подключения, подсистемы получения данных можно разделить на следующие классы:

1. Распределённые системы**.** Схема представляет собой набор сборщиков данных о сетевом трафике (probes) и его накопителях и набор его анализаторов (collectors), которые получают данные от сборщиков. При такой схеме наиболее актуальным становится вопрос знания топологии сети и точек получения трафика в рамках этой топологии системой анализа и учёт этих знаний при выполнении анализа. К таким системам относятся системы:
   - системы пассивного сетвого мониторинга (FlowMon, Ntop),
   - системы анализа поведения (Network Behaviour Analysys, NBA),
   - системы обнаружения аномального поведения (Network Behaviour Anomaly Detection, NBAD),
   - системы управления событиями информационной безопасности (Security Information and Event Managment, SEM, SIM, SIEM).
2. Беспроводные системы, в том числе мобильные. Особенностью данных систем является то, что «канал» типа точка-точка отсутствует и при достаточном уровне сигнала можно перехватывать беспроводные коммуникации в достаточно большом радиусе. Названия систем анализа, подключаемых таким образом, обычно начинаются с префикса «wireless» — например Wireless firewall и Wireless intrusion prevention system (WIPS).
3. Локальные системы. В данном классе подключение осуществляется к конкретному каналу передачи данных (сетевому кабелю). В зависимости от места сетевого канала, к которому осуществляется подключение, в общей топологии сети можно выделить следующие подклассы:
   - Конечный пользователь — подключение осуществляется на уровне сетевой карты конкретного пользователя. Названия систем анализа, подключаемых таким образом, обычно начинаются с префикса «host-based» — например host-based application firewall и host-based intrusion prevention system (HIPS). В свою очередь сам перехват данных в этом случае может осуществляться как на уровне сетевого стека с помощью соответствующего низкоуровнего API (PF_RING, NAPI), так и путём перехвата (hook) системных вызовов. Последний вариант является более медленным и используется в т.н. software application firewalls, к которым относятся большинство встроенных в ОС межсетевых экранов. Их отличием является то, что анализ

выполняется не по потокам, а по процессам, которые участвуют в сетевом обмене. Минусом таких решений является их уязвимость из-за неполной изоляции процессов и возможности переполнения памяти. На данный момент более продвинутой технологией является использование «песочниц» (sandbox) с дополнительной изоляцией процессов, к котором относятся например AppArmor и TrustedBSD MAC framework.

   - Шлюз — подключение осуществляется в точке, которая является единственным выходом в глобальную сеть (WAN) для некоторой локальной подсети (LAN). В случае установки на одном из нескольких шлюзов может наблюдаться ситуация с «односторонними потоками», когда, например, исходящий трафик некоторого соединения идёт через один шлюз, а входящий — через другой. Подобная ситуация, называется «сетевой ассиметрией» (network assymetry)[72] и также может иметь место при работе со спутниковыми каналами связи (см. рис. 12). Такие условия значительно повышают требования к системе анализа и применяемым алгоритмам (в частности усложняется TCP-нормалиация в отсутствие ACK-пакетов). Названия систем анализа, подключаемых таким образом, обычно начинаются с префикса «network-based» — например network-based application firewall и network-based intrusion prevention system (NIPS). Подключение при этом может выполнено по одной из трёх схем:
     - Зеркалирование (mirroring), при котором весь сетевой поток канала дублируется — первая копия идёт непосредственно в сеть, а вторая отправляется на вход системы анализа (см. рис 7, левая часть). При такой схеме в отсутствие обратной связи от системы анализа может выполняться только пассивный анализ, то есть система не может влиять на трафик, попадающий в сеть (например, путём фильтрации или приоритезации). Само зеркалирование также может технически выполняться тремя способами:
       - Кабельный сплиттер или сплиттер волокна, в случае оптики, позволяет дублировать пакеты, без какого либо влияния на них (в том числе и на задержку).
       - Пассивное оборудование, выполняющее т.н. зеркалирование портов (port mirroring) при котором данные с одного (или более) входных портов копируются на несколько выходных портов. Данный метод может вносить незначительные задержки. Также могут возникать потери пакетов из-за недостаточной производительности оборудования.
       - Активное оборудование, также выполняющее зеркалирование портов, но также реализующее некоторое решение, выполняющее анализ трафика, например

межсетевой экран или приоритезацию. Основной недостаток — вносимая задержка.

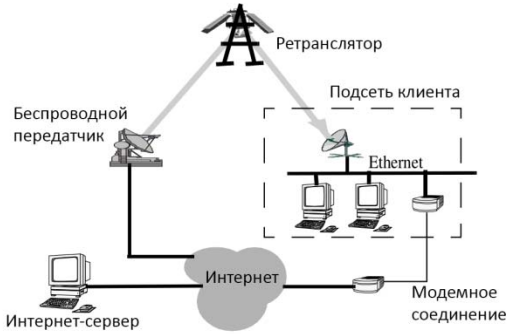

Рис. 12 - Пример «ассиметричных соединений» при использовании спутниковых каналов связи.

- Проксирование (proxy), при котором весь поток направляется на систему, которая становится посредником во всех взаимодействиях WAN и LAN (см. рис. 2). Плюсом такой схемы является «прозрачность» (transparency) трафика для системы анализа — например, система может проводить декомпрессию и расшифрование сжатого и шифрованного трафика, соответственно с последующей компрессией и отправкой получателю. Минусом является большая ресурсоёмкость проксирование и сравнительно большая задержка в доставке сетевых пакетов, вносимая анализом. Также такое подключение вносит дополнительные риски — при выходе системы анализы из строя в результате поломки или целенаправленной сетевой атаки (см. [73]) локальная сеть остаётся без связи с глобальной.
- Байпас (bypass). Общий вид такого подключения показан на рис. 6. Гибридный подход между зеркалированием и проксированием, решающий проблему выхода из строя системы анализа. В нормальных условиях система работает в режиме проксирования, однако при её выходе из строя или получении от неё соответствующего сигнала, соединение с LAN выполняется напрямую. На данный момент такой вид подключения наиболее распространён для решений, которым требуется активный анализ, то есть возможность изменять содержимое передаваемого трафика в соответствии с некоторыми политиками.

В следующем разделе будут подробно рассмотрены особенности реализации программных и программно-аппаратных систем анализа сетевого трафика.

7. Классификация высокоскоростных средств анализа содержимого сетевого трафика

Основные виды обрабатывающих элементов приведены на рис. 13.

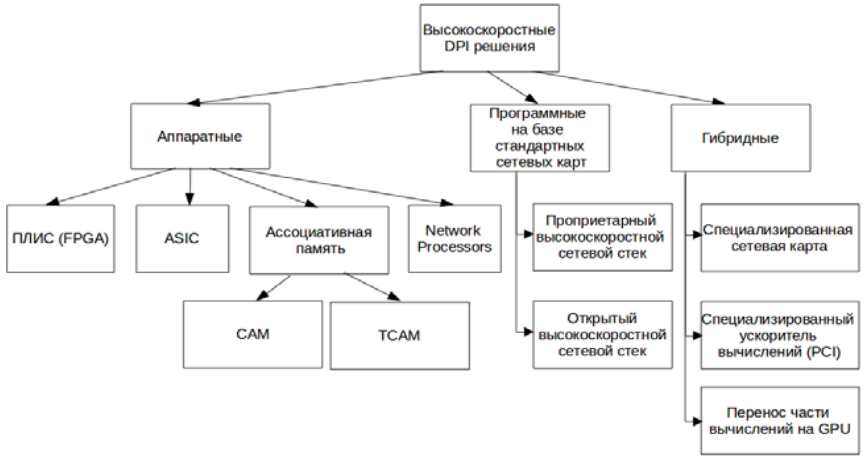

Рис. 13 - Классификация основных видов высокоскоростных DPI-систем.

Как видно из рисунка, все решения можно разделить на три большие группы.

1. Специализированные аппаратные решения, представляющие собой «чёрный ящик», являющийся, как правило, системой на чипе (System-on-a-Chip, SoC). Такое решение может содержать самые различные компоненты, позволяющие ускорить анализ сетевого трафика — ПЛИС (Field-Programmable Gate Array, FPGA), интегральные схемы специального назначения (Application-Specific Integrated Circuit, ASIC), сетевые процессоры (Network Processors, NP), бинарную (content addressable memory, CAM) и троичную (ternary content addressable memory, TCAM) ассоциативную память или их комбинации, например [74]. Особенности конкретных аппаратных систем следующие.

   - ASIC — одно из наиболее высокоскоростных решений. Минусом является то, что вся программа обработки, включая сигнатуры, закладывается в устройство на этапе производства и впоследствии не может быть изменена.
   - FPGA — устройства с достаточно высокой пропускной способностью, которые могут перепрограммироваться при необходимости изменить алгоритм обработки или набор сигнатур, хотя процесс обновления прошивки может занимать значительное время. Данные устройства достаточно неудобны в непосредственном программировании, поэтому существует некоторое количество промежуточных представлений и средств

трансляции/компиляции, генерирующих на выходе FPGA-программу. К их числу можно отнести, прежде всего, OpenCL [75]и более высокоуровневые среды: NetCOPE от InveaTech [76] и G от NetFPGA [77]. При реализации классификации на FPGA устройствах обычно используют алгоритм KMP.

- CAM — специальный вид ассоциативной памяти, который выполняет параллельное сравнение всего своего содержимого с поступившим на вход значением и возвращает адрес, значение которого совпало с входным. Скорость доступа достаточно высока и составляет 4 нс, а сложность поиска составляет *O(1)*. Однако данный вид памяти не может выполнять поиск наибольшего префикса, что существенно для большинства решений DPI, поэтому она подходит только для поиска строк фиксированной длины. Существуют различные реализации CAM, в том числе bitwise CAM (BCAM) — ассоциативная память на основе дерева, в которой строки представляются в виде булевских формул [59].
- TCAM — также специальный вид ассоциативной памяти, который помимо данных хранит три вида логических значений (0, 1 и ? - «не важно»). Также, в случае сигнатур, они упорядочены по убыванию. В результате над данной памятью можно эффективно проводить операцию поиска наибольшего префикса. Данная память получила широкое распространение в сетевых устройствах, в частности на её основе в роутерах и свитчах выполняется трансляция между именами и IP-адресами (IP address lookup или DNS lookup). Пример работы этого вида памяти приведён на рис. 14. Несмотря на общую эффективность, данный вид памяти имеет ряд недостатков:
  - высокая стоимость (в десятки раз дороже SRAM);
  - неэффективность хранения (меньшая ёмкость);
  - большое энергопотребление (до 180 раз по сравнению с SRAM);
  - плохая масштабируемость на длинные входные данные.
- На данный момент существуют реализации, эмулирующие функционал TCAM на SRAM памяти E-CAM, Z-CAM и др. (полный обзор таких реализаций приведён в [78]), преодолевающие часть из перечисленных недостатков ценой незначительного снижения производительности.
- Также ряд недостатков связан напрямую с применением в DPI решениях.
  - Т.н. «Range Representation Problem» - в TCAM легко хранить префикс, который требуется искать только в начале «слов», однако чтобы эффективно искать тот же префикс в середине «слов» требуется гораздо больше ячеек TCAM.
  - Т.н. «Multi-match Classification Problem» - в результате сравнения возвращаются все ячейки, с которыми сравнение дало

положительный результат, а не только более приоритетные, что создаёт дополнительную вычислительную нагрузку.

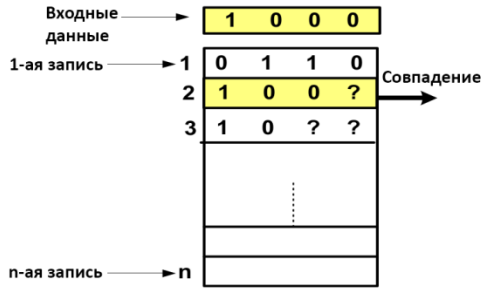

Рис. 14 - Пример работы TCAM-памяти.

- NP — специализированный вид процессоров для обработки сетевого трафика. Основные функции включают: поиск шаблонов, поиск по ключам, обработка полей протоколов, управление очередями, быстрые операции выделения и изменения буферов памяти. Архитектура использует конвейерную и суперскалярную обработку, многоядерность, специализированный набор инструкций (микрокод). Примером может служить IXP1200, использовавшийся для реализации первой IDS CardGuard на базе сетевых процессоров [79].

2. Чисто программные решения, использующие в качестве аппаратной платформы, стандартные высокопроизводительные (обычно-многопроцессорные) серверы со стандартными серверными сетевыми картами. Для обеспечения полноты перехвата на высоких скоростях используются:
   - Проприетарные разработки, использующие высокопроизводительные реализации сетевого стека, например Sniffer10G в случае Emulex и Myricom.
   - Открытые разработки, такие как Intel DPDK и Ntop PF_RING (Intel, Mellanox, Chelsio, Qlogic\Broadcom).
3. Программно-аппаратные решения, которые с одной стороны, также как и чисто программные решения, используют в качестве основного «вычислителя» стандартную серверную платформу, но некоторые вычисления реализуют на специализированных устройствах. К таким устройствам относятся.
   - Сетевые карты на базе технологий FPGA (Napatech, Endace, InveaTech, Accolade, Fiberblaze, Telesoft Technologies), ASIC (titanic systems, Hitech global) или многопроцессорных систем (Tillera).

- Специализированные вычислители на базе технологий FPGA, устанавливаемые в PCI-слот.
- Использование GPU-карт для отдельных видов вычислений. Известны решения для задачи маршрутизации, т.н. Software Routers [80] и для задачи классификации по регулярным выражениям [81]. Основой проблемой при использовании GPU, является «узкое горлышко» при передаче данных из основной памяти в память GPU для обработки. Для решения этой проблемы со стороны производителей GPU были предложены решения на основе прямого отображения памяти компьютера в память GPU с использованием DMA, что позволяет освободить ресурсы CPU от обработки операций копирования. У разных производителей эти технологии имеют разные названия — GPUDirect у Nvidia [82] и DirectGMA у AMD [83]. Известны примеры использования этих технологий для ускорения обработки сетевого трафика [84]. Примерная схема работа данных технологий приведена на рис. 15 справа, по сравнению с режимом по умолчанию (слева).

Рис. 15 - Схема потока сетевого трафика без применения (слева) и с применением (справа) технологий GPUDirect и DirectGMA.

8. Масштабирование системы анализа

Важным практическим вопросом при использовании любого средства анализа трафика является вопрос масштабирования при увеличении ширины анализируемого канала. Распространённый способ решения - добавление в схему ещё одного устройства обработки трафика. Однако при этом также требуется решать задачу распределения сетевых пакетов из исходного канала по устройствам обработки трафика. Для решения этой задачи в случае широких исходных каналов используются специальные сетевые устройства — пакетные брокеры или балансировщики, основной задачей которых является распределение сетевых пакетов, полученных из набора входных сетевых интерфейсов по набору выходных сетевых интерфейсов. В зависимости от логики, применяемой при этом выборе, можно выделить статическую и динамическую балансировку.

При статической балансировке стандартными схемам коммутации являются:

- One-to-One – каждый входной порт отображается на отдельный выходной;
- Any-to-Any - любой входной порт на любой выходной;
- Many-to-One - агрегация данных с нескольких входных портов в один выходной;
- One-to-Many - зеркалирование трафика на несколько потребителей;

При динамической балансировке, важной функциональностью является возможность разбора заголовков различных протоколов низкого уровня (таких как Ethernet, VLAN, MPLS и др.). Это важно прежде всего для того чтобы была возможность передавать пакеты, относящиеся к одному потоку транспортного уровня на один сетевой интерфейс, то есть к одному устройству обработки трафика. В частности, это позволяет корректно выполнять TCP-нормализацию. Также это позволяет уменьшить эффект перестановки пакетов в рамках одного потока, что, в свою очередь, повышает эффективность алгоритмов IP-дефрагментации и TCP-нормализации. Наличие такого функционала говорит о том, что сами балансировщики являются аппаратным средством сетевого анализа. Кроме того, многие из них также поддерживают аппаратную фильтрацию по содержимому пакетов (обычно поиск осуществляется по префиксу данных пакетов некоторой фиксированной длины), что говорит о том, что балансировщики потенциально относятся к DPI решениям.

Пропускная способность пакетных брокеров варьируется от десятков до сотен гигабит в секунду. Среди производителей пакетных брокеров можно упомянуть cPacket, netoptics, arista.

9. Выводы

Из приведённого обзора можно сделать ряд выводов. Можно констатировать как количественный (в связи с ростами объёмов трафика и ширины каналов связи), так и качественный рост (в связи с новыми прикладными задачами) потребностей в средствах анализа трафика. При этом, несмотря на огромное многообразие конкретных решений, реализующих различные виды анализа, в основе большинства этих решений лежит примерно одинаковая схема, что хорошо видно на примере внедрения концепции «DPI как сервис». В вопросах реализации систем анализа можно отметить следующий ряд тенденций:

- Развитие специализированных аппаратных средств (NP, TCAM и т.д.), позволяющих обрабатывать потоки данных в каналах максимально достижимой пропускной способности.
- Развитие программных и программно-аппаратных технологий (NAPI, RSS, MSI-X, DCA и т.д.), позволяющих обрабатывать потоки данных порядка 10 Гбит/с на стандартных серверных платформах.

- Перенос (offloading) значительной части сетевой обработки непосредственно на сетевые карты.
- Появление специализированных сетевых стеков (в том числе с открытым исходным кодом), позволяющих осуществлять перехват без потерь на каналах 10 и более Гбит/с. Это, в свою очередь, позволяет реализовывать достаточно мощные сетевые анализаторы на базе стандартных компонент, без использования дорогостоящих специализированных сетевых карт на базе FPGA и ASIC.
- Активные исследование в области переноса задачи классификации сетевого трафика на GPU в связи с её высокой ресурсоёмкостью и ограниченным количеством вычислительных ресурсов центрального процессора, даже на многопроцессорных системах.

Также можно видеть, что для каждого элемента общей схемы анализа в научном, техническом и IT-сообществах осуществляется поиск оптимальных решений под конкретные прикладные задачи. В научном сообществе решаются задачи поиска оптимальных алгоритмов, например для решения задачи классификации, являющейся наиболее ресурсоёмкой частью любой системы анализа. В техническом сообществе осуществляется разработка аппаратных средств, позволяющих обеспечить возможность решения прикладных задач на каналах с постоянно растущей пропускной способностью. В IT-сообществе осуществляется синтез решений, предлагаемых двумя другими сообществами, подбор конкретных параметров алгоритмов для отдельных прикладных задач и поиск баланса, который сводится, по сути, к решению оптимизационных задач в условиях большого числа переменных, среди которых можно упомянуть:

- необходимую полноту анализа;
- необходимую точность анализа;
- требуемую глубину анализа;
- цена конкретного программно-аппаратного решения;
- производительность этого решения;
- гибкость решения и его возможности по масштабированию и наращиванию функционала.

Список литературы

[1]. Sniffer. https://www.opennet.ru/base/sec/arp_snif.txt.html, дата обращения 01.12.2015.
[2]. Wireshark. https://www.wireshark.org/, дата обращения 01.12.2015.
[3]. NAT. https://tools.ietf.org/html/rfc1918, дата обращения 01.12.2015.
[4]. Software NAT. http://www.nat32.com/v2/, дата обращения 01.12.2015.
[5]. DDoS. http://ddos-protection.ru/chto-takoe-ddos, дата обращения 01.12.2015.
[6]. IANA Service Name and Transport Protocol Port Number Registry. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, дата обращения 01.12.2015.

[7]. Д. Виньяр. Deep Packet inspection: Technology and products. Семинар «DPI-технологии: архитектура и опыт», «Петер-Сервис», 3.12.2013.
[8]. SIP. https://www.ietf.org/rfc/rfc3261.txt, дата обращения 01.12.2015.
[9]. RTP. https://www.ietf.org/rfc/rfc3550.txt, дата обращения 01.12.2015.
[10]. Session Border Controller. http://www.voip-info.org/wiki/view/Session+Border+Controller, дата обращения 01.12.2015.
[11]. Wan Optimizations. http://searchenterprisewan.techtarget.com/definition/WAN-optimization, дата обращения 01.12.2015.
[12]. CIFS. https://msdn.microsoft.com/en-us/library/ee442092.aspx, дата обращения 01.12.2015.
[13]. MiddleBoxes. https://tools.ietf.org/html/rfc3234, дата обращения 01.12.2015.
[14]. DPI. http://nag.ru/articles/article/22432/dpi.html, дата обращения 01.12.2015.
[15]. Экономика программных и аппаратных DPI на примере Cisco SCE и СКАТ. http://nag.ru/articles/article/28436/ekonomika-programmnyih-i-apparatnyih-dpi-na-primere-cisco-sce-i-skat.html, дата обращения 01.12.2015.
[16]. Платформы глубокого анализа трафика и управления трафиком приложений. http://www.inlinetelecom.ru/solutions/access_network/platform_depth_analysis_of_traffic_and_traffic_control_applications/, дата обращения 01.12.2015.
[17]. David L. Cannon. CISA Certified Information Systems Auditor Study Guide, 2nd Edition, 2008, ISBN: 978-0-470-23152-4
[18]. ICAP. https://tools.ietf.org/html/rfc3507, дата обращения 01.12.2015.
[19]. Сергей Медведев. Deep Packet Inspection (DPI). Семинар «Живые встречи», Красноярск, 18.01.2014.
[20]. Рекомендация МСЭ-Т Y.2770, Требования к углубленной проверке пакетов в сетях последующих поколений, издание 1.0, 20.11.2012
[21]. Рекомендация МСЭ-Т Y.2771, Структура углубленной проверки пакетов, 01.07.2014
[22]. QUIC. https://tools.ietf.org/html/draft-tsvwg-quic-protocol-00, дата обращения 01.12.2015.
[23]. CAIDA Flow Types. https://www.caida.org/research/traffic-analysis/flowtypes/, дата обращения 01.12.2015.
[24]. П. Филимонов, М. Иванов. Современные подходы к классификации трафика физических каналов сети Интернет, Труды 18-ой Международной конференции «Распределенные компьютерные и коммуникационные сети: управление, вычисление, связь» (DCCN-2015), 19 - 22 октября 2015 г, стр. 466-474
[25]. F. Risso, M. Baldi, O. Morandi, A. Baldini, P. Monclus, “Lightweight, payload-based traffic classification: An experimental evaluation” in Proc. IEEE ICC, 2008, pp. 5869–5875.
[26]. Wireshark, VoIP calls. https://wiki.wireshark.org/VoIP_calls, дата обращения 01.12.2015.
[27]. MIME. https://tools.ietf.org/html/rfc2045, дата обращения 01.12.2015.
[28]. ASN.1. https://tools.ietf.org/html/rfc6025, дата обращения 01.12.2015.
[29]. P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker. P4: Programming protocol-independent packet processors. SIGCOMM Computer Communications Review, 2013.
[30]. A. Bremler-Barr, Y. Harchol, D. Hay, Y. Koral. Deep packet inspection as a service. In CoNEXT, pages 271–282, 2014.
[31]. SDN. https://tools.ietf.org/html/rfc7426, дата обращения 01.12.2015.

[32]. Qosmos ixEngine. http://www.qosmos.com/products/deep-packet-inspection-engine/, дата обращения 01.12.2015.
[33]. Ipoque PACE. https://www.ipoque.com/products/pace, дата обращения 01.12.2015.
[34]. Windriver Content Inspection Engine. http://www.windriver.com/products/product-overviews/PO_Wind-River-Content-Inspection-Engine.pdf, дата обращения 01.12.2015.
[35]. Procera PacketLogic Content Intelligence. https://www.proceranetworks.com/content-intelligence.html, дата обращения 01.12.2015.
[36]. Cisco NBAR. http://www.cisco.com/c/en/us/products/ios-nx-os-software/network-based-application-recognition-nbar/index.html, дата обращения 01.12.2015.
[37]. Junos OS Next Generation Application Identification. https://www.juniper.net/documentation/en_US/junos15.1x49/topics/concept/services-application-identification-techniques-understanding.html, дата обращения 01.12.2015.
[38]. Cascarano N, Ciminiera L, Risso F. Optimizing deep packet inspection for high-speed traffic analysis. Network System Manager. 2011; 19(1):7–31.
[39]. Duffield N., Lund C. “Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure”. ACM Internet Measurement Conference, 2003.
[40]. Callado A., Kamienski C., Szabo G., Gero B., Kelner J., Fernandes S., Sadok D. A Survey on Internet Traffic Identification; Communications Surveys & Tutorials, IEEE Volume 11, Issue 3, 3rd Quarter 2009 Page(s):37 – 52.
[41]. Duffield, N.; Lund, C.; Thorup, M., “Learn more, sample less: control of volume and variance in network measurement”, IEEE Transactions in Information Theory, vol. 51, no. 5, pp. 1756-1775, 2005.
[42]. PSAMP. http://www.rfc-editor.org/rfc/rfc5476.txt, дата обращения 01.12.2015.
[43]. Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju and James W. Hong, “The Architecture of NGMON: a Passive Network Monitoring System for High-Speed IP Networks”, Accepted to appear in the Proc. of the 13th IFIP/IEEE International Workshop on Distributed Systems:Operations & Management (DSOM 2002), Montreal, Canada, October 21-23, 2002.
[44]. InveaTech FlowMon. https://www.invea.com/en/products/flowmon, дата обращения 01.12.2015.
[45]. IPFIX. https://tools.ietf.org/html/rfc5101, дата обращения 01.12.2015.
[46]. M.-S. Kim, Y. J. Won, and J. W. Hong. Characteristic analysis of internet traffic from the perspective of flows. Comp. Comm., 29(10):1639–1652, 2006.
[47]. R. Sommer and A. Feldmann. NetFlow: Information loss or win? In ACM SIGCOMM Internet Meas. Workshop, 2002.
[48]. MIB. https://tools.ietf.org/html/rfc3418, дата обращения 01.12.2015.
[49]. SNMP. https://www.ietf.org/rfc/rfc1157.txt, дата обращения 01.12.2015.
[50]. Colin J. Bennett,Andrew Clement, Kate Milberry. Introduction to Cyber-Surveillance. Cyber-Surveillance in Everyday Life. Vol. 9, No 4 (2012)
[51]. T. Farah, and L. Trajkovic, "Anonym: A tool for anonymization of the Internet traffic." In IEEE 2013 International Conference on Cybernetics (CYBCONF), 2013, pp. 261-266.
[52]. F. Risso, and L. Degioanni, “An Architecture for High Performance Network Analysis”, Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001), Hammamet, Tunisia, July 2001.
[53]. PF_RING ZC. http://www.ntop.org/products/packet-capture/pf_ring/, дата обращения 01.12.2015.

[54]. Andrew M White, Srinivas Krishnan, Michael Bailey, Fabian Monrose, and Phillip Porras. Clear and Present Data: Opaque Traffic and its Security Implications for the Future. NDSS, 2013.
[55]. Network Intrusion Detection Signatures. http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-five, дата обращения 01.12.2015.
[56]. Yuji Waizumi, Yuya Tsukabe, Hiroshi Tsunoda, and Yoshiaki Nemoto. Network Application Identification Based on Communication Characteristics of Application Messages. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:3, No:12, 2009
[57]. T. Karagiannis, K. Papagiannaki and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In Proc. of ACM SIGCOMM, August 2005
[58]. Nguyen, T. T. T. and Armitage, G.2008. A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys &Tutorials 10, 4, (2008), 56-76.
[59]. T. AbuHmed, A. Mohaisen, and D. Nyang, “A survey on deep packet inspection for intrusion detection systems,” CoRR, vol. abs/0803.0037, 2008. [Online]. Available: http://arxiv.org/abs/0803.0037, дата обращения 01.12.2015.
[60]. Koloud Al-Khamaiseh, Shadi ALShagarin. A Survey of String Matching Algorithms. Int. Journal of Engineering Research and Applications. ISSN : 2248-9622, Vol. 4, Issue 7(Version 2), July 2014, pp.144-156.
[61]. D. E. Taylor, “Survey and taxonomy of packet classification techniques,” ACM Comput. Surv., vol. 37, no. 3, pp. 238–275, 2005.
[62]. L7-filter. http://l7-filter.sourceforge.net/, дата обращения 01.12.2015.
[63]. J. E. Hopcroft and J. D. Ullman, “Introduction to Automata Theory, Languages, and Computation,” Addison Wesley, 1979.
[64]. S. Kumar and P. Crowley. Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection. In Proc. of SIGCOMM, 2006.
[65]. D. Ficara, S. Giordano, G. Procissi. An Improved DFA for Fast Regular Expression Matching. In Proc. of SIGCOMM, 2008.
[66]. F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, and R. H. Katz. Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection. In Proc. of ANCS, 2006.
[67]. S. Kumar, B. Chandrasekaran, J. Turner, and G. Varghese. Curing Regular Expressions Matching Algorithms From Insomnia. In Proc. of ANCS, 2007.
[68]. R. Smith, C. Estan, S. Jha, and S. Kong. Deflating the Big Bang: Fast and Scalable Deep Packet Inspection with Extended Finite Automata. In Proc. of ACM SIGCOMM, 2008.
[69]. C. Liu, J. Wu. Fast Deep Packet Inspection with a Dual Finite Automata. IEEE Transactions on Computers, Vol. 62.
[70]. Zip Bomb. http://xeushack.com/zip-bomb/, дата обращения 01.12.2015.
[71]. J. Olivain and J. Goubault-Larrecq. Detecting subverted cryptographic protocols by entropy checking. Research Report LSV-06-13, Laboratoire Specification et Verification, ENS Cachan, France, June 2006.
[72]. H. Balakrishnan and V. Padmanabhan, ”How network asymmetry affects TCP”, in IEEE Communications Magazine,Vol. 39, pp. 60 -67, April 2001.
[73]. The Perils of Deep Packet Inspection. http://www.symantec.com/connect/articles/perils-deep-packet-inspection, дата обращения 01.12.2015.
[74]. A Hardware Platform for Network Intrusion Detection and Prevention. http://www.cc.gatech.edu/home/wenke/papers/np3.pdf, дата обращения 01.12.2015.

[75]. Jonathan Thompson, Kristofer Schlachter. An Introduction to the OpenCL Programming Model. Distributed Computing CSCI-GA.2631-001 (Multicore Programming), 2012
[76]. InveaTech NetCOPE. https://www.invea.com/en/products-and-services/fpga-development-kit, дата обращения 01.12.2015.
[77]. G NetFPGA. https://github.com/NetFPGA/netfpga/wiki/G, дата обращения 01.12.2015.
[78]. Shaly Laurence, Anuros Thomas K. Review of SRAM based architecture for TCAM. International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. II, Special Issue VI, February 2015.
[79]. Towards software-based signature detection for intrusion prevention on the network card. http://www.cs.vu.nl/~herbertb/papers/cardguard_raid05.pdf, дата обращения 01.12.2015.
[80]. S. Han, K. Jang, K. Park, and S. Moon, “Building a Single-Box 100 Gbps Software Router,” in IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), May 2010, pp. 1–4.
[81]. A. Feitoza Santos, S. F. de Lacerda Fernandes, P. Gomes Lopes Junior, D. Fawzi Hadj Sadok, and G. Szabo, “Multi-gigabit traffic identification on gpu,” in Proceedings of the First Edition Workshop on High Performance and Programmable Networking, ser. HPPN ’13. New York, NY, USA: ACM, 2013, pp. 39–44. [Online]. Available:http://doi.acm.org/10.1145/2465839.2465845
[82]. GPUDirect. https://developer.nvidia.com/gpudirect, дата обращения 01.12.2015.
[83]. DirectGMA. http://developer.amd.com/tools-and-sdks/graphics-development/firepro-sdk/firepro-directgma-sdk/, дата обращения 01.12.2015.
[84]. Utilizing GPUDirect 3RD Party DMA Features for 10GbE NIC and GPU applications. http://on-demand.gputechconf.com/gtc/2013/presentations/S3300-GPUDirect-DMA-Features.pdf, дата обращения 01.12.2015.

Wirespeed network traffic analysis: survey of applied problems, approaches and solutions*

A.I. Get'man, <thorin@ispras.ru>
E.F. Evstropov <john0606@yandex.ru>
Y.V. Markin <ustas@ispras.ru>
ISPRAN, 109004, Russia, Moscow,
A. Solzhenitsyna st., 25.

Annotation. This paper presents a survey of online network traffic analysis scientific research and particular hardware and software solutions. In this paper main analysis technologies progress directions are enumerated, and it is shown how they are used in solutions of applied problems. On the basis of vast survey of different applied problems common infrastructure components and algorithms are identified that most off-the-shelf solutions use in various combinations. These analysis components are sequentially considered with specification of methods applied, subproblems encountered, appropriate constraints. Through analysis of components and their interaction schemes requirement list for individual components and infrastructure as a whole is concluded. This is followed by examination of principal classes of existing analysis systems from the point of view of their way of connection to network and implementation features. The question of system scalability in response to network channel throughput increase is also considered.

Key words: network traffic analysis; DPI; loseless packet capture; traffic classification**.**

References

[1]. Sniffer. https://www.opennet.ru/base/sec/arp_snif.txt.html.
[2]. Wireshark. https://www.wireshark.org/.
[3]. NAT. https://tools.ietf.org/html/rfc1918.
[4]. Software NAT. http://www.nat32.com/v2/.
[5]. DDoS. http://ddos-protection.ru/chto-takoe-ddos.
[6]. IANA Service Name and Transport Protocol Port Number Registry. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.
[7]. Д. Виньяр. Deep Packet inspection: Technology and products. Workshop «DPI: architecture and experience архитектура и опыт», «Peter-Service », 3.12.2013.
[8]. SIP. https://www.ietf.org/rfc/rfc3261.txt.
[9]. RTP. https://www.ietf.org/rfc/rfc3550.txt.
[10]. Session Border Controller. http://www.voip-info.org/wiki/view/Session+Border+Controller.
[11]. Wan Optimizations. http://searchenterprisewan.techtarget.com/definition/WAN-optimization.

* This work is supported by grant RFBI 14-07-00606 A

[12]. CIFS. https://msdn.microsoft.com/en-us/library/ee442092.aspx.
[13]. MiddleBoxes. https://tools.ietf.org/html/rfc3234.
[14]. DPI. http://nag.ru/articles/article/22432/dpi.html.
[15]. Ekonomika programmnykh i apparatnykh DPI na primere Cisco SCE i SKAT http://nag.ru/articles/article/28436/ekonomika-programmnyih-i-apparatnyih-dpi-na-primere-cisco-sce-i-skat.html.
[16]. Platformy glubokogo analiza trafika i upravleniya trafikom prilozhenii. http://www.inlinetelecom.ru/solutions/access_network/platform_depth_analysis_of_traffic_and_traffic_control_applications/.
[17]. David L. Cannon. CISA Certified Information Systems Auditor Study Guide, 2nd Edition, 2008, ISBN: 978-0-470-23152-4
[18]. ICAP. https://tools.ietf.org/html/rfc3507.
[19]. Sergei Medvedev. Deep Packet Inspection (DPI). Seminar «Zhivye vstrechi», Krasnoyarsk, 18.01.2014.
[20]. Recommendation ITU-T Y.2770, Requirements for deep packet inspection in next generation networks, edition 1.0, 2012.11.20
[21]. Recommendation ITU-T Y.2771, Framework for deep packet inspection, 2014.07.01
[22]. QUIC. https://tools.ietf.org/html/draft-tsvwg-quic-protocol-00.
[23]. CAIDA Flow Types. https://www.caida.org/research/traffic-analysis/flowtypes/.
[24]. P. Filimonov, M. Ivanov. Sovremennye podkhody k klassifikatsii trafika fizicheskikh kanalov seti Internet, Trudy 18-oi Mezhdunarodnoi konferentsii «Raspredelennye komp'yuternye i kommunikatsionnye seti: upravlenie, vychislenie, svyaz'» (DCCN-2015), 19 - 22 oktyabrya 2015 g, str. 466-474.
[25]. F. Risso, M. Baldi, O. Morandi, A. Baldini, P. Monclus, “Lightweight, payload-based traffic classification: An experimental evaluation” in Proc. IEEE ICC, 2008, pp. 5869–5875.
[26]. Wireshark, VoIP calls. https://wiki.wireshark.org/VoIP_calls.
[27]. MIME. https://tools.ietf.org/html/rfc2045.
[28]. ASN.1. https://tools.ietf.org/html/rfc6025.
[29]. P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker. P4: Programming protocol-independent packet processors. SIGCOMM Computer Communications Review, 2013.
[30]. A. Bremler-Barr, Y. Harchol, D. Hay, Y. Koral. Deep packet inspection as a service. In CoNEXT, pages 271–282, 2014.
[31]. SDN. https://tools.ietf.org/html/rfc7426.
[32]. Qosmos ixEngine. http://www.qosmos.com/products/deep-packet-inspection-engine/.
[33]. Ipoque PACE. https://www.ipoque.com/products/pace.
[34]. Windriver Content Inspection Engine. http://www.windriver.com/products/product-overviews/PO_Wind-River-Content-Inspection-Engine.pdf.
[35]. Procera PacketLogic Content Intelligence. https://www.proceranetworks.com/content-intelligence.html.
[36]. Cisco NBAR. http://www.cisco.com/c/en/us/products/ios-nx-os-software/network-based-application-recognition-nbar/index.html.
[37]. Junos OS Next Generation Application Identification. https://www.juniper.net/documentation/en_US/junos15.1x49/topics/concept/services-application-identification-techniques-understanding.html.
[38]. Cascarano N, Ciminiera L, Risso F. Optimizing deep packet inspection for high-speed traffic analysis. Network System Manager. 2011; 19(1):7–31.

[39]. Duffield N., Lund C. “Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure”. ACM Internet Measurement Conference, 2003.
[40]. Callado A., Kamienski C., Szabo G., Gero B., Kelner J., Fernandes S., Sadok D. A Survey on Internet Traffic Identification; Communications Surveys & Tutorials, IEEE Volume 11, Issue 3, 3rd Quarter 2009 Page(s):37 – 52.
[41]. Duffield, N.; Lund, C.; Thorup, M., “Learn more, sample less: control of volume and variance in network measurement”, IEEE Transactions in Information Theory, vol. 51, no. 5, pp. 1756-1775, 2005.
[42]. PSAMP. http://www.rfc-editor.org/rfc/rfc5476.txt.
[43]. Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju and James W. Hong, “The Architecture of NGMON: a Passive Network Monitoring System for High-Speed IP Networks”, Accepted to appear in the Proc. of the 13th IFIP/IEEE International Workshop on Distributed Systems:Operations & Management (DSOM 2002), Montreal, Canada, October 21-23, 2002.
[44]. InveaTech FlowMon. https://www.invea.com/en/products/flowmon.
[45]. IPFIX. https://tools.ietf.org/html/rfc5101.
[46]. M.-S. Kim, Y. J. Won, and J. W. Hong. Characteristic analysis of internet traffic from the perspective of flows. Comp. Comm., 29(10):1639–1652, 2006.
[47]. R. Sommer and A. Feldmann. NetFlow: Information loss or win? In ACM SIGCOMM Internet Meas. Workshop, 2002.
[48]. MIB. https://tools.ietf.org/html/rfc3418.
[49]. SNMP. https://www.ietf.org/rfc/rfc1157.txt.
[50]. Colin J. Bennett, Andrew Clement, Kate Milberry. Introduction to Cyber-Surveillance. Cyber-Surveillance in Everyday Life. Vol. 9, No 4 (2012)
[51]. T. Farah, and L. Trajkovic, "Anonym: A tool for anonymization of the Internet traffic." In IEEE 2013 International Conference on Cybernetics (CYBCONF), 2013, pp. 261-266.
[52]. F. Risso, and L. Degioanni, “An Architecture for High Performance Network Analysis”, Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001), Hammamet, Tunisia, July 2001.
[53]. PF_RING ZC. http://www.ntop.org/products/packet-capture/pf_ring/.
[54]. Andrew M White, Srinivas Krishnan, Michael Bailey, Fabian Monrose, and Phillip Porras. Clear and Present Data: Opaque Traffic and its Security Implications for the Future. NDSS, 2013.
[55]. Network Intrusion Detection Signatures. http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-five.
[56]. Yuji Waizumi, Yuya Tsukabe, Hiroshi Tsunoda, and Yoshiaki Nemoto. Network Application Identification Based on Communication Characteristics of Application Messages. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:3, No:12, 2009
[57]. T. Karagiannis, K. Papagiannaki and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In Proc. of ACM SIGCOMM, August 2005
[58]. Nguyen, T. T. T. and Armitage, G.2008. A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys &Tutorials 10, 4, (2008), 56-76.
[59]. T. AbuHmed, A. Mohaisen, and D. Nyang, “A survey on deep packet inspection for intrusion detection systems,” CoRR, vol. abs/0803.0037, 2008. [Online]. Available: http://arxiv.org/abs/0803.0037.

[60]. Koloud Al-Khamaiseh, Shadi ALShagarin. A Survey of String Matching Algorithms. Int. Journal of Engineering Research and Applications. ISSN : 2248-9622, Vol. 4, Issue 7(Version 2), July 2014, pp.144-156.
[61]. D. E. Taylor, "Survey and taxonomy of packet classification techniques," ACM Comput. Surv., vol. 37, no. 3, pp. 238–275, 2005.
[62]. L7-filter. http://l7-filter.sourceforge.net/.
[63]. J. E. Hopcroft and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison Wesley, 1979.
[64]. S. Kumar and P. Crowley. Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection. In Proc. of SIGCOMM, 2006.
[65]. D. Ficara, S. Giordano, G. Procissi. An Improved DFA for Fast Regular Expression Matching. In Proc. of SIGCOMM, 2008.
[66]. F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, and R. H. Katz. Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection. In Proc. of ANCS, 2006.
[67]. S. Kumar, B. Chandrasekaran, J. Turner, and G. Varghese. Curing Regular Expressions Matching Algorithms From Insomnia. In Proc. of ANCS, 2007.
[68]. R. Smith, C. Estan, S. Jha, and S. Kong. Deflating the Big Bang: Fast and Scalable Deep Packet Inspection with Extended Finite Automata. In Proc. of ACM SIGCOMM, 2008.
[69]. C. Liu, J. Wu. Fast Deep Packet Inspection with a Dual Finite Automata. IEEE Transactions on Computers, Vol. 62.
[70]. Zip Bomb. http://xeushack.com/zip-bomb/.
[71]. J. Olivain and J. Goubault-Larrecq. Detecting subverted cryptographic protocols by entropy checking. Research Report LSV-06-13, Laboratoire Specification et V erification, ENS Cachan, France, June 2006.
[72]. H. Balakrishnan and V. Padmanabhan, "How network asymmetry affects TCP", in IEEE Communications Magazine,Vol. 39, pp. 60 -67, April 2001.
[73]. The Perils of Deep Packet Inspection. http://www.symantec.com/connect/articles/perils-deep-packet-inspection.
[74]. A Hardware Platform for Network Intrusion Detection and Prevention. http://www.cc.gatech.edu/home/wenke/papers/np3.pdf.
[75]. Jonathan Thompson, Kristofer Schlachter. An Introduction to the OpenCL Programming Model. Distributed Computing CSCI-GA.2631-001 (Multicore Programming), 2012
[76]. InveaTech NetCOPE. https://www.invea.com/en/products-and-services/fpga-development-kit.
[77]. G NetFPGA. https://github.com/NetFPGA/netfpga/wiki/G.
[78]. Shaly Laurence, Anuros Thomas K. Review of SRAM based architecture for TCAM. International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. II, Special Issue VI, February 2015.
[79]. Towards software-based signature detection for intrusion prevention on the network card. http://www.cs.vu.nl/~herbertb/papers/cardguard_raid05.pdf.
[80]. S. Han, K. Jang, K. Park, and S. Moon, "Building a Single-Box 100 Gbps Software Router," in IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), May 2010, pp. 1–4.
[81]. A. Feitoza Santos, S. F. de Lacerda Fernandes, P. Gomes Lopes Junior, D. Fawzi Hadj Sadok, and G. Szabo, "Multi-gigabit traffic identification on gpu," in Proceedings of the First Edition Workshop on High Performance and Programmable Networking, ser. HPPN '13. New York, NY, USA: ACM, 2013, pp. 39–44. [Online]. Available:http://doi.acm.org/10.1145/2465839.2465845
[82]. GPUDirect. https://developer.nvidia.com/gpudirect.

[83]. DirectGMA. http://developer.amd.com/tools-and-sdks/graphics-development/firepro-sdk/firepro-directgma-sdk/.
[84]. Utilizing GPUDirect 3RD Party DMA Features for 10GbE NIC and GPU applications. http://on-demand.gputechconf.com/gtc/2013/presentations/S3300-GPUDirect-DMA-Features.pdf.