

Решения сетевого учета Juniper Networks

Чак Семерия, инженер по маркетингу
Ханнес Гредлер, сервисный инженер

Poplar
SYSTEMS

Содержание

Введение	3
Учет по данным интерфейса: традиционное решение	3
Командная строка.....	4
Ручной ввод команд.....	4
Сценарии запроса-ожидания	4
JUNOScript API	5
Протокол SNMP	6
Идеальное учетное решение	7
NetFlow – новая технология учета потоков	8
Альтернативы решениям потокового учета, предлагаемые компанией Juniper Networks ..	10
Учет на основе фильтров	11
Учет с использованием протокола MPLS	13
Учет по классам назначения	13
Правила маршрутизации.....	14
Действующая модель DCU	15
Влияние DCU на производительность форвардинга	17
Административная масштабируемость решений биллинга и учета.....	18
Учет по интерфейсам	18
Учет на основе фильтров.....	18
Учет с использованием протокола MPLS.....	19
Учет по классам назначения (DCU).....	19
Учетные профили Juniper Networks	20
Заключение	22

Введение

Поставщикам услуг для максимизации прибыли на инвестированный капитал необходимы инструменты, позволяющие обеспечивать параметры предоставляемых услуг и выставять абонентам счета за пользование услугами. В недавнем прошлом провайдерам было достаточно простых моделей биллинга с фиксированными ставками (flat-rate). Мы уверены, что последние разработки Juniper Networks в области аппаратного сбора статистики помогут провайдерам уйти от простых моделей и перейти к многоуровневому биллингу.

В первой части данной статьи дается обзор традиционно использовавшихся провайдерами для сбора статистики средств учета на основе интерфейсов. Здесь же говорится о механизмах переноса статистики от маршрутизатора-сборщика к системам согласования (mediation systems), генерирующим на основе полученных данных информацию для биллинга. Основное внимание уделяется тому, как эти традиционные средства поддерживаются магистральными Интернет-маршрутизаторами Juniper Networks® и операционной системой JUNOS™.

Вторая часть рассказывает о некоторых новых инструментах биллинга и учета, специально разработанных Juniper Networks для упрощения и облегчения процесса сбора учетной информации. Учет с использованием фильтров позволяет провайдеру конфигурировать индивидуальные фильтры пакетов с целью идентификации и последующего подсчета пользовательских потоков трафика. Класс назначения (Destination class usage, DCU) позволяет координировать учетную политику с политикой маршрутизации и динамически адаптировать ее к быстро меняющейся маршрутной среде. Т.к. механизмы учета с использованием фильтров и с использованием DCU реализованы аппаратно, их можно применять как на границе, так и в магистральной, без существенного влияния на производительность продвижения пакетов. В заключение мы расскажем об учетных профилях (accounting profiles), предназначенных для предварительной обработки учетной информации и повышения эффективности передачи статистики от маршрутизатора к системам согласования.

Учет по данным интерфейса: традиционное решение

Исторически, провайдеры собирали учетные данные с маршрутизаторов, основываясь на данных интерфейса, т.е. объеме данных, передаваемых и принимаемых абонентом (рис. 1).



Рисунок 1. Учет по данным интерфейса

Абонент на рис. 1 подключен к локальному провайдеру, предоставляющему услугу доступа в Интернет. Обычно учет по данным интерфейса выполняется на входном интерфейсе граничного маршрутизатора провайдера (provider edge, PE) или, если провайдер предоставляет управляемую услугу маршрутизации, на выходном интерфейсе клиентского маршрутизатора. Собираемая информация состоит из счетчиков входных и выходных пакетов и байтов.

ОС JUNOS предусматривает доступ к интерфейсной учетной информации несколькими способами.

- Командная строка (непосредственное взаимодействие или сценарии запроса).
- Прикладной программный интерфейс JUNOScript™ API.
- Протокол SNMP.

Командная строка

Интерфейс командной строки (command-line interface, CLI) автоматически загружается по окончании загрузки маршрутизатора. Он позволяет администратору выполнять различные задачи, включая конфигурирование ОС JUNOS, мониторинг и диагностику программного обеспечения (ПО), установление сетевых соединений, мониторинг маршрутизатора. Вводить команды можно непосредственно или через сценарии запроса-ожидания (send/expect scripts).

Ручной ввод команд

Команды вводятся либо с консоли, либо через сетевое соединение. Статистику по интерфейсному трафику показывает команда

```
show interfaces <interface-name> detail
```

Сценарии запроса-ожидания

Учетная информация также доступна с помощью сценариев, используемых администратором для автоматизации управления маршрутизаторами. Обычно языки сценариев достаточно просты и используют команды CLI. Но программирование сценариев с такими командами бывает сложным: выходные данные в символах ASCII должны анализироваться на наличие определенных строк, которые затем

извлекаются и возвращаются. Если формат символьного вывода вдруг изменится (например, в результате смены версии ПО), придется перепрограммировать логику анализа. Таким образом, использование сценариев для продолжительной работы с различными версиями ПО усложняет задачу и увеличивает вероятность ошибок.

JUNOScript API

Язык XML (extensible markup language) позволяет писать сценарии, которые передают и принимают данные в структурированном формате, схожем с языком HTML. Язык XML прост, легко генерируется, однозначно анализируется и не зависит от платформы. Задача сценария – найти в выходных данных формата XML определенные метки, указывающие на конкретные атрибуты в множестве объектов, что значительно проще, чем анализировать выходной поток на наличие символьных строк, которые к тому же могут меняться от одной версии ПО к другой. Интерфейс JUNOScript предусматривает API на базе XML для программного конфигурирования и мониторинга маршрутизаторов Juniper Networks.

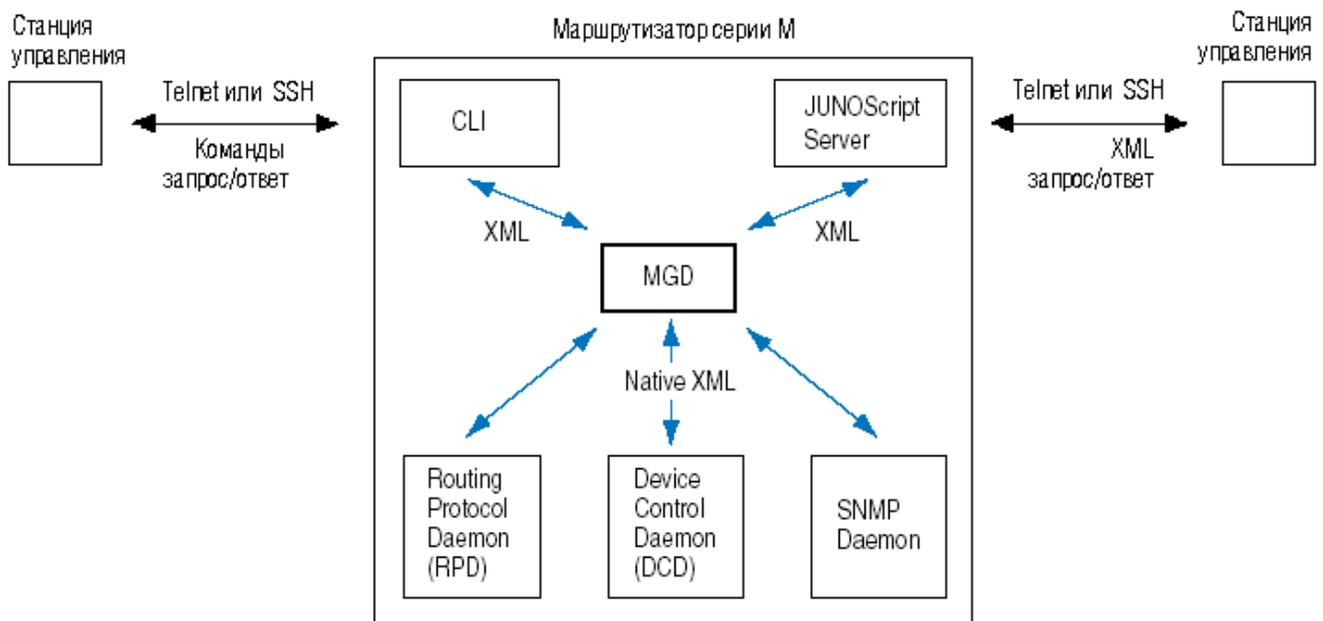


Рисунок 2. Программный интерфейс JUNOScript API

ОС JUNOS состоит из ряда процессов (daemon), каждый из которых функционирует в своем пространстве защищенной памяти как независимый процесс поверх общего ядра. Язык XML полностью интегрирован в процесс управления (MGD), который использует XML для связи с другими системными процессами (см. рис. 2). Это означает, что новые функции можно сразу же использовать с помощью JUNOScript API; а не ждать почти год, пока будет разработан управляющий интерфейс XML. На самом деле, JUNOScript API поддерживает расширенный набор функций, по сравнению с командным интерфейсом. Полная интеграция XML в MGD, осуществленная Juniper Networks, - уникальное явление на рынке поставщиков услуг.

Применение JUNOScript API ускоряет и упрощает разработку сценариев, делая их более надежными, т.к. метки XML не меняются от одной версии ПО к другой, даже

если меняются команды CLI. Язык XML позволяет нам гибко изменять командный интерфейс, гарантируя при этом устойчивую и надежную работу сценариев в действующих сетях.

Протокол SNMP

Протокол SNMP (Simple Network Management Protocol) позволяет системе управления получать учетную информацию от сетевых элементов через запросы к базе данных управляющей информации устройств (Management Information Base, MIB). Модель SNMP состоит из сборщиков сетевой информации (менеджеров), периодически опрашивающих сетевые элементы (агенты) для получения учетной информации. Полученная исходная информация обрабатывается системой согласования и биллинга (рис. 3).

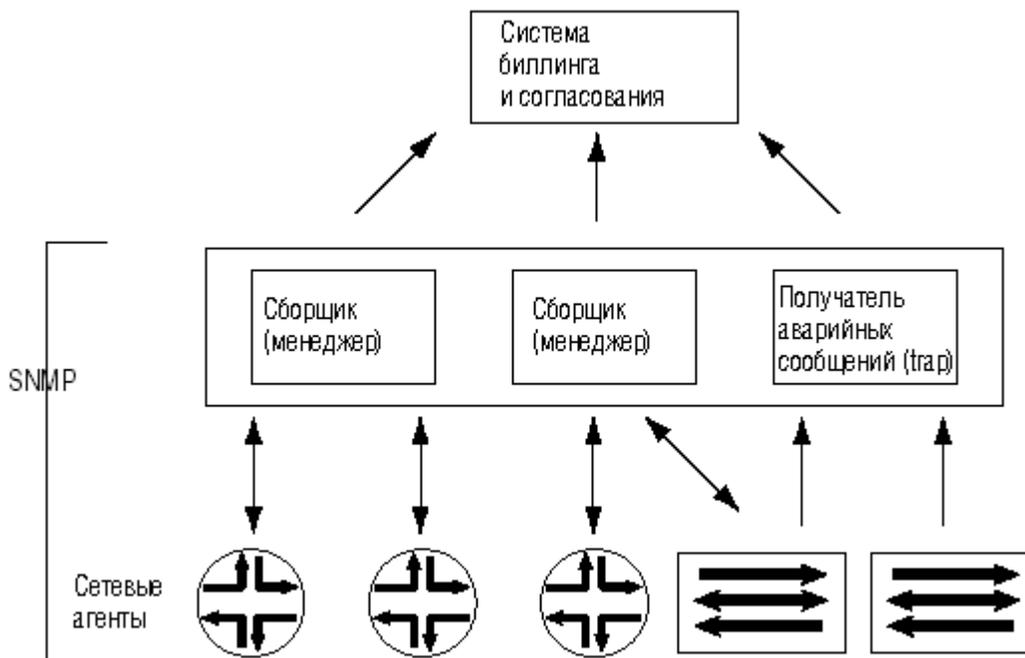


Рисунок 3. Взаимодействие протокола SNMP с системой согласования и биллинга

Самый важный объект SNMP с точки зрения биллинга и учета – это таблица интерфейсов IfTable. Каждая строка IfTable содержит подробную информацию об интерфейсе маршрутизатора, в том числе:

- тип интерфейса;
- состояние интерфейса (готов / не готов);
- максимальный размер кадра передачи (MTU), поддерживаемый интерфейсом;
- счетчики с количеством входных и выходных байт, входных и выходных пакетов, ошибок.

SNMP разрабатывался в конце 80-х, когда сети строились с использованием интерфейсов Ethernet (10 Мбит/с), Token Ring, DS-0 и T-1, т.е. 32-разрядных счетчиков было более чем достаточно для учета объема передаваемой и

принимаемой информации. Сегодня 32-разрядный счетчик может зациклиться за достаточно короткий промежуток времени. Противостоять этому можно, сократив интервал опроса, т.е. увеличив частоту обращения к счетчику. К сожалению, этот подход неприменим для высокопроизводительных маршрутизаторов, необходимых для поддержки интерфейсов OC-192c/STM-64 (SONET/SDH) и OC-768c/STM-256 или тысяч channelized интерфейсов. Рекомендация RFC 2233, которую поддерживает ОС JUNOS, предлагает другое решение: расширение IfTable (IFXTable), использующее 64-разрядные счетчики, разрядности которых должно быть достаточно для устранения проблемы закливания на долгие годы.

Но у SNMP есть ряд других ограничений, касающихся передачи больших объемов учетной информации для высокопроизводительных маршрутизаторов.

- Передаваемые протоколом SNMP блоки данных (protocol data units, PDU) очень неэффективно используют полосу пропускания за счет большого объема служебной информации: для выборки одного значения счетчика управляющая станция SNMP должна передать запрос GetRequest (приблизительно 80 байт) и получить ответ GetResponse (тоже около 80 байт). Таким образом, для получения значения счетчика, которое может занимать 4 или 8 байт, нужно передать 160 байт служебной информации.
- SNMP – очень медленный протокол, т.к. использует транспортный протокол с нулевым окном. Это означает, что каждый запрос должен получить свой ответ, прежде чем можно будет передавать следующий запрос. Т.к. SNMP использует протокол UDP (User Datagram Protocol), а не протокол с переменным окном типа TCP, он не поддерживает работу с одновременно рассылаемыми PDU, что вызывает проблемы при необходимости решения параллельных задач.
- SNMP потребляет значительную долю ресурсов ЦПУ. Это относится как к сборщикам информации, генерирующим запросы, так и к управляемым элементам, обрабатывающим запросы и определяющим, какую информацию запрашивает сборщик.

Идеальное учетное решение

Используя ОС JUNOS, провайдер может получить доступ к учетной информации разными способами: через командный интерфейс, сценарии send/expect, JUNOScript API, SNMP. Учет по данным интерфейса предоставляет просто необработанные данные об объемах переданного и принятого каждым абонентом трафика.

Традиционные операторы голосовой связи собирали большие объемы биллинговой информации по своим клиентам, что позволяло выставлять счета, например, по количеству звонков, по номеру назначения, по времени суток, по продолжительности разговора. В сравнении с этим, учет по данным интерфейса, основанный только на объеме трафика и не принимающий во внимание тип трафика и адрес назначения, является неприемлемым решением для высокопроизводительных сетей.

Попробуем описать идеальное с точки зрения Интернет-маршрутизации учетное решение. Оно должно предоставлять такую учетную информацию, которая позволит перейти от ограничений фиксированных тарифов к преимуществам многоуровневого биллинга. В конце нашего обзора мы сравним идеальное решение с различными учетными инструментами, поддерживаемыми маршрутизаторами Juniper Networks и

ОС JUNOS, чтобы увидеть, насколько близко Juniper Networks подошла к идеальной модели.

Идеальное решение учета характеризуется рядом определенных признаков.

Система учета должна предоставлять информацию о полном объеме данных, переданных и принятых сайтом клиента.

Модель учета должна предусматривать информацию о конкретных типах передаваемого трафика (приложение, класс обслуживания, одноадресный, многоадресный) для поддержки биллинга в мультисервисных сетях.

Схема учета должна содержать информацию об источнике и адресате потока трафика для поддержки классов передачи, учитывающих эти параметры.

Принципы учета должны позволять маршрутизатору, собирающему данные, принимать на себя часть функций системы биллинга и согласования для предварительной обработки или агрегации статистических данных.

Стратегия учета должна предусматривать различные форматы представления учетных данных (SNMP, сценарии XML, плоские файлы, передаваемые по FTP).

Система учета должна обеспечивать передачу статистики без неоправданного увеличения загрузки сети и без перегрузки системы сбора и согласования.

Должна быть организована надежная передача данных от маршрутизатора-сборщика до системы управления, генерирующей информацию, используемую для биллинга.

Стратегия учета должна предусматривать адаптацию к изменениям в среде маршрутизации. Это требование отменяет использование статических шаблонов учета, которые не меняются при изменении сетевых условий.

Сбор учетной информации не должен влиять на сетевую производительность, чтобы решение учета можно было использовать не только на границе сети, где создаются услуги, но и в магистрали, на высокоскоростных каналах.

Наконец, идеальное решение учета должно предусматривать средства гибкой разработки и внедрения учетной стратегии, отражающей конкретные требования провайдера.

NetFlow – новая технология учета потоков

Учет потоков (Flow) изначально разрабатывался с тем, чтобы провайдер мог собирать учетную информацию по отдельным сетевым услугам и использовать различные клиентские тарифы для каждой услуги. Отдельный сетевой поток можно в общем виде определить как однонаправленную последовательность пакетов между определенными IP-адресами источника и назначения. Однако можно дать и более подробные определения, рассматривая протокол IP, порты TCP/UDP источника и адресата, биты типа обслуживания в IP-адресе (IP precedence).

Первое решение учета потоков, известное как NetFlow, представляло собой расширение стандартного процесса поиска маршрута (route-lookup). Когда первый пакет потока трафика поступает в маршрутизатор, производится поиск в кэш-памяти маршрутизатора с использованием хешированного ключа, вычисляемого на основе полей заголовка пакета. Если система не находит соответствия в кэше, то

происходит стандартный поиск наиболее длинного соответствия в IP таблице, и пакет продвигается дальше. После первого просмотра адресной таблицы в кэше создается новый объект flow (поток), и определяются соответствующие ему структуры учетных данных. При поступлении в маршрутизатор последующих пакетов, принадлежащих данному потоку, вычисляется ключ, происходит обращение к объекту flow, выполняются учетные операции, пакет продвигается с использованием информации в кэш-памяти, и обновляется таймер старения кэш-памяти. Процесс учета потока пересылает учетную запись сборщику после прекращения передачи потока. Учетная запись содержит статистику (счетчики и временные метки), собранные в период активности потока. До того как учетная запись станет доступной системе биллинга и согласования, объект flow либо должен автоматически прекратить свое существование, либо его необходимо удалить.

NetFlow поддерживает два типа учетных записей. Первый тип (V5) предоставляет для каждого потока пакетов учетную информацию, включающую адрес источника, адрес назначения, классификацию TCP или UDP, поля автономной системы (Autonomous System, AS), время начала и окончания. Второй тип (V8) предоставляет ту же учетную информацию, что и V5, но добавляет возможность агрегации записей для уменьшения нагрузки на сборщиков sflowd.

Вначале инструментарий NetFlow имел успех, несмотря на ряд недостатков, проявляющихся при использовании с магистральными Интернет-маршрутизаторами или при учете потоков на высокоскоростных каналах.

- Каждый раз, когда из-за нестабильности сети меняется адресная таблица на маршрутизаторе NetFlow, необходимо обнулить весь локальный кэш маршрутизации и построить его заново. Это может серьезно сказаться на производительности ЦПУ старых маршрутизаторов. Стоит отметить, что Juniper Networks не использует кэширование маршрутов в своем процессе форвардинга. Вместо этого подсистема форвардинга пакетов (Packet Forwarding Engine) выполняет традиционный поиск наиболее длинного соответствия для каждого пакета с использованием высокопроизводительной IC Internet Processor II™.
- Маршрутизаторы NetFlow должны поддерживать информацию состояния для каждого потока пакетов. Хранение больших объемов такой информации может создавать проблемы как для нагруженных маршрутизаторов доступа, так и для магистральных маршрутизаторов, которым может потребоваться хранить информацию состояния для миллионов потоков.

Еще одной проблемой использования NetFlow для учета потоков на интерфейсах со скоростями, превышающими OC-3/STM-1, является необходимость статистической выборки для обработки увеличивающихся объемов трафика и уменьшения нагрузки на ЦПУ маршрутизаторов и сборщиков. Однако статистическая выборка пока может выполняться только на границе сети, т.к. традиционно она реализуется программно (что может серьезно повлиять на производительность форвардинга унаследованных маршрутизаторов).

По этим причинам применение учетного механизма NetFlow ограничено слабо нагруженными интерфейсами на границе сети, и NetFlow не может без проблем применяться в сетевой магистрали. Зачастую проектировщикам приходится

создавать обходные решения, развертывая несколько двухпортовых учетных маршрутизаторов на границе (см. рис. 4).

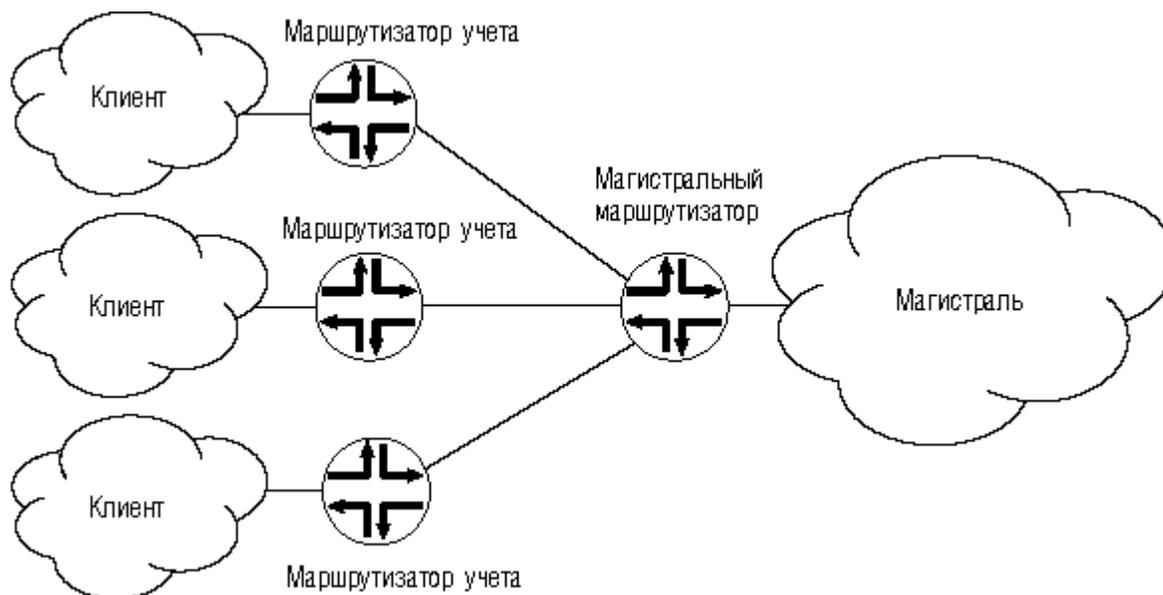


Рисунок 4. Граничные учетные маршрутизаторы

Как показано на рис. 4, один порт учетного маршрутизатора обеспечивает клиентский доступ, второй – связь с сетевой магистралью. Учетные маршрутизаторы выполняют только базовые функции маршрутизации, т.к. используются в основном как учетные платформы. Необходимость установки специальных маршрутизаторов для поддержки учета на основе потоков увеличивает как общесетевые капитальные затраты, так и эксплуатационные расходы, связанные с управлением дополнительными системами.

На магистральном маршрутизаторе может использоваться механизм форвардинга без кэш-памяти, т.к. учетная информация уже была собрана на границе сети. Такой механизм может рассматриваться как простейшая версия интегрального Интернет-процессора Internet Processor II, т.к. при этом выполняется аппаратный поиск адресов, но отсутствует интеллектуальность, необходимая для дополнительных функций обработки пакетов (фильтрации, ограничения скорости, сбора статистики и учета).

Альтернативы решениям потокового учета, предлагаемые компанией Juniper Networks

Методы потокового учета схожи с применяемыми на традиционных УАТС, отсюда возникает вопрос, должен ли провайдер выставить счета, отслеживая все потоки трафика. В качестве ответа на ограничение масштабируемости решений потокового учета мы разработали ряд уникальных решений учета, основывающихся на использовании классов.

- Учет на основе фильтров

- Учет с использованием протокола MPLS
- Учет по классам назначения (DCU)

Все решения реализованы аппаратно на базе интегральной схемы Internet Processor II и ОС JUNOS.

Учет на основе фильтров

Сердцем подсистемы форвардинга пакетов всех маршрутизаторов Juniper Networks является ИС Internet Processor II, выполняющая не только поиск маршрута, но и ряд дополнительных функций обработки пакетов без существенного влияния на производительность форвардинга. Дополнительные функции включают фильтрацию, выборку, подсчет пакетов, а также контроль трафика (traffic policing).

Производительность ИС Internet Processor II достигается за счет применения исключительно гибких методов программирования ИС (см. рис. 5). Сетевому администратору достаточно написать фильтры классификации входного и выходного трафика в уже привычной ему форме. Компилятор, разработанный Juniper Networks, автоматически оптимизирует фильтры для ИС Internet Processor II в ходе процесса конфигурирования. Наконец, фильтры записываются в ИС, где они выполняются аппаратно, т.е. быстро и эффективно. Способность ИС Internet Processor II выполнять эти функции, сохраняя стабильную производительность форвардинга, позволяет использовать их на границе сети, в высокоскоростных каналах и в магистрали.

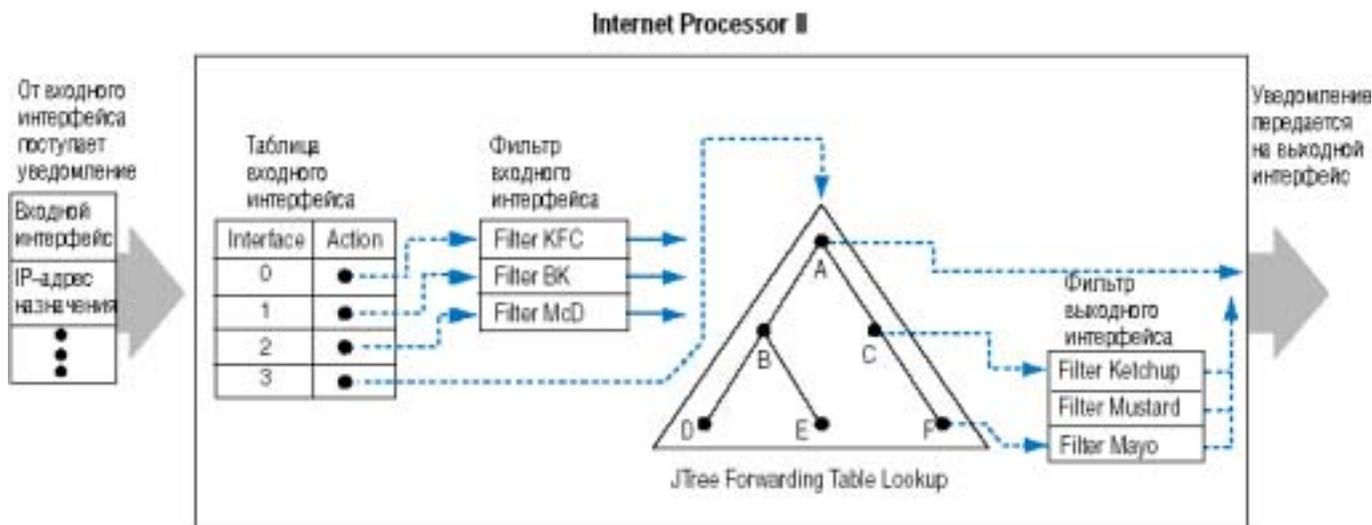


Рисунок 5. Архитектура обработки пакетов ИС Internet Processor II

Для приложений биллинга и учета ОС JUNOS и ИС Internet Processor II позволяют администраторам конфигурировать фильтры, предназначенные для идентификации, пропуска и подсчета классов трафика, определенных пользователем. Критерии, используемые для определения классов трафика, весьма разнообразны и могут включать, к примеру, адрес источника, адрес назначения, порты TCP и UDP источника и назначения. Система поддерживает 64-разрядные счетчики пакетов и байтов, которые можно считывать из командной строки, из сценариев, через интерфейс JUNOScript API и по протоколу SNMP.

Рисунок 6 иллюстрирует применение ОС JUNOS и IC Internet Processor II для поддержки типичного учетного приложения с использованием фильтров.

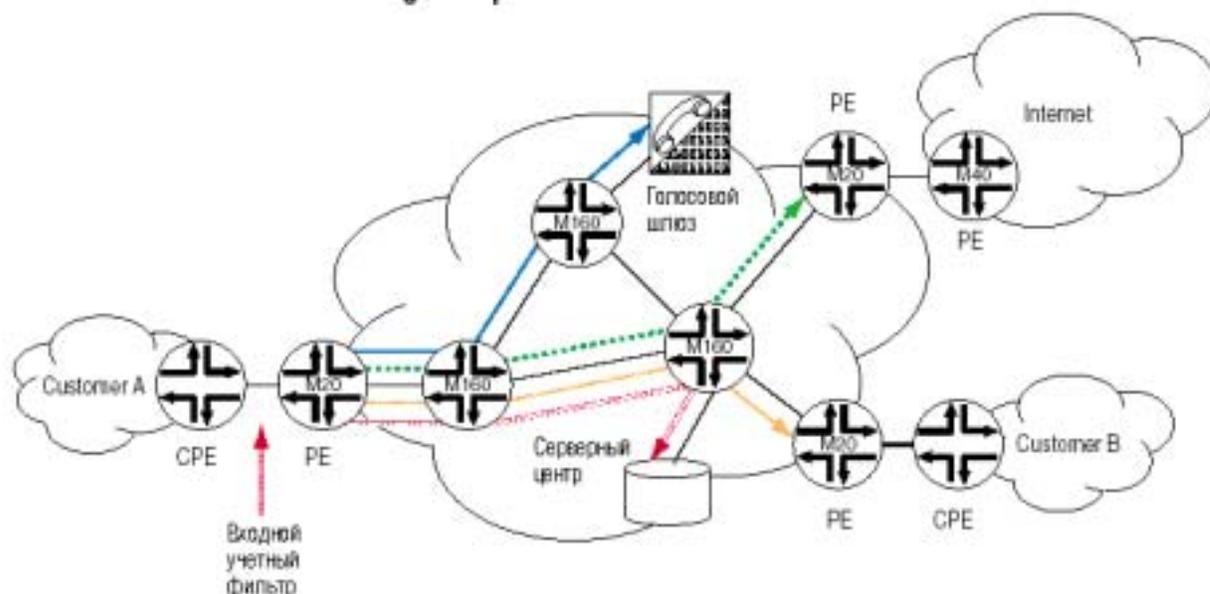


Рисунок 6. Пример учетного приложения с использованием фильтров

В примере на рис. 6 сетевой администратор создает счетчики пакетов и байтов для четырех классов трафика.

- Первый класс – трафик реального времени клиента А (Customer A), направляемый в голосовую сеть. Т.к. для трафика реального времени провайдер использует направленный виртуальный канал (expedited forwarding), этот класс идентифицируется по байту DiffServ в заголовке пакета.
- Второй класс – это трафик клиента А класса best-effort, остающийся в пределах автономной системы (AS) провайдера и поступающий в серверный центр (server farm). Этот класс идентифицируется с помощью фильтра по IP-адресу назначения, указывающему на серверный центр.
- Третий класс состоит из трафика клиента А класса best-effort, который выходит за пределы AS и предназначается клиенту В (Customer B). Этот класс идентифицируется с помощью фильтра по IP-адресу назначения клиента В.
- Четвертый класс – это трафик клиента А класса best-effort, который выходит за пределы AS и направляется в сеть Интернет. К этому классу относится весь трафик, не принадлежащий ни одному из первых трех классов.

Как говорилось выше, счетчики пакетов и байтов для этих фильтров можно считывать из командной строки, из сценариев, через интерфейс JUNOScript API и по протоколу SNMP.

Учет с использованием фильтров весьма эффективен, но имеет некоторые ограничения: администратор должен заранее знать, какой трафик нужно учитывать, и иметь достаточный для написания фильтров опыт. Например, если провайдеру нужно подсчитывать объем входного трафика по всем абонентам, то каждый раз при

добавлении или удалении абонента администратор должен обновлять фильтры на всех маршрутизаторах, выполняющих учетные функции в сети провайдера. Хотя пакетные фильтры могут поддерживаться с помощью сценариев, этот подход не является действительно масштабируемым.

В действующих сетях учет с использованием фильтров полезен для грубой классификации передаваемого клиентом трафика. Это превосходный инструмент для учета явных классов трафика (по классам обслуживания или байту TOS), а также для подсчета трафика, посылаемого в сети с постоянной конфигурацией, такие как серверный центр.

Учет с использованием протокола MPLS

Протокол MPLS (Multiprotocol Label Switching) имеет ряд преимуществ, таких как регулирование трафика, поддержка ВЧС уровней 2 и 3, обеспечение устойчивости сети. Как показывает рис. 7, он обладает также еще одним преимуществом с точки зрения биллинга и учета.

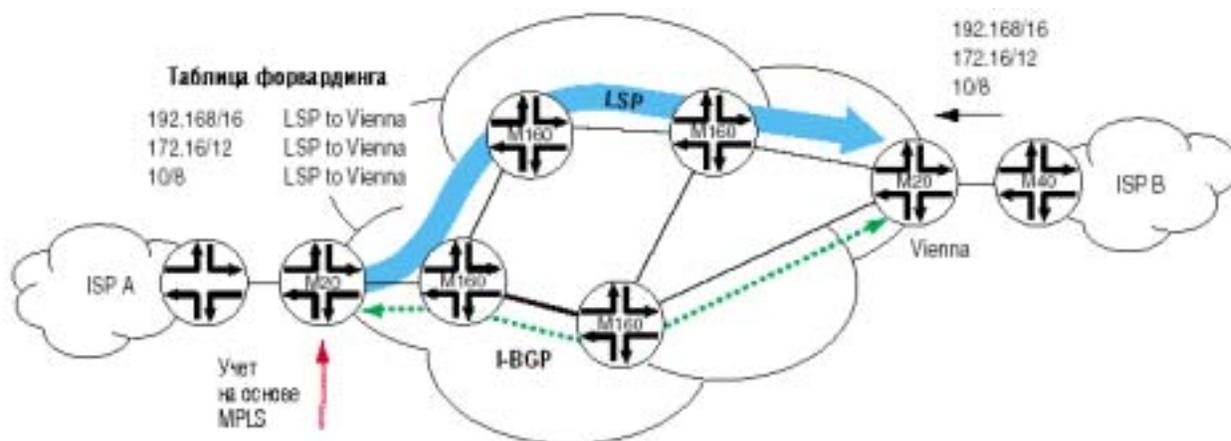


Рисунок 7. Учет с использованием протокола MPLS

В этом примере администратор собирает статистику по трафику, проходящему по пути коммутации меток (LSP) к маршрутизатору с обозначением Vienna. Здесь мы находим привлекательное решение, т.к. префиксы провайдера B (ISP B) динамически отображаются на путь LSP, т.е. решение учета легко масштабируется от одного до более 100000 префиксов. Это значительно облегчает задачу администратора, т.к. отображение префиксов на путь LSP происходит динамически, а не статически. Разнообразную статистику LSP, так же как счетчиков пакетов и байтов, можно считывать из командной строки, из сценариев, через интерфейс JUNOScript API и по протоколу SNMP.

Учет по классам назначения

Несмотря на все преимущества MPLS как средства динамического учета, нужно понимать, что не все провайдеры хотят внедрять протокол MPLS в своих сетях. Тем не менее, им все равно необходимы средства, позволяющие учитывать на входных и выходных точках трафик, проходящий по их сетям. Входная точка для некоторого класса трафика легко идентифицируется входным интерфейсом. Выходные точки можно идентифицировать с помощью предложенного Juniper Networks механизма классификации трафика по назначению (DCU), группирующего префиксы назначения

по непересекающимся множествам и поддерживающего счетчики пакетов и байтов на каждом интерфейсе для каждого множества.

Для простоты описания Juniper Networks DCU мы остановимся на трех важных вопросах, касающихся данного механизма.

- Правила маршрутизации
- Действующая модель DCU
- Влияние DCU на производительность форвардинга

Правила маршрутизации

Маршрут – это указатель на сеть назначения. Получая объявление маршрута, маршрутизатор узнает не только префикс маршрута, но и ряд других атрибутов, относящихся к маршруту. Протоколы OSPF (Open Shortest Path First) и IS-IS (Intermediate System to Intermediate System) обычно несут ограниченное число атрибутов, тогда как протокол BGP (Border Gateway Protocol) поддерживает передачу большого количества атрибутов. Интересно заметить, что объявления маршрута распространяются в направлении, противоположном потоку данных пользователя, направленному к объявляемому префиксу (см. рис. 8).



Рисунок 8. Объявления маршрута и поток трафика данных передаются в разных направлениях

Администратор может создавать самые разнообразные правила маршрутизации. Обычно они используются для контроля импорта и экспорта маршрутов к другим маршрутизаторам (или автономным системам). Кроме того, с их помощью можно изменять атрибуты маршрута, такие как метрика, AS-Path или сообщество протокола BGP.

Реализация фильтров на базе правил для группировки маршрутов и последующей их совместной обработки коренным образом отличается от реализации фильтров межсетевых экранов. В последнем случае фильтр используется для классификации пользовательских *пакетов данных* по результатам анализа заголовка пакета. При соответствии пакета условиям межсетевого фильтра пакет может быть принят, отброшен, подсчитан, выбран, зарегистрирован и т.д. Фильтр же на основе правил маршрутизации используется для классификации *маршрутов*, основанной на

анализе атрибутов, объявленных вместе с маршрутом. Атрибуты могут включать идентификатор зоны OSPF или значения полей тэгов, номер уровня IS-IS, BGP AS-Path, сообщество, локальные предпочтения и т.п. Для всех маршрутов одного класса выполняются одинаковые действия по результатам применения правил маршрутизации, например:

- пропустить и распространить все соответствующие маршруты;
- отклонить и не распространять все соответствующие маршруты;
- модифицировать определенные атрибуты для всех соответствующих маршрутов до импорта или экспорта маршрута.

Изменяя фильтры правил маршрутизации, администратор может менять направление потока пакетов в сети, т.к. такое изменение очень сильно влияет на распространение маршрутной информации.

Протокол BGP разрабатывался как маршрутный протокол, управляемый правилами. Первоначально провайдеры писали фильтры для классификации маршрутов по префиксам. Однако с ростом Интернета задача поддержания актуальности фильтров и их распространения в постоянно меняющихся условиях просто исключала масштабирование. Это привело к использованию провайдерами фильтров на базе путей AS, подхода с меньшей дискретностью, но большей управляемостью. С развитием конкуренции и приобретением большими провайдерами провайдеров меньшего масштаба задача управления фильтрами AS усложнилась, т.к. у одного провайдера могло быть несколько номеров AS. Сегодня многие провайдеры для классификации маршрутов используют еще менее дискретный подход на базе атрибутов протокола BGP community и extended community .

Действующая модель DCU

Использование классов назначения (destination class usage, DCU) – это учетный механизм на базе правил, позволяющий поддерживать до 16 различных классов трафика. Операторы сотовой связи обычно предлагают максимум 6 или 7 разных тарифов – локальный, областной, национальный, международный, пиковый, ночной, выходного дня. Применение DCU позволяет сотовому оператору:

- определить до 16 классов обслуживания на маршрутизатор (классы обслуживания иначе называются биллинговыми наборами, или корзинами от англ. bucket);
- использовать локальные правила маршрутизации для ассоциирования маршрутов в адресной таблице с одним или несколькими наборами;
- поддерживать на каждом интерфейсе счетчики пакетов и байтов для учета трафика, соответствующего каждому из наборов.

В примере на рис. 9 поставщик услуг покупает транзитную службу Интернета у двух других провайдеров – провайдера A (AS номер 701) и провайдера Z (AS номер 902). Поставщик услуг хочет выставить своим клиентам счета по двум тарифам, в зависимости от того, через какого провайдера идет трафик.

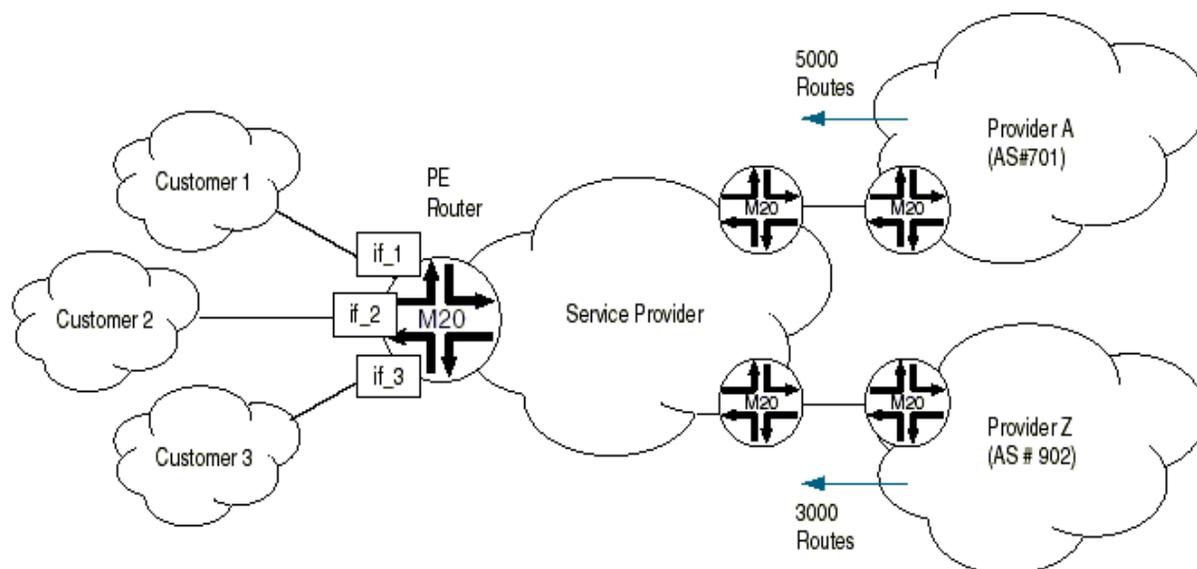


Рисунок 9. Пример использования классов назначения DCU

Поставщик услуг создает фильтр маршрутных правил на базе номера AS или атрибута community, в соответствии с которым все маршруты, полученные от провайдера A, будут ассоциированы с набором DCU 1, а от провайдера Z – с набором DCU 16. Если провайдер A объявит 5000 маршрутов, то ИС Internet Processor II, в соответствии с правилом маршрутизации, установит в единицу первый бит в 16-битовой маске DCU, ассоциированной с каждым из 5000 маршрутов (связав таким образом каждый из этих маршрутов с набором DCU 1). Если провайдер Z объявит 3000 маршрутов, то ИС Internet Processor II, в соответствии с правилом маршрутизации, установит в единицу 16-й бит в 16-битовой маске DCU, ассоциированной с каждым из 3000 маршрутов (связав таким образом все маршруты провайдера Z с набором DCU 16). Теперь предположим, что механизм DCU установлен на интерфейсах If_1 и If_3 маршрутизатора PE, но не установлен на интерфейсе If_2.

Посмотрим, как DCU считает пакеты, прибывающие на маршрутизатор PE от клиентов, обозначенных как Customer 1, Customer 2 и Customer 3.

Предположим, что на интерфейс If_1 поступает пакет, направляющийся в подсеть провайдера Z. ИС Internet Processor II выполняет стандартный поиск маршрута в таблице. Т.к. в битовой маске, ассоциированной с маршрутом, 16-й установлен в 1, ИС Internet Processor II увеличивает счетчик набора 16, связанный с интерфейсом If_1. При этом счетчики набора 16, связанные с другими интерфейсами, не увеличиваются.

Предположим, что на интерфейс If_2 поступает пакет, направляющийся в подсеть провайдера A. ИС Internet Processor II выполняет стандартный поиск маршрута в таблице. Т.к. для интерфейса If_2 механизм DCU не сконфигурирован, ИС Internet Processor II не увеличивает ни один счетчик.

Предположим, что на интерфейс If_3 поступает пакет, направляющийся в подсеть провайдера A. ИС Internet Processor II выполняет стандартный поиск маршрута в таблице. Т.к. в битовой маске, ассоциированной с маршрутом, 1-й установлен в 1,

ИС Internet Processor II увеличивает счетчик набора 1, связанный с интерфейсом If_3. При этом счетчики набора 1, связанные с другими интерфейсами, не увеличиваются.

Обычно каждый маршрут ассоциируется с единственным набором. Но в зависимости от применяемых в системе правил маршрутизации, конкретный маршрут может быть связан с несколькими наборами. Т.е. в битовой маске маршрута все биты могут быть нулями или несколько битов установлены в единицу.

Влияние DCU на производительность форвардинга

Учет с использованием механизма DCU минимально влияет на производительность форвардинга пакетов ИС Internet Processor II, т.к. он практически не увеличивает объем обработки пакетов. Поскольку ИС Internet Processor II не имеет прямого доступа к атрибутам пути AS или сообщества BGP, для создания битовых масок DCU, ассоциированных с префиксами в таблице форвардинга, ОС JUNOS использует правила маршрутизации. Когда пакет обрабатывается процессором ИС Internet Processor II, он производит стандартный поиск в таблице, по маске DCU определяет классы трафика и затем по номеру входного интерфейса, который переносится в объявлении пакета, увеличивает соответствующие счетчики пакетов и байтов данного интерфейса.

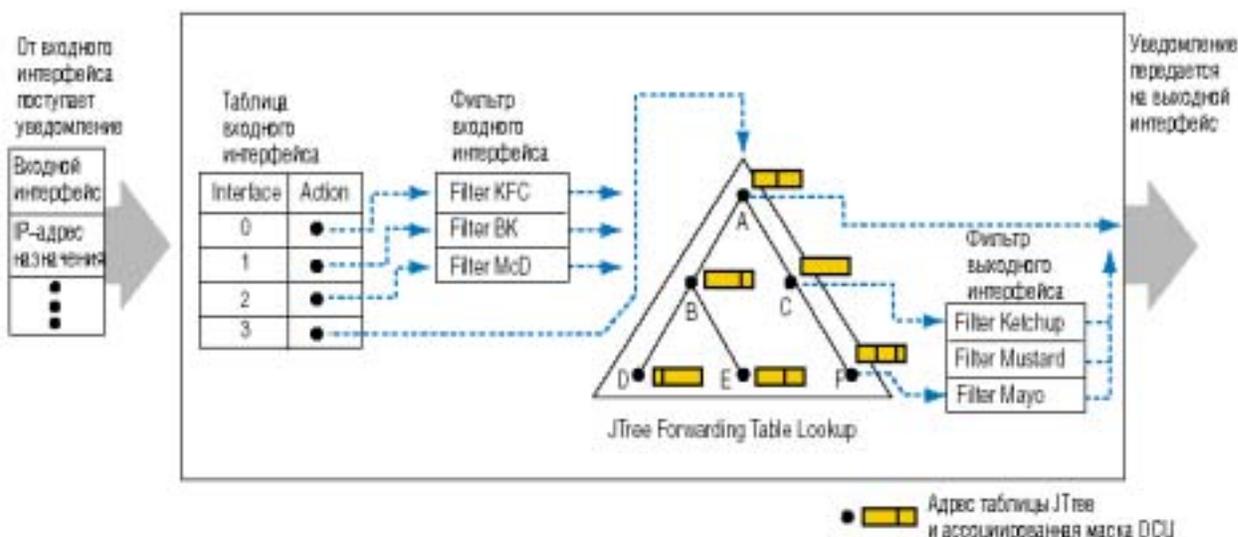


Рисунок 10. ИС Internet Processor II и поддержка DCU

Рисунок 10 иллюстрирует поддержку учетного механизма DCU Интернет-процессором. Единственным изменением в тракте форвардинга стало появление битовой маски, связанной с каждым маршрутом адресной таблицы. Реализация DCU практически не увеличивает сложность аппаратной части ИС Internet Processor II: требуется только простая проверка битовой маски и возможность увеличения счетчика.

Теперь рассмотрим несколько примеров того, как ИС Internet Processor II обрабатывает уведомления пакетов для поддержки DCU.

Предположим, что на входной интерфейс 1 поступает пакет, IP-адрес назначения которого соответствует префиксу A в таблице форвардинга. После того как входной фильтр *Filter BK* примет пакет, и будет произведен поиск маршрута, ИС Internet Processor II анализирует битовую маску префикса A и определяет соответствие пакета набору DCU 8. ИС Internet Processor II увеличивает счетчики пакетов и байтов набора DCU 8, связанные с интерфейсом 1. Затем пакет передается на выходной интерфейс.

Предположим, что на входной интерфейс 2 поступает пакет, IP-адрес назначения которого соответствует префиксу C в таблице форвардинга. После того как входной фильтр *Filter McD* примет пакет, и будет произведен поиск маршрута, ИС Internet Processor II анализирует битовую маску префикса C и определяет, что он не соответствует ни одному из наборов DCU. ИС Internet Processor II продолжает обработку пакета, выполняя фильтр *Filter Ketchup*, не обращаясь к счетчикам.

Предположим, что на входной интерфейс 3 поступает пакет, IP-адрес назначения которого соответствует префиксу F в таблице форвардинга. После того как будет произведен поиск маршрута, ИС Internet Processor II анализирует битовую маску префикса F и определяет соответствие пакета наборам DCU 6 и 14. ИС Internet Processor II увеличивает счетчики пакетов и байтов наборов DCU 6 и 14, связанные с интерфейсом 3. Затем продолжает обработку пакета, выполняя выходной фильтр *Filter Mayo*.

Административная масштабируемость решений биллинга и учета

Услуги Интернета должны масштабироваться в двух направлениях. Первое – это техническое масштабирование, где проблема заключается в создании систем, выполняющих постоянно усложняющиеся задачи обработки пакетов без заметного ухудшения производительности форвардинга. Второе направление – это административное масштабирование, для которого необходимы средства формирования услуг и выставления счетов за пользование этими услугами. Успех услуги зависит от обоих направлений, т.к. еще до продажи услуги необходимо иметь механизм, с помощью которого она будет оплачиваться.

Далее мы рассмотрим административную масштабируемость решений биллинга и учета Juniper Networks.

Учет по интерфейсам

С административной точки зрения учет по интерфейсам легко адаптируется к изменению характера трафика. После задания параметров клиентского интерфейса маршрутизатор автоматически поддерживает статистику, которая описывает объемы данных, принимаемых и передаваемых клиентом по данному интерфейсу. Это самое простое учетное решение, но оно не дает достаточной информации для поддержки многоуровневых систем биллинга.

Учет на основе фильтров

Учет на основе фильтров прекрасно административно масштабируется до тех пор, пока не меняется информация о префиксах. Он великолепно работает при подсчете трафика к любому адресу на основе байта DiffServ (например, для различения трафика реального времени и трафика класса best-effort) или при учете трафика к сетям с постоянной конфигурацией. Решение Juniper Networks для учета на основе фильтров весьма привлекательны для таких приложений, т.к. на наших

маршрутизаторах подсчет фильтров не оказывает существенного влияния на производительность форвардинга.

Однако административная масштабируемость исчезает, когда начинается постоянное изменение сетевых префиксов. Это связано с тем, что провайдеру необходимо знать конкретный префикс, для которого пишется фильтр, а также с трудностями отслеживания добавления и удаления маршрутов в сетях других провайдеров и с необходимостью постоянного обновления фильтров для учета этих изменений. Учет на основе фильтров требует постоянного вмешательства человека для поддержания соответствия правил маршрутизации в изменяющейся сетевой среде.

Учет с использованием протокола MPLS

Динамическая природа отображения префиксов на путь LSP позволяет учету на основе протокола MPLS достичь некоторой степени гибкости, обеспечиваемой механизмом DCU. Если вещатель BGP объявляет некоторое множество путей, то для продвижения трафика к этим префиксам будет использован путь LSP к вещателю BGP. Правила маршрутизации дают провайдеру возможность управлять выбором префиксов, отображаемых на каждый путь LSP, контролируя объявление маршрутов вещателями BGP. Таким образом, средства учета на базе MPLS обеспечивают административную масштабируемость как в сетях, использующих стек с одной меткой (single label-stack environment) (для регулирования трафика), так и в сетях, где стек состоит из нескольких меток (multiple label-stack environment) (для ВЧС на базе MPLS).

Учет по классам назначения (DCU)

Учет на базе механизма DCU разрабатывался для обеспечения административного масштабирования в условиях флуктуации префиксов. В данном случае учет напрямую координируется с правилами маршрутизации. Это важное преимущество, т.к. DCU не требует от администратора посвящать все время синхронизации правил маршрутизации с правилами фильтрации или учета. Нет необходимости постоянно обновлять правила учета при добавлении или удалении маршрутов, при изменении сетевой топологии. Как только меняется объявление маршрута, происходит динамическое изменение правил учета, т.к. учет на базе DCU связан с процессом поиска маршрутов, а не с конкретным интерфейсом маршрутизатора. Только Juniper Networks удалось добиться таких функций учета, благодаря гибкости обработки и производительности форвардинга, обеспечиваемым IC Internet Processor II .

Ограничениями учета на базе механизма DCU является присвоение учетных наборов только по адресу назначения в заголовке пакета, а также то, что ОС JUNOS в данный момент не поддерживает учет трафика, передаваемого в противоположном направлении. Но программируемые возможности IC Internet Processor II и гибкость ОС JUNOS позволяют нам продолжать работы по совершенствованию механизма DCU и превращению его в инструмент обработки пакетов общего назначения.

Учетные профили Juniper Networks

Выше мы обсуждали ограничения использования SNMP для переноса больших объемов учетной информации от маршрутизаторов, собирающих эту информацию, к системам сетевого управления, ее обрабатывающим. Вспомним три основных ограничения.

- Формат сообщений SNMP требует большого объема служебной информации для передачи небольших значений 32- или 64-разрядных счетчиков.
- SNMP транспортный протокол прикладного уровня с нулевым окном, очень медленно переносящий данные между агентами и менеджерами.
- Генерация сообщений SNMP требует значительных ресурсов ЦПУ как на управляющей платформе, так и на управляемых элементах.

В начале 90-х, когда маршрутизаторы провайдеров имели не так много интерфейсов, SNMP являлся приемлемым инструментом сбора биллинговой и учетной информации. Но для высокочисленных маршрутизаторов, число логических интерфейсов, которых может исчисляться тысячами, SNMP абсолютно неэффективен по нескольким причинам.

- Для поддержки одного устройства с несколькими тысячами интерфейсов протокол SNMP требует большого количества блоков данных GetRequest и GetResponse, значительной доли ресурсов ЦПУ управляющей платформы и управляемых элементов, большой полосы пропускания для передачи сообщений, а также огромной работы по осмыслению всех данных.
- Не будем забывать и о последствиях потери информации вследствие невозможности для сборщика SNMP связаться с управляемыми элементами из-за сетевых проблем. Невозможность приема и передачи блоков GetRequest и GetResponse означает потерю учетной информации, продолжение приращения счетчиков и их закливание, т.е. невозможность выставления счетов за услуги.

Для преодоления ограничений SNMP при работе с высокопроизводительными маршрутизаторами Juniper Networks разработала так называемые *учетные профили (accounting profiles)*. Учетные профили – это механизм, позволяющий администратору конфигурировать маршрутизатор для сбора учетной информации и записывать ее вместе с временными метками в плоский файл на локальном диске. Теперь сетевой администратор может выбрать учетные файлы и импортировать предварительно обработанные данные в систему согласования и биллинга. Для клиентов, предпочитающих получать данные без предварительного опроса можно сконфигурировать учетный профиль для пересылки файлов на указанную станцию биллинга или сбора информации (см. рис. 11).

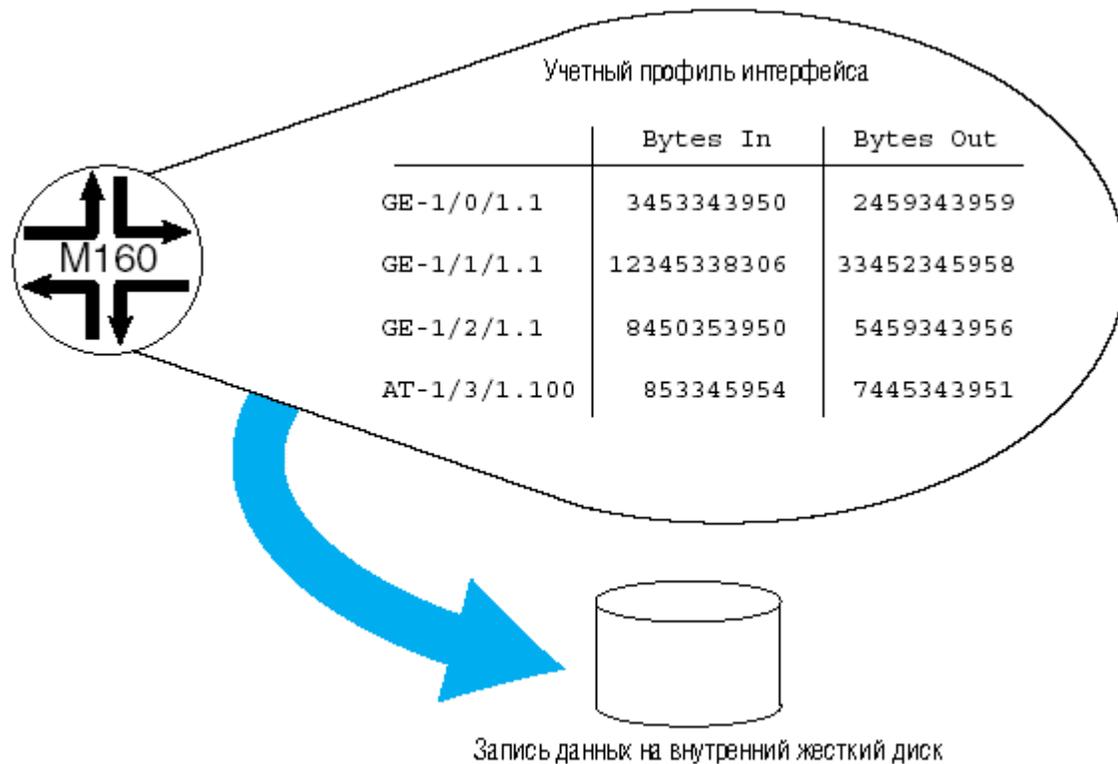


Рисунок 11. Учетный профиль интерфейса

Учетный профиль определяет характеристики собираемых данных, включая интервал сбора, имя файла статистики, максимальный размер файла, количество перезаписываемых файлов, собираемые поля и номера счетчиков. ОС JUNOS поддерживает учетные профили трех типов.

- Профиль интерфейса собирает информацию об указанных ошибках и статистику интерфейсов и суб-интерфейсов маршрутизатора.
- Профиль фильтра собирает счетчики пакетов и байтов для указанных фильтров межсетевого экрана.
- Профиль DCU собирает счетчики пакетов и байтов для указанных счетчиков интерфейсов DCU.

Использование учетных профилей позволяет преодолеть многие ограничения SNMP при работе с высокочастотными маршрутизаторами доступа.

- Снижение количества передаваемых блоков SNMP и соответствующего объема служебной информации повышает эффективность использования полосы пропускания.
- Высвободившиеся ресурсы ЦПУ используются для обработки других задач.
- Низкую производительность SNMP можно преодолеть, используя протоколы с переменным окном, функционирующие поверх TCP, такие как FTP или SCP (secure shell copy), специально разработанные для надежной передачи больших объемов данных.

- Наконец, если система управления теряет связь с маршрутизатором, на котором установлен учетный профиль, сборщик может получить учетную информацию после восстановления связи, т.е. вероятность потери учетной информации сведена к минимуму.

Заключение

В данной статье мы описали идеальное учетное решение с точки зрения Интернет-маршрутизации. Совершенно ясно, что один инструмент не может удовлетворить требования всех учетных приложений, принимая во внимание различные компромиссы, необходимые при сборе статистической информации для разных приложений из разных участков сети (имеется в виду объем передаваемой информации, требования по надежности, предварительная обработка или немедленная передача исходных данных, статистика по интерфейсам или общая статистика, подсчет всех пакетов или статистическая выборка, временное хранение на маршрутизаторе с последующей передачей всего объема или немедленная передача данных).

Возможно, единственным способом реализации идеального решения будет создание набора инструментов, специализирующихся на выполнении отдельных функций идеального решения. Чем больше гибкости предоставит производитель, тем больше инструментов сможет использовать администратор для сбора статистики и передачи ее в систему управления. Действительно гибкий подход позволяет рассмотреть конкретные требования для каждого учетного приложения и затем выбрать наилучшую комбинацию из всего спектра учетных инструментов.

В таблице 1 по атрибутам идеального учетного решения сравниваются характеристики всех учетных механизмов, поддерживаемых маршрутизаторами Juniper Networks и ОС JUNOS.

Таблица 1. Сбор учетной информации

Идеальное учетное решение	Учетные механизмы Juniper Networks			
	Учет по интерфейсам	Учет по фильтрам	Учет по MPLS	Учет по DCU
Статистика по объему трафика	Да	Да	Да	Да
Статистика по типу трафика		Да		Да
Статистика по источнику трафика		Да		
Статистика по назначению трафика		Да	Да	Да
Адаптируемость к динамической маршрутной среде			Хорошо	Отлично
Влияние на производительность форвардинга пакетов	Минимальное	Минимальное	Минимальное	Минимальное

В таблице 2 по атрибутам идеального учетного решения сравниваются характеристики всех учетных механизмов, поддерживаемых маршрутизаторами Juniper Networks и ОС JUNOS с точки зрения переноса учетной информации к системе управления. Несмотря на ограничения SNMP, мы продолжаем его использовать, давая провайдерам свободу выбора, так же мы поддерживаем большинство систем сетевого управления.

Таблица 2. Перенос учетной информации

Идеальное учетное решение	Сценарии Send / Expect, командная строка	JUNOScriptAPI	SNMP	Учетные профили
Предварительная обработка или агрегация статистики	Нет	Нет	Нет	Да
Передача статистики без значительной загрузки сети	Да	Да	Нет	Да
Надежная передача статистики управляющим станциям	Да	Да	Нет	Да

Мы уверены, что наши аппаратные средства учета очень близко подходят по своим характеристикам к идеальному учетному решению. Больше нет необходимости решать проблемы учета в ограниченных рамках, т.к. уже сегодня доступны реальные альтернативы традиционным программным средствам. Решение проблемы учета с помощью микросхем высокой степени интеграции позволило Juniper Networks фундаментально изменить природу учетных механизмов. Мощь и гибкость ОС JUNOS и IC Internet Processor II позволяют нам продолжать развитие этих механизмов, не заставляя наших клиентов модернизировать свою аппаратуру для использования их преимуществ.

Copyright © 2002 Poplar systems

Poplar
SYSTEMS

<http://www.poplar.ru>