Network Working Group
Request for Comments: 3654
Category: Informational

H. Khosravi, Ed.
T. Anderson, Ed.
Intel

November 2003

Требования к разделению управления и пересылки IP

Requirements for Separation of IP Control and Forwarding

Статус документа

Этот документ содержит информацию для сообщества Internet. Документ не задает каких-либо стандартов Internet. Распространение документа не ограничивается.

Авторские права

Тезисы

Этот документ определяет архитектуру разделения уровней управления и пересылки ForCES[1] и связанную с ней терминологию. Документ также определяет набор требований к архитектуре, моделированию и протоколу для логического разделения уровней управления и пересылки IP (IPv4, IPv6) в сетевых устройствах.

Оглавление

1. Введение

Элементы сетей IP состоят из множества логически разделенных частей, которые совместно используются для обеспечения заданной функциональности (например, маршрутизации или коммутации IP), а для внешнего наблюдателя представляются единым элементом. Существует два основных типа таких элементов - компоненты уровня управления (control-plane) и компоненты уровня пересылки (forwarding-plane). В общем случае компонентами уровня пересылки служат специализированные контроллеры ASIC, сетевые процессоры или устройства на процессорах общего назначения, которые выполняют все операции пути передачи данных. Компоненты уровня управления, напротив, обычно включают процессоры общего назначения, которые обеспечивают такие функции управления, как обработка протоколов маршрутизации или сигнализации. Нужен стандартный набор механизмов для связи между этими компонентами, который обеспечит лучшее масштабирование и позволит развивать уровни управления и пересылки независимо один от другого, что ускорит разработку и внедрение новинок.

Рассмотрим в качестве иллюстрации архитектуру маршрутизатора, чтобы показать концепцию разделения уровней управления и пересылки. Архитектура маршрутизатора включает две основных части. Эти взаимосвязанные компоненты выполняют функции, которые также тесно связаны между собой. Нижняя часть пути пересылки работает на уровне пересылки данных и отвечает за обработку и пересылку каждого пакета. Над уровнем пересылки размещается сетевая операционная система, которая отвечает за операции уровня управления. В случае маршрутизатора или коммутатора сетевая ОС обеспечивает протоколы маршрутизации, сигнализации и управления (например, RIP, OSPF и RSVP), определяя поведение уровня пересылки путем манипуляций с таблицами пересылки, таблицами QoS для потоков и списками контроля доступа (ACL[2]). Обычно архитектура таких устройств объединяет всю эту функциональность в единое целое с точки зрения внешнего наблюдателя.

[1]Forwarding and Control Element Separation.
[2]Access control list.

2. Определения

Addressable Entity (AE) – адресуемый объект (элемент)

Физическое сетевое устройство, которое непосредственно адресуется в данной технологии соединения. Например, в сетях IP - это устройства, к которым можно обращаться по адресу IP, а в машине коммутации (switch fabric) - это устройства, к которым можно обращаться по номеру порта.

Physical Forwarding Element (PFE) – физический элемент пересылки

Элемент AE, который включает оборудование, используемое для обработки и обслуживания каждого пакета. Это оборудование может включать сетевые процессоры, ASIC, линейные платы с множеством микросхем, автономные устройства с процессорами общего назначения и др.

Physical Control Element (PCE) – физический элемент управления

Элемент AE, который включает оборудование, используемое для обеспечения функций управления. Обычно это оборудование включает процессор общего назначения.

Forwarding Element (FE) – элемент пересылки

Логический элемент, реализующий протокол ForCES. Элементы FE используют базовое оборудование для обработки каждого пакета и управляются (контролируются) одним или множеством CE по протоколу ForCES. FE может быть отдельным элементом (или PFE), частью PFE или множеством PFE.

Control Element (CE) – элемент управления

Логический объект, который реализует протокол ForCES и инструктирует один или множество FE по части обработки пакетов. Функциональность CE включает исполнение протоколов управления и сигнализации. CE может состоять из частей PCE или целых PCE.

Pre-association Phase – фаза до объединения

Интервал времени, в течение которого менеджер FE (см. ниже) и менеджер CE (см. ниже) определяют каким FE и CE следует быть частью одного сетевого элемента. Все разделение PFE и PCE происходит на этом этапе.

Post-association Phase – фаза после объединения

Интервал времени, в течение которого FE знает управляющие им устройства CE и наоборот, включая интервал, в течение которого CE и FE организуют связи между собой.

ForCES Protocol – протокол ForCES

Хотя в архитектуре ForCES может применяться множество протоколов, термин «протокол ForCES» относится лишь к протоколу ForCES фазы после объединения (см. ниже).

ForCES Post-Association Phase Protocol – протокол ForCES после объединения

Протокол, используемый в коммуникациях между CE и FE после объединения. Этот протокол не применяется для коммуникаций CE-CE, FE-FE или между менеджерами FE и CE. Протокол ForCES работает в режиме «ведущий-ведомый» (master-slave), где FE являются ведомыми, а CE - ведущими. Этот протокол включает управление коммуникационным каналом (например, организацию соединения, обмен heartbeat) и сами управляющие сообщения. Протокол может представлять единое целое или набор совместно работающих протоколов.

FE Model – модель элемента пересылки

Модель, описывающая функции логической обработки в FE.

FE Manager – менеджер FE

Логический элемент, которые работает в фазе до объединения и отвечает за определение CE, с которыми элементу FE следует взаимодействовать. Этот процесс называется обнаружением CE и может включать определение менеджером FE возможностей доступных CE. Менеджер FE может применять все, что угодно от статической конфигурации до протокола фазы до объединения (см. ниже) для определения используемого CE. Однако протокол фазы до объединения выходит за рамки документа. Будучи логическим устройством, менеджер FE может быть физически объединен с любыми логическими элементами, упомянутыми в этом разделе.

CE Manager (CEM) – менеджер элементов управления

Логический объект, который отвечает за генерацию базовых задач управления CE. Используется, в частности, на этапе до объединения (pre-association phase) для определения FE, с которым CE следует взаимодействовать. Этот процесс называется обнаружением FE и может включать определение менеджером CE возможностей доступных FE. Менеджер CE может использовать все, что угодно от статической конфигурации до протокола фазы до объединения (см. ниже) для определения используемых FE. Протокол фазы до объединения выходит за рамки документа. Будучи логическим элементом, менеджер CE может быть физически объединен с любыми из логических элементов, упомянутых в этом разделе.

Pre-association Phase Protocol – протокол фазы до объединения

Протокол взаимодействия менеджеров FE и CE для определения используемых элементов CE и FE. Этот протокол может включать механизм определения возможностей CE и/или FE. Отметим, что этот процесс определения возможностей полностью отделен (и не служит заменой) от процесса, используемого протоколом ForCES (см. требование 1 в разделе 6). Однако эти два механизма определения возможностей могут использовать одну модель FE (см. раздел 5). Протокол фазы до объединения в этом документе не рассматривается.

ForCES Network Element (NE) – элемент сети ForCES

Объект, состоящий из одного или множества CE и одного или множества FE. Для внешних наблюдателей NE представляется единой точкой управления. Обычно NE скрывает свою внутреннюю структуру от внешних наблюдателей.

ForCES Protocol Element – элемент протокола ForCES

FE или CE.

High Touch Capability – работа с верхними уровнями

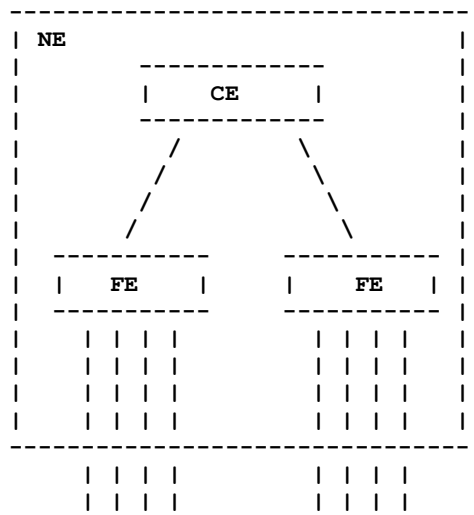
Этот термин обозначает возможности некоторых устройство пересылки (forwarder) выполнять операции над содержимым или заголовками пакетов, на основе данных, не входящих в заголовок IP. Примерами таких возможностей служат NAT-PT, межсетевое экранирование, распознавание содержимого L7.

3. Архитектура

Основными компонентами архитектуры NE являются элементы CE, FE и протокол соединения (interconnect) между ними. CE отвечает за сигнализацию и обработку протокола управления, а также за реализацию протоколов администрирования. На основе информации, полученной при обработке данных управления элементы CE задают поведение пересылки пакетов элементами FE с помощью протокола соединения. Например, CE может управлять элементами FE, меняя таблицы пересылки, состояния интерфейсов, а также добавляя или удаляя привязки NAT.

FE работают на уровне пересылки и отвечают за обработку и обслуживание каждого пакета. Благодаря независимому развитию уровней управления и пересылки, можно разрабатывать разнотипные FE как общего назначения, так и более специализированные. Функции, которые могут выполнять FE, включают пересылку на сетевом уровне, измерение, формирование (shaping), межсетевое экранирование, NAT, инкапсуляцию (например, туннелирование), декапсуляцию, шифрование, учет и т. п. Почти все комбинации этих функций присутствуют в реальных FE.

На рисунке показан пример NE, включающего CE и два FE. Элементы FE и CE требуют минимальной настройки в процессе предварительного задания конфигурации, которую можно выполнить с помощью менеджеров FE и CE, соответственно. Других ролей для менеджеров FE и CE не предусмотрено. Эти компоненты выходят за рамки архитектуры и требований к протоколу ForCES, которые включают элементы CE и FE.



4. Архитектурные требования

Ниже перечислены архитектурные требования к разделению уровней пересылки и управления.

1) Элементы CE и FE **должны** быть способны соединяться между собой на основе различных технологий. Примерами технологий соединения, используемых в современных архитектурах, являются Ethernet, шинные магистрали и ATM (ячейки). FE **могут** соединяться между собой на основе технологии, отличающейся от используемой в соединениях между CE и FE.

2) Элементы FE **должны** поддерживать минимальный набор возможностей, требуемых для организации сетевых соединений (например, обнаружение интерфейсов, функции включения и выключения портов). Сверх этого архитектуре ForCES **недопустимо** ограничивать типы и число возможностей, поддерживаемых FE.

3) Пакеты **должны** иметь возможность поступать в NE через один элемент FE и уходить через другой FE.

4) Элемент NE **должен** поддерживать представление себя в виде одного функционального устройства. Например, в маршрутизаторе значение TTL в пакете следует декрементировать при прохождении через NE только один раз, независимо от числа вовлеченных элементов FE. Однако внешние элементы (например, менеджеры FE и CE) **могут** иметь прямой доступ к отдельным элементам протокола ForCES для предоставления информации, позволяющей перейти из фазы до объединения (pre-association) в фазу после объединения (post-association).

5) Архитектура **должна** обеспечивать способ предотвращения несанкционированного присоединения элементов протокола ForCES к NE (более подробно это рассмотрено в требовании 2 раздела 6).

6) Элементы FE **должны** быть способны асинхронно информировать CE об отказе, а также изменении доступных ресурсов и возможностей FE. Таким образом, FE **должны** поддерживать мониторинг и уведомления (поскольку нет взаимно-однозначного соответствия между FE и PFE, возможно изменение отношений между FE и его физическими ресурсами с течением времени). Например, может меняться число физических портов или объем выделенной FE памяти. Элемент CE должен получать информацию о таких изменениях для обеспечения аккуратного управления элементами FE.

7) Архитектура **должна** поддерживать механизмы резервирования (избыточности) или отказоустойчивости CE. Это включает возможность элементов CE и FE обнаруживать потерю связи между ними, восстанавливать эту связь и эффективно (ре)синхронизировать состояние. Это также включает возможность заранее установить действия FE при потере связи со своим CE (например, продолжение пересылки пакетов или прекращение работы).

8) Элементы FE **должны** быть способны перенаправлять пакеты (например, сообщения RIP, OSPF), полученные на своих интерфейсах элементу CE. Они **должны** также перенаправлять другие относящиеся к делу пакеты (например, пакеты с опцией Router Alert Option) своим CE. Элементы CE **должны** быть способны настроить перенаправление пакетов на элементах FE. Элементы CE **должны** быть способны генерировать пакеты и предоставлять их своим FE для доставки.

9) Любая предлагаемая архитектура ForCES **должна** объяснять, как она будет поддерживать все функции маршрутизации, определенные в [RFC1812]. Должны быть разъяснены такие функции пересылки IPv4, как проверка пригодности заголовков IP, реализация алгоритма поиска самого длинного соответствия, декрементирование TTL, расчет контрольных сумм, генерация сообщений ICMP об ошибках и другие, определенные в RFC 1812 функции.

10) В ForCES NE элементы CE **должны** быть способны определять топологию соединений между элементами FE в NE.

11) Архитектура ForCES NE **должна** быть способна поддерживать по меньшей мере сотни FE и десятки тысяч портов.

12) Архитектура ForCES **должна** позволять элементам FE **и** CE динамически присоединяться к NE и покидать их.

13) Архитектура ForCES NE **должна** поддерживать множество CE и FE. Однако координация между элементами CE выходит за рамки ForCES.

14) Для фазы до объединения при организации, мониторинге и настройке конфигурации **может** оказаться полезным использование стандартных механизмов администрирования CE и FE. Архитектура и требования к ForCES не препятствуют этому. В общем случае для фазы после объединения большую часть задач администрирования **следует** решать путем взаимодействия с CE. В некоторых случаях (например, при потере связи между CE и FE) может оказаться полезным использование инструментов управления (например, SNMP) для диагностики и устранения проблем. При этом **должны** выполняться приведенные ниже рекомендации.

    1. Возможности средств управления (например, SNMP), служащие для чтения (но не изменения) данных состояния FE **не следует** запрещать.

    2. **Недопустимо** разрешать средствам управления (например, SNMP) изменять состояние FE так, что это будет воздействовать на поведение NE в целом, без уведомления CE.

5. Требования к модели FE

Разнообразие функций FE, разрешаемых архитектурой ForCES, создает потенциальную проблему для элементов CE. Для эффективного управления FE элемент CE должен понимать, как FE обрабатывает пакеты. Поэтому **требуется** создание модели FE, которая будет выражать возможности логической обработки пакетов в FE. Эта модель будет применяться в протоколе ForCES для описания возможностей FE (требование 1 в разделе 6). Модель FE **должна** определять модели возможностей и состояний, которые отражают текущую конфигурацию устройства. Модель FE **должна** также поддерживать множество элементов FE в архитектуре NE.

5.1. Типы логических функций

Модель FE **должна** показывать, какие логические функции могут быть применены к пакетам при их прохождении через FE. Логические функции - это функции обработки пакетов, применяемые при пересылке пакета через FE. Примерами таких функция являются пересылка на сетевом уровне, межсетевое экранирование, NAT, формирование трафика (shaping). В параграфе 5.5 определен минимальный набор логических функций, которые модель FE **должна** поддерживать.

5.2. Вариации логический функций

Модель FE **должна** быть способна разрешать (поддерживать) вариации способов реализации логических функций в FE. Например, в некоторых FE логическая функция пересылки может иметь адреса IP и MAC следующего интервала (next hop), а в других FE это может быть реализовано в разных логических функциях. Другим примером могут служить разновидности функции NAT - Traditional/Outbound NAT, Bi-directional NAT, Twice NAT, Multihomed NAT [RFC2663]. Модель должна быть достаточно гибкой, чтобы разрешать такие вариации в функциях.

5.3. Порядок логических функций

Модель **должна** быть способна описать порядок применения логических функций в FE, который во многих случаях важен. Например, функция NAT может изменять IP-адреса отправителя или получателя, которые могут применяться другими логическими функциями (пересылка L3, фильтрация на входе или выходе, формирование и учет трафика) для принятия решений. Элемент CE должен знать, следует использовать в таких функциях адреса до трансляции или после нее. Кроме того, модель **должна** быть способна выразить множество экземпляров одной функции в пути обработки FE. Здесь снова может послужить примером трансляция NAT, один экземпляр которой может применяться до принятия решения о пересылке (замена публичных адресов в пакетах извне на внутренние адреса сети), а другой после принятия такого решения (замена внутренних адресов на публичные для пакетов наружу).

5.4. Гибкость

Моделям FE **следует** обеспечивать гибкую инфраструктуру, в которую могут добавляться новые логические функции, а также классификации, действия и данные параметризации. Кроме того, модель FE **должна** быть способна описать типы статистики, собираемой каждой логической функцией.

5.5. Минимальный набор логических функций

В этом параграфе определяется минимальный набор логических функций, которые **должна** поддерживать каждая модель FE. Этот минимальный набор **не предполагает**, что все FE должны обеспечивать эту функциональность, требования лишь указывают, что модель должна быть способна выразить возможности, из которых элементы FE могут выбирать поддерживаемый ими набор.

1) Функции портов

   Модель FE **должна** быть способна выразить число портов устройства, статические атрибуты каждого порта (например, тип порта, скорость линии) и его настраиваемые атрибуты (например, IP-адрес, административный статус).

2) Функции пересылки

   Модель FE **должна** быть способна выразить данные, которые могут использоваться функцией пересылки для принятия решений. Модель **должна** обеспечивать поддержку функций индивидуальной и групповой пересылки для IPv4 и IPv6.

3) Функции QoS

   Модель FE **должна** позволять FE выразить свои возможности QoS с точки зрения функций измерения, применения правил, формирования и управления очередями. Модель FE **должна** быть способна выразить применение этих функций для обеспечения функциональности IntServ или DiffServ в соответствии с [RFC2211], [RFC2212], [RFC2215], [RFC2475] и [RFC3290].

4) Базовые функции фильтрации

   Модель FE **должна** быть способна выразить комплексные наборы функций фильтрации, а также присутствие этих функций в любой точке процесса обработки пакетов в FE. Модель FE **должна** быть способна выразить широкий набор возможностей классификации от одного поля (например, адрес получателя) до произвольных наборов полей. Модель FE **должна** быть способна выразить действия, которые эти функции классификации могут применять для соответствующих пакетов.

5) Фирменные функции производителей

   Модели FE **следует** быть расширяемой для выражения новой, не известной в настоящий момент функциональности FE. Модель FE **не следует** расширять для поддержки стандартных (общих) функций с использованием фирменных (proprietary) решений. Это **не будет** отвечать требованиям ForCES.

6) Функции high touch

   Модель FE **должна** быть способна выразить возможности инкапсуляции и туннелирования элементов FE. Модель FE **должна** поддерживать функции маркировки класса обслуживания, который следует обеспечить для пакетов (например, октет TOS в заголовке IPv4 или IPv6 Traffic Class). Модель FE **может** поддерживать другие функции high touch (например, NAT, ALG).

7) Функции защиты

   Модель FE **должна** быть способна выразить типы шифрования, которое может быть применено к пакетам на пути пересылки.

8) Отложенные (Off-loaded) функции

   Обработка каждого пакета может сохранять состояние в FE, чтобы логические функции в процессе обработки могли выполняться согласованно (например, каждый пакет может обновлять состояние «маркерного ковша» - token bucket). Кроме того, модель FE **должна** разрешать асинхронное выполнение логических функций в соответствии с той или иной машиной конечных состояний для того чтобы можно было, например, выполнить функции «выгруженные» в FE элементом управления CE. Модель FE **должна** быть способна выразить такие асинхронные функции. Примеры таких функций включают выполнение операций машины конечных состояний при прерывании сессии TCP или обработке OSPF Hello, вызываемой не только связанными с пакетами событиями, но и таймерами. Это не означает выгрузку (off-loading) какого-либо кода в FE, просто модели FE следует обеспечивать возможность выразить существующие отложенные функции в FE.

9) Функции IPFLOW/PSAMP

   Некоторые приложения (например, учет использования и формирования трафика) требуют от элементов сети измерений на уровне потокола IP. [IPFLOW] определяет архитектуру мониторинга, измерения и экспорта для потоков трафика IP. Модели FE **следует** обеспечивать возможность выразить функции измерения и учета для экспорта данных о потоках трафика IP. Для поддержки основанных на измерениях приложений [PSAMP] описывает модель определения стандартного набора возможностей элементов сети по выборке подмножества пакетов на основе статистических или иных методов. Модели FE **следует** обеспечивать возможность выражать функции статистической фильтрации пакетов и информацию, требуемую для поддержки приложений выборки пакетов.

6. Требования к протоколу ForCES

Данная спецификация задает некоторые требования, которые протокол ForCES **должен** выполнять.

1) Настройка элементов по модели

   Протокол ForCES **должен** разрешать элементам CE определять возможности каждого FE. Эти возможности **нужно** указывать с использованием модели FE, требования к которой определены в разделе 5. Кроме того, протокол **должен** обеспечивать элементам CE способ управления всеми возможностями FE, определенными из модели FE. Протокол **должен** быть способен добавлять и удалять записи для событий и классификации, устанавливать и удалять параметры, запрашивать статистику и регистрироваться для получения событий.

2) Поддержка защищенных коммуникаций

   a) Конфигурация FE содержит информацию, играющую важную роль в работе сети (например, таблицы пересылки IP). Поэтому **должна** обеспечиваться защита целостности всех сообщений протокола ForCES, а также защита от MITM-атак[1].

   b) Конфигурационные данные FE могут также включать информацию, полученную из деловых соглашений (например, SLA[2]). Поскольку такая информация является конфиденциальной, **должна** обеспечиваться защита конфиденциальности (приватности) всех сообщений протокола ForCES.

   c) Для обеспечения участия в работе NE только уполномоченных элементов CE и FE и защиты от подмены CE или FE, архитектура ForCES **должна** выбрать способы проверки подлинности CE и FE.

   d) В некоторых реализациях ForCES предполагается, что связанные между собой элементы CE и FE размещаются на одной плате (backplane) и физическая защита корпуса предоставращает MITM-атаки,

[1]Man-in-the-middle — перехват и изменение данных в пути с участием человека.
[2]Service level agreement — соглашение об уровне обслуживания.

перехват и подмену. В таких случаях архитектура ForCES **может** полагаться на физическую защиту устройства и протокольные механизмы защиты можно отключить.

e) В тех случаях, когда элементы CE и FE соединены через сеть, **должны** быть заданы механизмы защиты протокола ForCES от упомянутых выше атак. Любое решение, используемое для защиты ForCES, **должно** указывать способы защиты от таких атак.

3) Расширяемость

Протокол ForCES **должен** быть способен поддерживать по меньшей мере сотни FE и десятки тысяч портов. Например, размеры полей протокола ForCES, соответствующих FE или номерам портов, **нужно** делать достаточно большими для поддержки требуемого числа элементов или портов. Эти требования не связаны с производительностью NE по мере роста числа портов или FE.

4) Связь через маршрутизаторы

Когда элементы CE и FE разделены более, чем одним интервалом маршрутизации L3 (hop), протокол ForCES будет использовать соответствующий требованиям RFC 2914 существующий протокол L4 с подходящими гарантиями, защитой и контролем перегрузок (например, TCP или SCTP) в качестве транспортного протокола.

5) Приоритет сообщений

Протокол ForCES **должен** обеспечивать возможность задать приоритет протокольных сообщений.

6) Надежность

a) Протокол ForCES будет применяться для доставки информации, которая требует различный уровней гарантии. Строгой или надежной гарантией здесь считается отсутствие повреждений и изменения порядка доставки информации, а также своевременность такой доставки.

b) Некоторые данные (payload) типа перенаправленных пакетов или выборки пакетов могут не требовать гарантий доставки (быть устойчивыми к некоторому уровню потерь). Для информации такого сорта протокол ForCES **недопустимо** ограничивать строгими гарантиями.

c) Такие данные, как конфигурационная информация (например, ACL, записи FIB или информация о возможностях FE, упомянутая в требовании 1 раздела 6), критичны для работы и должны доставляться с надежными гарантиями. Поэтому для такой информации протокол ForCES **должен** обеспечивать встроенные механизмы или использовать транспортный протокол с гарантиями доставки.

d) Для таких типов информации типа, как пакеты heartbeat, которые могут применяться для обнаружения потери связности между CE и FE (см. требование 8 в разделе 6), своевременность доставки может оказаться более предпочтительной, чем гарантии. Для информации такого сорта протокол ForCES **недопустимо** ограничивать строгими гарантиями.

e) Когда протокол ForCES работает через сеть с множеством интервалов маршрутизации IP, ForCES **должен** использовать соответствующий требованиям [RFC2914] транспортный протокол.

f) В случаях, когда протокол ForCES не работает через сеть IP (например, сеть Ethernet или коммутация ячеек между CE и FE), гарантии **должны** обеспечиваться при передаче важной информации, отмеченной выше в п. c), за счет встроенных механизмов протокола или нижележащих уровней.

7) Независимость от технологии соединений

Протокол ForCES **должен** поддерживать различные технологии для соединений между элементами (см. требование 1 в разделе 4).

8) Избыточность или отказоустойчивость CE

Протокол ForCES **должен** поддерживать механизмы резервирования CE или восстановления при отказе CE. Это включает возможность элементов CE и FE определять потерю связи между ними, возможность восстанавливать связь и эффективные механизмы (ре)синхронизации. Это включает также возможность заранее задать действия FE в ответ на потерю связи со своим CE - например, продолжение пересылки пакетов или прерывание работы (см. п. 7 в требованиях раздела 4).

9) Перенаправление и «зеркалирование» пакетов

a) Протокол ForCES **должен** определять способ перенаправления пакетов от FE к CE и наоборот. Перенаправление пакетов прерывает дальнейшую обработку пакета в FE.

b) Протокол ForCES **должен** определять способ отображения (mirror) пакетов от FE к CE. Отображение позволяет элементу FE дублировать пакет в точке «отражения» для передачи его элементу CE с продолжением обработки в FE.

Примеры пакетов, которые могут перенаправляться или отображаться, включают пакеты управления (например, сообщения RIP или OSPF), адресованные на интерфейсы, или другие относящиеся к делу пакеты (например, пакеты с опцией Router Alert Option). Протокол ForCES **должен** также определять для CE способ настройки поведения в описанных выше случаях a) и b) для выбора пакетов в каждом из них.

10) Обмен топологической информацией

Протокол ForCES или переносимая им информация **должны** позволять элементам FE, имеющим топологические данные о соединениях между FE, предоставлять такую информацию элементам CE.

11) Динамические соединения (ассоциации)

Протокол ForCES **должен** позволять элементам CE и FE динамически подключаться к NE и выходить из него (требование 12 в разделе 4).

12) Группировка команд

Протокол ForCES **должен** быть способен группировать упорядоченный набор команд для FE. Каждую такую группу команд **следует** передавать FE с использованием минимально возможного числа сообщений. Кроме того, протокол **должен** поддерживать возможность указать, **должна** ли группа использовать семантику «все или ничего» (all-or-nothing).

13) Асинхронные уведомления о событиях

Протокол ForCES **должен** быть способен асинхронно уведомлять CE о событиях в элементах FE, таких как отказ или изменение доступных ресурсов или возможностей (требование 6 в разделе 4).

14) Статистика запросов

Протокол ForCES **должен** обеспечивать для CE возможность запроса статистики (мониторинг производительности) у элементов FE.

15) Защита от атак на отказ службы (DoS), связанных с перегрузкой процессора или переполнением очередей

Система, использующая протокол ForCES, может быть атакована путем перегрузки CPU или переполнения очередей. Протокол ForCES может использоваться в таких атаках для того, чтобы лишить CE возможности управления FE или требуемого для работы взаимодействия с другими маршрутизаторами или системами. Поэтому протокол ForCES **должен** обеспечивать механизмы контроля возможностей FE, которые могут применяться для защиты от таких атак. Возможности FE, которые **должны** управляться через ForCES, включают способность устанавливать классификаторы и фильтры для детектирования и отбрасывания пакетов атаки, а также способность ограничивать скорость пакетов, которые представляются корректными, но могут оказаться частью атаки (например, ложные пакеты BGP).

7. Литература

7.1. Нормативные документы

[RFC3290] Bernet, Y., Blake, S., Grossman, D. and A. Smith, "An Informal Management Model for DiffServ Routers", RFC 3290, May 2002.

[RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, September 1997.

[RFC2212] Shenker, S., Partridge, C. and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC2215] Shenker, S. and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, September 1997.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weisss, "An Architecture for Differentiated Service", RFC 2475, December 1998.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 14, RFC 2914, September 2000.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

7.2. Дополнительная литература

[RFC3532] Anderson, T. and J. Buerkle, "Requirements for the Dynamic Partitioning of Switching Elements", RFC 3532, May 2003.

[IPFLOW] Quittek, et al., "Requirements for IP Flow Information Export", Work in Progress[1], February 2003.

[PSAMP] Duffield, et al., "A Framework for Passive Packet Measurement ", Work in Progress, March 2003.

8. Вопросы безопасности

См. требование 5 для архитектуры и требование 2 для протокола.

9. Адреса авторов и благодарности

Этот документ был написан командой ForCES Requirements, включающей:

Todd A. Anderson (редактор)

Ed Bowen

IBM Zurich Research Laboratory

Saumerstrasse 4

CH-8803 Rueschlikon Switzerland

Phone: +41 1 724 83 68

EMail: edbowen@us.ibm.com

[1]Работа завершена и опубликована в RFC 3917. *Прим. перев.*

Ram Dantu
Department of Computer Science
University of North Texas,
Denton, Texas, 76203
Phone: 940 565 2822
EMail: rdantu@unt.edu

Avri Doria
ETRI
161 Gajeong-dong, Yuseong-gu
Deajeon 305-350 Korea
EMail: avri@acm.org

Ram Gopal
Nokia Research Center
5, Wayside Road,
Burlington, MA 01803
Phone: 1-781-993-3685
EMail: ram.gopal@nokia.com

Jamal Hadi Salim
Znyx Networks
Ottawa, Ontario
Canada
EMail: hadi@znyx.com

Hormuzd Khosravi (редактор)

Muneyb Minhazuddin
Avaya Inc.
123, Epping road,
North Ryde, NSW 2113, Australia
Phone: +61 2 9352 8620
EMail: muneyb@avaya.com

Margaret Wasserman
Nokia Research Center
5 Wayside Road
Burlington, MA 01803
Phone: +1 781 993 3858
EMail: margaret.wasserman@nokia.com

Авторы благодарят Vip Sharma и Lily Yang за их значимый вклад в работу.

10. Адреса редакторов

Hormuzd Khosravi
Intel
2111 NE 25th Avenue
Hillsboro, OR 97124 USA
Phone: +1 503 264 0334

EMail: hormuzd.m.khosravi@intel.com

Todd A. Anderson

Intel

2111 NE 25th Avenue

Hillsboro, OR 97124 USA

Phone: +1 503 712 1760

EMail: todd.a.anderson@intel.com

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

11. Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.