

## A PCE-Based Architecture for Application-Based Network Operations

Основанная на PCE архитектура ABNO

### Аннотация

Такие службы, как распространение содержимого, распределенные базы данных и связность между ЦОД, могут вносить новые требования к работе сетей. Им нужно резервирование связности, надёжности и ресурсов (таких как пропускная способность) по потребности в зависимости от приложений (таких как соединения «точка-точка», виртуализация сетей, транзит мобильной связи) для разных сетевых технологий от пакетных (IP/MPLS) до оптических. Среды, работающие с такими требованиями, называют основанными на приложениях сетевыми операциями (Application-Based Network Operations или ABNO). ABNO объединяет множество существующих технологий и может рассматриваться как использование инструментария имеющихся компонентов, дополненного новыми элементами.

В этом документе описывается архитектура и модель для ABNO с рассмотрением совместной работы компонентов. Это по сути «поваренная книга» об имеющихся технологиях, соответствующих потребностям приложений.

### Статус документа

Этот документ не является спецификацией стандарта Internet и публикуется с информационными целями.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 5741.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <http://www.rfc-editor.org/info/rfc7491>.

### Авторские права

Авторские права (с) 2015 принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно.

## Оглавление

1. Введение.....	2
1.1. Область действия.....	3
2. Сетевые операции на основе приложений.....	3
2.1. Допущения.....	3
2.2. Реализация архитектуры.....	3
2.3. Базовая архитектура ABNO.....	3
2.3.1. Компоненты ABNO.....	4
2.3.1.1. NMS и OSS.....	4
2.3.1.2. Координатор прикладных служб.....	4
2.3.1.3. Контроллер ABNO.....	4
2.3.1.4. Агент правил.....	4
2.3.1.5. Клиент интерфейса с системой маршрутизации (I2RS).....	5
2.3.1.6. Обработчик OAM.....	5
2.3.1.7. Элемент расчёта пути (PCE).....	5
2.3.1.8. Базы данных.....	5
2.3.1.8.1. База данных организации трафика (TED).....	5
2.3.1.8.2. База данных LSP.....	5
2.3.1.8.3. Базы данных SRLG.....	6
2.3.1.8.4. Другие базы данных.....	6
2.3.1.9. Сервер ALTO.....	6
2.3.1.10. Диспетчер топологии виртуальной сети (VNTM).....	6
2.3.1.11. Диспетчер обеспечения.....	6
2.3.1.12. Уровни сети клиента и сервера.....	6
2.3.2. Функциональные интерфейсы.....	7
2.3.2.1. Интерфейсы настройки и программирования.....	7
2.3.2.2. Создание TED по данным из сети.....	7
2.3.2.3. Улучшение TED.....	7
2.3.2.4. Представление TED.....	7

<sup>1</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>2</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

2.3.2.5. Запросы расчёта пути из сети.....	8
2.3.2.6. Управление сетями с помощью диспетчеров обеспечения.....	8
2.3.2.7. Аудит сети.....	8
2.3.2.8. Управление системой маршрутизации.....	9
2.3.2.9. Интерфейс контроллера ABNO с PCE.....	9
2.3.2.10. Интерфейс VNTM с PCE.....	9
2.3.2.11. Управляющие интерфейсы ABNO.....	9
2.3.2.12. Запросы обеспечения ABNO.....	9
2.3.2.13. Интерфейсы политики.....	9
2.3.2.14. OAM и отчёты.....	9
3. Варианты применения ABNO.....	10
3.1. Связность между AS.....	10
3.2. Многоуровневая сеть.....	12
3.2.1. Соединение ЦОД через многоуровневые сети.....	14
3.3. Работа без прерывания.....	15
3.3.1. Работа без прерывания при повторной оптимизации.....	15
3.3.2. Работа без прерывания при восстановлении.....	15
3.3.3. Работа без прерывания при проверке и выборе пути.....	16
3.4. Глобальная одновременная оптимизация.....	17
3.4.1. Пример использования GCO с MPLS LSP.....	17
3.5. Адаптивное управление сетью (ANM).....	18
3.5.1. Включение ANM.....	18
3.5.2. Обработка запроса на расчёт GCO.....	18
3.5.3. Автоматизированный процесс обеспечения.....	19
3.6. Операции псевдопроводов и управление ими.....	19
3.6.1. Многосегментные псевдопровода.....	19
3.6.2. Псевдопровода с разными путями.....	20
3.6.3. Многосегментные псевдопровода с разными путями.....	20
3.6.4. Защита сегментов псевдопровода.....	21
3.6.5. Применимость ABNO для псевдопроводов.....	21
3.7. Межслойная оптимизация (CSO).....	21
3.7.1. Работа ЦОД.....	21
3.7.1.1. Перенос виртуальных машин.....	22
3.7.1.2. Распределение нагрузки.....	22
3.7.2. Применение архитектуры ABNO.....	22
3.7.2.1. Внедрённые приложения, службы и продукция.....	23
3.8. Сервер ALTO.....	23
3.9. Другие возможные варианты применения.....	24
3.9.1. Обслуживание трафика.....	24
3.9.2. Планирование пропускной способности.....	24
4. Живучесть и резервирование в рамках архитектуры ABNO.....	24
5. Вопросы безопасности.....	25
6. Вопросы управляемости.....	25
7. Литература.....	25
Приложение А. Не определённые интерфейсы.....	27
Благодарности.....	27
Участники работы.....	28
Адреса авторов.....	28

## 1. Введение

Современные сети объединяют множество технологий, позволяющих инфраструктуре сети предоставлять разнообразные услуги для поддержки различных характеристик и потребностей приложений. Растёт потребность в реагировании сети на запросы служб, вводимые напрямую с уровня приложений. Это отличается от устоявшейся модели, где услуги предоставляются сетью в ответ на команды управления, вводимые человеком.

Управляемые приложениями запросы и организуемые по ним службы вносят ряд новых требований к работе сетей. Они требуют резервирования сетевой связности, надёжности и ресурсов (таких как пропускная способность) в соответствии с потребностями и с учётом специфики приложения для различных сетевых приложений (например, соединений «точка-точка», виртуализации сетей, транзита мобильной связи) в широком диапазоне технологий — от пакетных (IP/MPLS) до оптических. Среды, работающие с такими требованиями, называют основанными на приложениях сетевыми операциями (ABNO).

Был разработан элемент расчёта пути (Path Computation Element или PCE) [RFC4655] для предоставления услуг по расчёту путей в сетях с управлением GMPLS и MPLS. Применимость PCE может быть расширена для предоставления возможностей расчёта пути и исполнения правил для платформ и служб ABNO.

ABNO может предоставлять приложениям указанные ниже типы услуг, координируя компоненты, которые работают и управляют сетью.

- Оптимизация потоков трафика между приложениями с целью создания наложенной сети для взаимодействия в таких вариантах применения, как совместное использование файлов, кэширование или отражение данных, потоковая передачи или взаимодействия в реальном масштабе времени, описанных как оптимизация трафика прикладного уровня (Application-Layer Traffic Optimization или ALTO) [RFC5693].
- Удалённое управление компонентами сети, позволяющее координировать программирование ресурсов сети с помощью таких методов, как разделение элементов управления и пересылки (Forwarding and Control Element Separation или ForCES) [RFC3746], OpenFlow [ONF], интерфейс с системой маршрутизации (Interface to the Routing System или I2RS) [I2RS-Arch] или плоскость управления, координируемую по протоколу взаимодействия PCE (PCE Communication Protocol или PCEP) [PCE-Init-LSP].

- Взаимосвязь сетей доставки содержимого (Content Delivery Networks или CDN) [RFC6707] путём организации и управления (resizing) соединениями между такими сетями. Аналогичным путём ABNO может координировать соединения между ЦОД.
- Координация ресурсов сети для автоматизации предоставления и облегчения обслуживания трафика, планирования пропускной способности и глобальной одновременной оптимизации с использованием RCEP [RFC5557].
- Планирования виртуальных частных сетей (Virtual Private Network или VPN) для поддержки развёртывания новых клиентов VPN и облегчения связности между ЦОД.

В этом документе описаны архитектура и примеры применения ABNO, а также показано, как можно использовать архитектуру ABNO для координации запросов системы управления и приложений на расчёт путей, применения правил и управления ресурсами сети в интересах использующих сеть приложений. Рассмотрение примеров использования показывает архитектуру ABNO как инструментарий, включающий множество имеющихся компонентов и протоколов, поэтому документ походит на поваренную книгу. ABNO совместима с развёрнутыми системами управления сетью (Network Management System или NMS) и поддержки операций (Operations Support System или OSS), а также с более современными разработками в области программируемых сетей, такими как программно-определяемые сети (Software-Defined Networking или SDN).

## 1.1. Область действия

Этот документ описывает инструментарий, показывая, как описанные во множестве документов функциональные компоненты можно собрать вместе в рамках единой архитектуры для обеспечения функций, требуемых ABNO. Во многих случаях известно, что имеющиеся протоколы достаточно или почти достаточно подходят для выполнения требований к интерфейсам между компонентами. В таких случаях протоколы считаются пригодными для применения в реализации ABNO. В других случаях очевидно, что нужна дополнительная работа и для них приводятся указатели на текущие работы, которые могут быть полезны. Для случаев, когда такие работы не известны авторам, в Приложении А даны краткие описания отсутствующих интерфейсных протоколов.

Таким образом, документ можно считать заявлением о применимости существующих протоколов и рекомендациями для разработчиков новых протоколов и расширений.

## 2. Сетевые операции на основе приложений

### 2.1. Допущения

Основным допущением, лежащим в основе этого документа, является то, что следует применять имеющиеся технологии там, где они подходят для решения соответствующих задач. Кроме того, когда имеющейся технологии почти достаточно, предполагается, что лучше внести в неё незначительные расширения, чем создавать новую.

Отметим, что этот документ описывает архитектуру. Функциональные компоненты являются концепциями архитектуры и имеют свои чёткие зоны ответственности. Пары функциональных компонентов взаимодействуют через функциональные интерфейсы, которые поэтому тоже являются концепциями архитектуры.

### 2.2. Реализация архитектуры

Необходимо подчеркнуть, что этот документ описывает функциональную архитектуру, а не устройство программ. Поэтому не предполагается, что архитектура будет ограничивать реализации. Разделение функций ABNO по отдельным функциональным компонентам с чёткими интерфейсами между ними позволит разработчикам выбирать включаемые функции и распределять их между разными процессами и даже процессорами. Реализация этой архитектуры может принимать некоторые важные решения о функциональных компонентах, как указано ниже.

- Несколько функциональных компонентов можно объединить в одном программном компоненте с раскрытием лишь внешних интерфейсов. Это может обеспечить явные преимущества для быстрых путей в программах и снижать издержки взаимодействия между процессами. Например, можно реализовать активный элемент PCE с учётом состояний как один сервер, объединяющий компоненты ABNO PCE, базы данных организации трафика (Traffic Engineering Database или TED), базу данных о путях с коммутацией по меткам (Label Switched Path Database) и диспетчер обеспечения (Provisioning Manager, см. параграф 2.3).
- Функциональные компоненты можно распределить между процессами, процессорами или серверами, чтобы интерфейсы раскрывались как внешние протоколы. Например обработчик OAM<sup>1</sup> (см. параграф 2.3.1.6) можно представить как выделенный сервер в сети, который воспринимает из сети все отчёты о состоянии, агрегирует и сопоставляет их, затем отправляет уведомления другим серверам, которым нужно понимать происходящее.
- Каждый компонент может присутствовать в нескольких экземплярах, т. е. функции компонента могут быть разделены между несколькими программными компонентами, каждый из которых отвечает за обработку определённой функции или части сети. Например, в реализации может быть несколько экземпляров TED (см. параграф 2.3.1.8), каждый из которых содержит сведения о топологии отдельного домена сети (такого как сетевой уровень или автономная система). Может применяться несколько экземпляров PCE, каждый из которых обрабатывает свою базу TED и может быть распределён по нескольким серверам с разным управлением. Ещё одним примером может быть несколько контроллеров ABNO, каждый из которых поддерживает свой класс приложений или прикладных служб.

Описание архитектуры предназначено для упрощения реализаций с обеспечением функциональной совместимости между реализациями ключевых компонентов и простым взаимодействием с приложениями и устройствами сети.

### 2.3. Базовая архитектура ABNO

Архитектура ABNO показана на рисунке 1, а компоненты и функциональные интерфейсы рассматриваются в параграфах 2.3.1 и 2.3.2, соответственно. Примеры использования, приведённые в разделе 3, показываются отдельно

<sup>1</sup>Operations, Administration, and Maintenance - операции, администрирование и поддержка.

как применяются различные компоненты для предоставления разных услуг. Важно понимать, что показанные на рисунке связи и интерфейсы между компонентами иллюстрируют некоторые базовые и вероятные взаимодействия, однако не исключены и другие интерфейсы и взаимосвязи, требуемые для реализации конкретной функциональности.

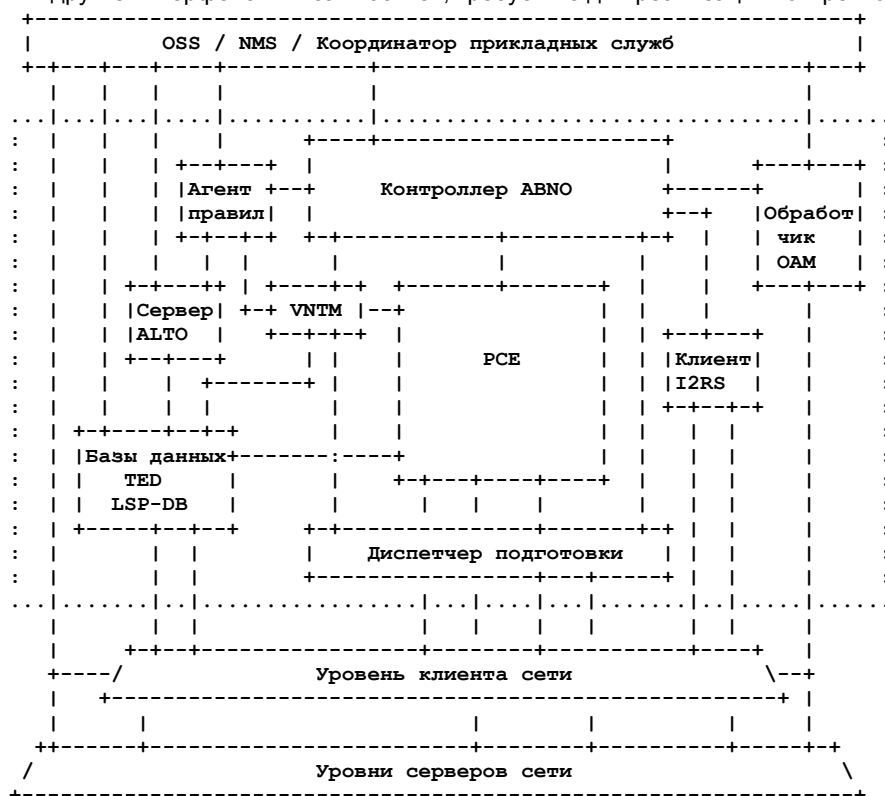


Рисунок 1. Базовая архитектура ABNO.

### 2.3.1. Компоненты ABNO

В этом параграфе описываются функциональные компоненты, показанные прямоугольниками на рисунке 1. Взаимодействия между этими компонентами (функциональные интерфейсы) описаны в параграфе 2.3.2.

#### 2.3.1.1. NMS и OSS

Система управления сетью (NMS) или система поддержки операций (OSS) может служить для контроля, эксплуатации и управления сетью. В архитектуре ABNO система NMS или OSS может передавать высокоуровневые запросы обслуживания контроллеру ABNO, а также организовывать правила для действий компонентов в архитектуре.

NMS и OSS могут воспринимать сетевые события через обработчик OAM и действовать в соответствии с этим, а также выводить отчёты пользователям и генерировать сигналы. NMS и OSS могут обращаться к базе данных организации трафика (TED) и базе данных о путях с коммутацией по меткам (Label Switched Path Database или LSP-DB), чтобы показать пользователям текущее состояние сети. Кроме того, NMS и OSS могут использовать прямой программный или конфигурационный интерфейс для взаимодействия с элементами сети.

#### 2.3.1.2. Координатор прикладных служб

Помимо NMS и OSS, услуги в архитектуре ABNO могут запрашиваться приложениями или от их имени. В этом контексте термин «приложение» трактуется очень широко и может пониматься как программа, работающая на хосте или сервере и предоставляющая услуги пользователю (например, программа для видеоконференций), программный инструмент, применяемый пользователем для отправки в сеть запросов на организацию конкретной услуги (например, сквозное соединение или плановое резервирование пропускной способности), многофункциональная система управления, отвечающая за организацию предоставления комплексной сетевой услуги, такой как виртуальная частная сеть.

В рамках этой архитектуры все эти концепции приложения собраны воедино в координаторе прикладных служб (Application Service Coordinator), поскольку все они в той или иной мере отвечают за координацию действий сети по обеспечению услуг для использования приложениями. На практике координатор прикладных служб может быть распределён между разными приложениями или серверами. Координатор взаимодействует с контроллером ABNO для запроса сетевых операций.

#### 2.3.1.3. Контроллер ABNO

Контроллер ABNO является главным шлюзом в сеть для NMS, OSS и контроллера прикладных служб с целью обеспечения расширенной координации и функций сети. Контроллер ABNO управляет поведением сети ответ на изменение сетевых условий и в соответствии с требованиями приложений к сети и правилами. Он является точкой присоединения и вызывает нужные компоненты в нужном порядке. Примеры использования в разделе 3 дают более чёткое представление о взаимодействии контроллера ABNO с другими компонентами архитектуры ABNO.

#### 2.3.1.4. Агент правил

Политика (правила) играют важную роль в поддержке сетей и управлении ими, а поэтому оказывает значительное влияние на работу ключевых компонентов архитектуры ABNO. На рисунке 1 показан агент правил (Policy Agent), как компонент, в котором NMS/OSS задаёт правила для применения. Агент отвечает за распространение правил в другие компоненты системы. Для упрощения рисунка пришлось опустить многие из происходящих взаимодействий политики.

Хотя показано взаимодействие агента лишь с контроллером ABNO, сервером ALTO и диспетчером топологии виртуальной сети (Virtual Network Topology Manager или VNTM), на деле он взаимодействует с другими компонентами и самими элементами сети. Например, элемент расчёта пути (PCE) будет точкой применения правил (Policy Enforcement Point или PEP) [RFC2753], как описано в [RFC5394], а клиент интерфейса в систему маршрутизации (I2RS) будет также PEP, как отмечено в [I2RS-Arch].

### 2.3.1.5. Клиент интерфейса с системой маршрутизации (I2RS)

Интерфейс с системой маршрутизации (I2RS) описан в [I2RS-Arch] и обеспечивает программируемый способ доступа (чтение и запись) к состоянию маршрутизации и сведениям о правилах на маршрутизаторах сети. Клиент I2RS предложен в [I2RS-PS] и его назначение состоит в управлении запросами сведений на множестве маршрутизаторов сети (где работает агент I2RS) и координации установки и сбора состояний на маршрутизаторах.

### 2.3.1.6. Обработчик OAM

Операции, администрирование и обслуживание (OAM) играют очень важную роль в понимании работы сети, обнаружения отказов и принятии мер по реагированию на проблемы в сети.

В архитектуре ABNO обработчик OAM отвечает за приём уведомлений (их часто называют сигналами тревоги — alert) из сети о возможных проблемах, сопоставление их и инициирование других компонентов системы для выполнения действий по сохранению или восстановлению служб, организованных контроллером ABNO. Обработчик OAM также сообщает о проблемах в сети (в частности, о влияющих на службы проблемах) NMS, OSS и координатору прикладных служб. Кроме того, обработчик OAM взаимодействует с устройствами сети для инициирования действий OAM в плоскости данных (таких как мониторинг и тестирование).

### 2.3.1.7. Элемент расчёта пути (PCE)

Элемент расчёта пути (PCE) определён в [RFC4655] и является функциональным компонентом, обслуживающим запросы на расчёт пути по графу сети. В частности, PCE генерирует маршруты с организацией трафика для путей с коммутацией по меткам (Label Switched Path или LSP) в MPLS-TE и GMPLS. PCE может принимать запросы от контроллера ABNO, диспетчера топологии виртуальной сети (Virtual Network Topology Manager) или самих элементов сети. PCE работает с представлением топологии сети, хранящимся в базе данных организации трафика (TED). Более сложные расчёты может предоставлять PCE с учётом состояний (Stateful PCE), дополняющий TED базой данных, содержащей сведения о LSP, которые подготовлены и работают в сети (LSP-DB, см. параграф 2.3.1.8.2), как описано в [RFC4655] и [Stateful-PCE].

Дополнительная функциональность активных PCE позволяет функциональным компонентам с Stateful PCE подавать запросы на организацию новых услуг или изменение имеющихся, как описано в [Stateful-PCE] и [PCE-Init-LSP]. Эта функция может обращаться к элементам сети напрямую или через диспетчер обеспечения (Provisioning Manager).

Координация между PCE, работающими с разными TED, может быть полезна для расчёта путей в многодоменных и многоуровневых сетях. Доменом в этом случае может быть автономная система (Autonomous System или AS), что позволяет рассчитывать пути между AS.

PCE является ключевым компонентом архитектуры ABNO и дополнительное представление о роли этих элементов можно получить из рассмотрения примеров, представленных в разделе 3.

### 2.3.1.8. Базы данных

Архитектура ABNO включает ряд баз данных, где хранятся сведения для использования системой. Основными базами данных являются базы TED и LSP (LSP-DB), но могут применяться и другие базы со сведениями о топологии (сервер ALTO), политике (агент правил), службах (контролёр ABNO) и т. п.

В последующем тексте выделены основные компоненты, использующие базы данных. Следует отметить, что базы доступны для проверки любым компонентам ABNO. Обновления баз данных следует обрабатывать с осторожностью, поскольку разрешение записи в них нескольким компонентам может вызывать проблемы связанные последовательностью действий и одновременными попытками записи.

#### 2.3.1.8.1. База данных организации трафика (TED)

TED - это хранилище данных о топологии сети, которые могут быть дополнены сведениями о возможностях (например, о метрике или пропускной способности) и активными данными о состоянии (например, up/down или оставшаяся свободная полоса). Базу TED можно создать на основе сведений из сети или от NMS/OSS (например, данные инвентаризации).

Основным применением TED в архитектуре ABNO является предоставление необработанных (raw) данных, с которыми работают элементы расчёта пути. TED может просматриваться пользователями NMS/OSS для определения текущего состояния сети и предоставлять сведения прикладным службам, таким как оптимизация трафика на прикладном уровне (ALTO) [RFC5693].

#### 2.3.1.8.2. База данных LSP

LSP-DB - это хранилище данных о LSP, которые организованы или могут быть созданы в сети, включая пути и использование ресурсов LSP. Базу LSP-DB можно создать из локально генерируемых сведений. Например, эта база обновляется при предоставлении LSP. Можно создать базу также на основе сведений из сети, получаемых путём опроса или считывания состояний LSP, которые уже организованы.

Основным применением LSP-DB в архитектуре ABNO является улучшение планирования и оптимизации LSP. Новые LSP можно создавать так, чтобы они не были связаны с путями (path-disjoint) других LSP (для поддержки защищённых служб). LSP можно перемаршрутизировать для оптимизации путей или освобождения ресурсов для других LSP. Возможен быстрый ремонт LSP при уведомлении о неполадках в сети или перенос LSP на другой путь для исключения ресурсов, которые планируется отключить для обслуживания. Основным потребителем LSP-DB являются Stateful PCE (параграф 2.3.1.7).

### 2.3.1.8.3. Базы данных SRLG

Базу TED можно дополнить сведениями о группах каналов с общим риском (Shared Risk Link Group или SRLG), которые содержат для каждого сетевого ресурса один или несколько идентификаторов, связывающих этот ресурс с другими ресурсами в той же базе TED, для которых существует такой же риск отказа.

Хотя эти данные могут быть очень полезны, их можно дополнить дополнительными подробными сведениями, поддерживаемыми в отдельной базе данных и индексируемыми по идентификатору SRLG из базы TED. Такая база данных может интерпретировать сведения SRLG из других сетей (например, сетей серверов), предоставлять вероятности отказов, связанные с каждой SRLG, предлагать приоритет, когда не удаётся найти пути, не связанные с SRLG, и сопоставлять SRLG разных серверных сетей или сетей партнёров.

### 2.3.1.8.4. Другие базы данных

В систему ABNO могут встраиваться другие базы данных, используемые в работе сети, например, сведения о потоках и потребностях в трафике, предсказанных и плановых запросах трафика, истории отказов и восстановления каналов и узлов, ресурсах сети, таких как метки пакетов и физические метки (MPLS и GMPLS) и т. п.

Как отмечено в параграфе 2.3.1.8.1, TED можно дополнить данными инвентаризации. Вполне вероятно, что во многих сетях такие сведения хранятся в отдельной базе данных (Inventory Database), которая включает информацию о производителе, модели, дате установки и т. п.

### 2.3.1.9. Сервер ALTO

Сервер ALTO предоставляет сведения о сети уровню приложений на основе абстрактных карт области сети. Эти данные представляются в упрощённом виде, но полезны для управления трафиком на уровне приложений. Службы ALTO помогают поставщикам услуг совместно использовать данные о сетевых местоположениях и стоимости путей между ними. Критерии выбора между местоположениями могут зависеть от таких сведений, как максимальная пропускная способность, минимальный трафик между доменами, стоимость для пользователя и т. п.

Сервер ALTO создаёт представления ALTO для совместного использования данных с координатором прикладных служб, чтобы тот мог выбирать лучшие пути в сети для передачи трафика прикладного уровня. Представления ALTO рассчитываются на основе сведений из сетевых баз данных, политики, заданной агентом правил (Policy Agent), по алгоритмам, используемым элементом PCE.

Базовый протокол ALTO [RFC7285] определяет, в частности, одноузловое абстрактное представление сети для координатора прикладных служб, состоящее из двух частей - карты сети и карты стоимости. Карта сети указывает множество задаваемых провайдером идентификаторов (Provider-defined Identifier или PID), представляющих точки входа в сеть. Каждый узел на прикладном уровне известен как конечная точка (End Point или EP), которой назначен PID, поскольку PID являются точками входа в сеть из приложений. Как указано в [RFC7285], PID может обозначать подсеть, набор подсетей, участок города, точку присутствия (Point of Presence или PoP) и т. п. Каждый такой регион сети может быть одним доменом или множеством сетей, это лишь представление, выдаваемое сервером ALTO уровню приложений. Карта стоимости указывает стоимость путей между EP и/или PID. Критерии используемые координатором прикладных служб для выбора между двумя местоположениями в сети зависят от атрибутов, таких как максимальная пропускная способность, минимальный междоменный трафик, стоимость для пользователя и т. п.

### 2.3.1.10. Диспетчер топологии виртуальной сети (VNTM)

Топология виртуальной сети (Virtual Network Topology или VNT) определена в [RFC5212] как один или множество LSP в одной или нескольких сетях, обеспечивающие сведения для эффективной обработки путей в сети верхнего уровня. Например, набор LSP в сети с мультиплексированием по длине волны (wavelength division multiplexed или WDM) может обеспечивать связность в качестве виртуальных каналов в вышележащей сети с коммутацией пакетов.

VNT расширяет число физических/выделенных каналов в сети верхнего уровня и настраивается путём создания или уничтожения нижележащих LSP и анонсирования изменений в вышележащую сеть. VNT можно приспособить к потребностям трафика, чтобы в сети верхнего уровня можно было создавать или освобождать пропускную способность по мере потребностей. Освобождение ненужных ресурсов VNT делает их доступными в сети нижележащего уровня для других применений.

Создание виртуальной топологии для включения в сеть является непростой задачей. Нужно принять решения о том, какие узлы верхнего уровня лучше подключить, в какой сети нижнего уровня обеспечивать LSP для связности и как маршрутизировать LSP в сети нижнего уровня. Кроме того, нужны определённые действия для обеспечения низкоуровневых LSP и анонсирования связности в сети вышележащего уровня.

В [RFC5623] описано, как диспетчер VNTM может создавать соединения на уровне серверов для поддержки связности на клиентском уровне. В архитектуре ABNO создание новых соединений можно передать менеджеру обеспечения, как описано в параграфе 2.3.1.11.

Все эти действия и решения в значительной мере зависят от политики, поэтому координирующий их компонент VNTM получает данные от агента правил (политики). Диспетчер VNTM тесно связан с PCE сети верхнего уровня и каждым PCE в сетях нижнего уровня.

### 2.3.1.11. Диспетчер обеспечения

Диспетчер обеспечения (Provisioning Manager) отвечает за подачу или доставку запросов на организацию LSP. Это могут быть инструкции для плоскости управления, работающей в сети, или программирование отдельных сетевых устройств. Во втором случае диспетчер обеспечения может выступать в качестве контроллера OpenFlow [ONF].

Взаимодействия между сетью и диспетчером обеспечения более подробно рассматриваются в параграфе 2.3.2.6.

### 2.3.1.12. Уровни сети клиента и сервера

Сети клиентов и серверов показаны на рисунке 1 как иллюстративные примеры использования архитектуры ABNO для координации услуг через несколько сетей, где сети нижнего уровня обеспечивают связность для сетей верхнего уровня. В параграфе 3.2 представлены варианты применения для многоуровневых сетей.

### 2.3.2. Функциональные интерфейсы

В этом параграфе описываются интерфейсы между функциональными компонентами, которые могут быть сделаны внешними в реализациях, разрешающих распределять компоненты по платформам. Когда имеющиеся протоколы могут обеспечивать все или большинство требуемых возможностей, это отмечается. В Приложении А указаны интерфейсы, для которых могут потребоваться дополнительные спецификации протоколов.

Как отмечено в начале параграфа 2.3, важно понимать, что взаимоотношения и интерфейсы между компонентами, показанные на рисунке 1, являются лишь иллюстрацией общих или вероятных взаимодействий. Однако на рисунке и в последующих описаниях не показаны другие интерфейсы и взаимосвязи, которые могут требоваться для реализации конкретных функций.

#### 2.3.2.1. Интерфейсы настройки и программирования

Сетевые устройства можно настраивать и программировать напрямую из NMS/OSS. Имеется много протоколов для решения этих задач, включая:

- SNMP [RFC3412];
- протокол настройки сети (Network Configuration Protocol или NETCONF) [RFC6241];
- RESTCONF [RESTCONF];
- общий протокол управления коммутаторами (General Switch Management Protocol или GSMP) [RFC3292];
- ForCES [RFC5810];
- OpenFlow [ONF];
- PCEP [PCE-Init-LSP].

Разработан стандарт TeleManagement Forum (TMF) Multi-Technology Operations Systems Interface (MTOSI) [TMF-MTOSI] для облегчения сетевого взаимодействия приложений и обеспечения возможности обнаруживать, настраивать и активировать ресурсы. Исходно информационная модель MTOSI могла представлять лишь ориентированные на соединения сети и ресурсы. В последующих выпусках была добавлена поддержка сетей без организации явных соединений (connectionless). Для NMS интерфейс MTOSI является северным и основан на web-услугах SOAP.

С точки зрения ABNO настройка сети является сквозной (pass-through) функцией. Это можно увидеть в левой части рисунка 1.

#### 2.3.2.2. Создание TED по данным из сети

Как указано в параграфе 2.3.1.8, база TED предоставляет подробные сведения о возможностях и состоянии сети для использования системой ABNO и PCE, в частности. Базу TED можно создать через участие в протоколах IGP-TE, работающих в сети (например, OSPF-TE [RFC3630] и IS-IS TE [RFC5305]). Можно также создать TED с использованием расширений BGP для распространения сведений о состоянии каналов [BGP-LS].

Система ABNO может поддерживать одну базу TED для множества сетей или использовать свою TED в каждой сети. Кроме того, сервер ALTO [RFC5693] может предоставлять абстрактную топологию сети для построения базы TED на прикладном уровне, которую PCE может использовать для расчёта путей между серверами и объектами прикладного уровня для предоставления прикладных служб.

#### 2.3.2.3. Улучшение TED

Базу TED можно улучшить данными инвентаризации от NMS/OSS, что дополняет сведения, полученные, как указано в параграфе 2.3.2.2, информацией, которая обычно не распространяется в сети, например, данными о типах и возможностях узлов или характеристиках оптических каналов. Протокол для этого интерфейса ещё не выбран, но протокол, разработанный или адаптированный в соответствии с требованиями к интерфейсу в систему маршрутизации (I2RS) [I2RS-Arch], может оказаться подходящим кандидатом для распространения объёмной информации о состоянии маршрутизации с чётко определенным языком представления. Другим кандидатом может быть протокол NETCONF [RFC6241], передающий данные с использованием языка YANG [RFC6020].

Отметим, что в общем случае любая комбинация протокола и кодирования, подходящая для представления TED, как описано в параграфе 2.3.2.4, подойдёт, вероятно, или может быть сделана подходящей для обеспечения возможности записи в TED как описано здесь.

#### 2.3.2.4. Представление TED

База TED может быть представлена на северном интерфейсе системы ABNO для использования в NMS/OSS или координаторе прикладных служб. Это позволит пользователям и приложениям получить представление топологии сети и статус сетевых ресурсов, а также планировать и предоставлять сетевые услуги. Имеется несколько протоколов, позволяющих экспортировать TED на северной границе.

- Протокол ALTO [RFC7285] предназначен для распространения абстрактной топологии, используемой сервером ALTO, и может быть полезен для экспорта TED. Сервер ALTO предоставляет стоимость пути между EP или PID, поэтому прикладной уровень может выбрать соединение, наиболее подходящее для обмена информацией между своими точками.
- Протокол, служащий для экспорта топологических сведений из сети, можно применять для экспорта топологии из TED [BGP-LS].
- Для I2RS [I2RS-Arch] потребуется протокол, способный обрабатывать обмен большими объёмами маршрутных сведений, который подойдёт и для экспорта TED. В этом случае имеет смысл стандартизовать представление TED на формальном языке моделирования данных, таком как YANG [RFC6020], чтобы можно было использовать имеющиеся протоколы, например, NETCONF [RFC6241] или расширяемый протокол обмена сообщениями и сведениями о присутствии (Extensible Messaging and Presence Protocol или XMPP) [RFC6120].

Отметим, что экспорт из TED может быть полным дампом содержимого (выраженным на подходящем языке абстракции), как описано выше, а также набором данных, агрегированным или отфильтрованным в соответствии с правилами или конкретными требованиями. Таким образом, связи, показанные на рисунке 1, могут быть несколько упрощёнными, поскольку контроллер ABNO может участвовать в подготовке и представлении сведений из TED на северном интерфейсе.

### 2.3.2.5. Запросы расчёта пути из сети

Как задаёт архитектура PCE [RFC4655], элементы сети могут подавать запросы на расчёт пути элементу PCE по протоколу PCEP [RFC5440]. Это упрощает создание в сети LSP в ответ на простые запросы связности и позволяет сети повторно оптимизировать и ремонтировать LSP.

### 2.3.2.6. Управление сетями с помощью диспетчеров обеспечения

Как указано в параграфе 2.3.1.11, диспетчер обеспечения подаёт или направляет запросы на предоставление ресурсов в сети. Эти операции могут происходить на двух уровнях: запросы на предоставление/настройку конкретных ресурсов в плоскостях данных или пересылки и запросы на инициирование набора действий, которые запрограммированы с помощью плоскости управления. Для предоставления ресурсов сети уже имеется много протоколов, указанных ниже.

- Программирование/настройка конкретных сетевых ресурсов
  - ForCES [RFC5810] определяет протокол для отделения элемента управления (диспетчер обеспечения) от элементов пересылки на каждом узле сети.
  - Обобщенный протокол управления коммутатором (General Switch Management Protocol или GSMP) [RFC3292] - это асимметричный протокол, позволяющий одному или нескольким внешним контроллерам коммутаторов (например, диспетчерам обеспечения) организовать и поддерживать состояние коммутатора по меткам (такого как MPLS).
  - OpenFlow [ONF] - это коммуникационный протокол, предоставляющий контроллеру OpenFlow (такому как диспетчер обеспечения) доступ к плоскости пересылки коммутатора или маршрутизатора в сети.
  - Исторически сложилось так, что для организации состояния пересылки/коммутации на отдельных узлах сети используются другие механизмы, основанные на конфигурации. Эти механизмы варьируются от нестандартных командных интерфейсов (command line interface или CLI) до различных стандартизованных вариантов вроде языка трансляции 1 (Translation Language 1 или TL1) [TL1] и SNMP [RFC3412]. Эти механизмы не предназначены для быстрых операций в сети и не обеспечивают простого программирования. Не предполагается их использование диспетчером обеспечения в архитектуре ABNO.
  - NETCONF [RFC6241] предоставляет более активный протокол настройки, который может подойти для массового программирования ресурсов сети. Использование протокола зависит от определения подходящих модулей YANG для требуемых вариантов. Ранние работы группы IETF NETMOD направлены на функции маршрутизации высокого уровня, сравнимые с функциями, рассмотренными в параграфе 2.3.2.8 (см. [YANG-Rtg]).
  - Спецификация [TMF-MTOSI] обеспечивает подготовку, активацию, деактивацию и освобождение ресурсов через интерфейс активации услуг (Service Activation Interface или SAI). Базовый коммуникационный транспорт (Common Communication Vehicle или CCV) - это программы промежуточного уровня (middleware) для реализации MTOSI. CCV применяется для обеспечения абстракции промежуточных программ в сочетании с языком описания услуг Web (Web Services Description Language или WSDL), чтобы можно было связать MTOSI с разными технологиями middleware по мере необходимости.
- Инициирование действий через плоскость управления
  - LSP можно запрашивать с помощью интерфейса системы управления головной части LSP, используя такие инструменты, как CLI, TL1 [TL1], SNMP [RFC3412]. Настройка с таким уровнем детализации не столь критична по времени, как при программировании отдельных сетевых ресурсов, поскольку основная задача программирования сквозной связности передаётся плоскости управления. Тем не менее, эти механизмы остаются неподходящими для программируемого управления сетью и не предлагаются для использования диспетчером обеспечения в архитектуре ABNO.
  - Как отмечено выше, NETCONF [RFC6241] обеспечивает более активный протокол настройки. Это может быть особенно подходящим для запросов на организацию LSP. Нужна работа по созданию модуля YANG.
  - Протокол взаимодействия PCE (PCEP) [RFC5440] был предложен для запросов организации LSP [PCE-Init-LSP] и работает достаточно хорошо, поскольку требуемые элементы протокола совпадают с применяемыми для откликов на запросы расчёта пути. Функциональный элемент, подающий запросы PCEP для организации LSPs называют активным PCE, однако следует отметить, что функциональным компонентом ABNO для запроса организации LSP является диспетчер обеспечения. Другие контроллеры, такие как VNTM и контроллер ABNO пользуются услугами диспетчера обеспечения для изоляции функций расчёта и запроса путей от механизмов обеспечения, использующихся в любой данной сети.

Отметим, что I2RS не предоставляет механизмов управления ресурсами сети на этом уровне, поскольку предназначен для управления состоянием маршрутизации в маршрутизаторах, а не состоянием пересылки в плоскости данных.

### 2.3.2.7. Аудит сети

После предоставления ресурсов или организации соединений важно, чтобы система ABNO могла определить состояние сети, а при изменении предоставленных ресурсов или выводе их из эксплуатации изменения в сети должны быть понятны системе ABNO. Эти функции делится на четыре категории, описанных ниже.

- Сбор обновлений базы TED, как описано в параграфе 2.3.2.2.



- Явные уведомления об успешном создании LSP и его последующем состоянии могут предоставляться с использованием расширений PCEP, как описано в [Stateful-PCE] и [PCE-Init-LSP].
- Может применяться OAM с проверкой результатов обработчиком OAM, как описано в параграфе 2.3.2.14.
- Ряд компонентов ABNO может подавать запросы и проверять состояние сети с помощью различных технологий, включая I2RS, NETCONF, SNMP.

### 2.3.2.8. Управление системой маршрутизации

Как обсуждалось в параграфе 2.3.1.5, интерфейс с системой маршрутизации (I2RS) обеспечивает программируемый способ доступа (чтение и запись) к данным состояния маршрутизации на маршрутизаторах сети. Клиент I2RS подаёт запросы маршрутизаторам сети для организации или извлечения состояния маршрутизации. Для этих запросов служит протокол I2RS, который будет основан на NETCONF [RFC6241] и RESTCONF [RESTCONF] с рядом дополнений.

### 2.3.2.9. Интерфейс контроллера ABNO с PCE

Контроллер ABNO должен иметь возможность обращаться к PCE для определения услуг, которые могут быть предоставлены в сети и нет причин отказываться для этого от стандартного протокола PCEP, заданного в [RFC5440].

### 2.3.2.10. Интерфейс VNTM с PCE

Между диспетчером топологии виртуальной сети (VNTM) и PCE имеется два вида взаимодействий.

1. Когда VNTM хочет определить, какие LSP можно организовать в сети, он использует стандартный интерфейс PCEP [RFC5440] для запроса расчёта пути.
2. Когда PCE понимает, что он не может выполнить расчёт запрошенного пути, или видит, что в сети (в соответствии с теми или иными заданными правилами) недостаточно ресурсов (например, пропускная способность того или иного канала близка к истощению), он может уведомить VNTM, который (в соответствии с правилами) может принять меры для создания дополнительной виртуальной топологии. Интерфейс для этого в настоящее время не задан, хотя может оказаться, что протокол, разработанный для I2RS, будет иметь подходящие функции (см. параграф 2.3.2.8). Другим вариантом может быть расширение сообщений PCEP Notify (PCNtf) [RFC5440].

### 2.3.2.11. Управляющие интерфейсы ABNO

Северный интерфейс контроллера ABNO используют NMS, OSS и координатор прикладных служб для запроса в сети услуг поддержки приложений. Этот интерфейс также должен быть способен сообщать об асинхронном завершении запросов услуг и изменениях в состоянии услуг. Интерфейс должен обладать строгими средствами защиты, проверки подлинности и соблюдения правил. В настоящее время этот интерфейс не задан. Это должен быть основанный на транзакциях интерфейс, поддерживающий спецификацию абстрактных услуг с достаточной гибкостью для простоты расширения в сочетании с компактностью и простотой синтаксического анализа. Протокол, разработанный для I2RS (см. параграф 2.3.2.8) может оказаться подходящим для этого.

### 2.3.2.12. Запросы обеспечения ABNO

При некоторых обстоятельствах контроллер ABNO может направлять запросы напрямую диспетчеру обеспечения. Например, если такой диспетчер служит контроллером SDN, контроллер ABNO может использовать один из API, определённых для запросов к контроллеру SDN (например, Floodlight REST API [Flood]). Поскольку диспетчер обеспечения может получать инструкции от Stateful PCE, в некоторых случаях может быть уместно использование расширений PCEP [PCE-Init-LSP].

### 2.3.2.13. Интерфейсы политики

Как отмечено в параграфе 2.3.1.4 и других местах этого документа, политика (правила) является важнейшим компонентом архитектуры ABNO. Роль политики заключается в соблюдении указанных ниже требований и правил.

- Добавление ресурсов по запросам должно предоставляться лишь полномочным функциям.
- Микропотокам клиентов не следует инициировать организацию или выделение на уровне сервера.
- Следует поддерживать возможности учёта.
- Требуются механизмы защиты для проверки полномочности запросов и возможностей (функций).

Связанная с правилами функциональность системы может включать правила маршрутизации и пересылки:

- поведение ECMP;
- классификация пакетов по LSP или категориям QoS.

Определены разные варианты архитектуры с поддержкой правил, включая схему для использования политики в системах с поддержкой PCE [RFC5394]. Однако принятый IETF протокол Common Open Policy Service (COPS) [RFC2748] не получил широкого распространения.

Потребуется работа по определению всех интерфейсов политики в архитектуре ABNO, а также по определению внутренних и внешних интерфейсов (последние нуждаются в протоколах). Обсуждается вопрос возможности поддержки протоколом I2RS настройки правил и манипуляций с ними.

### 2.3.2.14. OAM и отчёты

Обработчик OAM должен взаимодействовать с сетью для:

- включения функций OAM в сети;
- выполнения упреждающих операций OAM в сети;
- приёма уведомления о событиях в сети.

Для этого могут применяться любые интерфейсы настройки и программирования, описанные в параграфе 2.3.2.1. Уведомления NETCONF описаны в [RFC5277], OpenFlow поддерживает множество асинхронных уведомлений о событиях [ONF], протокол Syslog [RFC5424] передаёт отчёты о событиях в сети, а IP Flow Information Export (IPFIX) [RFC7011] позволяет агрегировать и передавать статистику сети.

Обработчик OAM сопоставляет полученные и сети сведения о событиях и передаёт их контроллеру ABNO, который может использовать их для восстановления предоставляемых услуг, а также NMS, OSS и координатору прикладных служб. Здесь подходит механизм, используемый для передачи из сети сведений о событиях и новых протокол не требуется, хотя для независимых от технологии передачи отчётов OAM могут потребоваться новые модели данных.

### 3. Варианты применения ABNO

В этом разделе приведены примеры использования архитектуры ABNO для управляемых приложениями и NMS/OSS сетевых операций. Примеры дают конкретный материал, демонстрирующий архитектуру, для облегчения её понимания и применения путём «профилирования» и выбора нужных компонентов и интерфейсов. Представленный набор примеров не содержит всех вариантов применения ABNO и предназначен для широкого охвата приложений, которые обычно рассматриваются, но не исключаются и другие варианты применения.

Описания в этом разделе не являются подробными заявлениями о применимости ABNO. Предполагается, что такие заявления для описанных здесь и иных случаев могут быть подходящим материалом для других документов.

#### 3.1. Связность между AS

Приведённый здесь пример описывает использование модели ABNO для организации сквозных услуг MPLS между автономными системами (AS). На рисунке 2 показана топология простой сети, где три AS (ASa, ASb, and ASc) соединены граничными маршрутизаторами (AS Border Router или ASBR) a1, a2, b1 - b4, c1, c2. Узел-источник (s) размещён в ASa и соединяется с адресатом (d), размещённым в ASc. Нужно рассчитать оптимальный путь для LSP от s к d, а затем инициировать в сети организацию LSP.

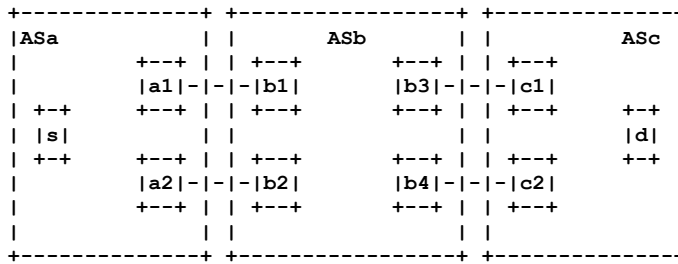


Рисунок 2. Топология связи между AS с иерархией PCE (Parent PCE).

Далее описываются этапы предоставления услуги в рамках архитектуры ABNO.

1. Управление запросами.

Как показано на рисунке 3, NMS/OSS подаёт контроллеру ABNO запрос для пути между s и d. Контроллер ABNO проверяет наличие у NMS/OSS права вносить такой запрос услуги.

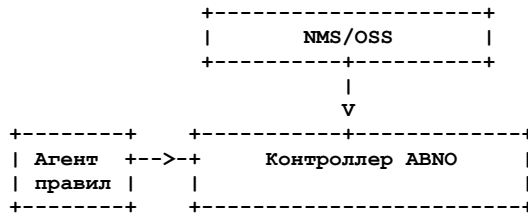


Рисунок 3. Управление запросом в ABNO.

2. Расчёт пути с иерархией PCE.

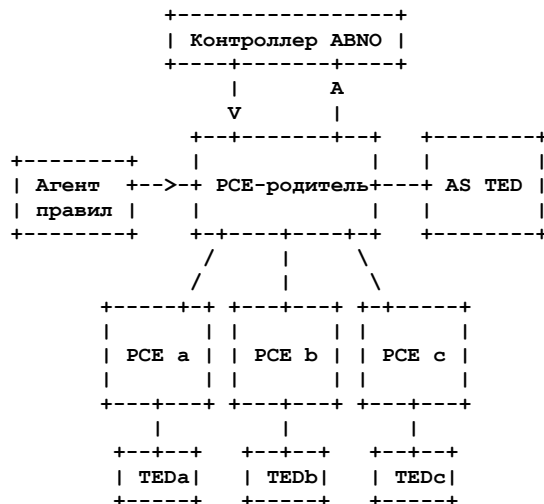


Рисунок 4. Запрос расчёта пути с иерархией PCE.

Контроллеру ABNO нужно найти сквозной путь для LSP. Поскольку AS хотя и поддерживают конфиденциальность сведений о своих внутренних ресурсах и топологии, они не будут использовать общую базу TED и в каждой будет свой PCE. Здесь применима архитектура Hierarchical PCE (H-PCE) [RFC6805].

Как показано на рисунке 4, контроллер ABNO передаёт запрос родительскому PCE для сквозного пути. Родительский PCE (см. [RFC6805]) обращается к базе TED, которая показывает связность между AS. Это помогает понять, что сквозной путь должен проходить через ASa, ASb и ASc, поэтому передаются отдельные запросы на расчёт пути каждому из PCE a, b и c на определения лучшего пути через AS.

Каждый из дочерних PCE применяет к полученному запросу правила для проверки его допустимости и выбора типов ресурсов сети, которые могут использоваться в результате расчёта. В целях конфиденциальности каждый из дочерних PCE может предоставлять отклик с использованием ключа пути [RFC5520] для сокрытия деталей рассчитанного сегмента. Родительский PCE сопоставляет отклики от дочерних и применяет свои правила для их объединения в лучший сквозной путь, который возвращается контроллеру ABNO.

### 3. Предоставление сквозного LSP

Имеется несколько вариантов обеспечения сквозного LSP в архитектуре ABNO, часть которых описана ниже.

#### 3a Обеспечение от контроллера ABNO с плоскостью управления

На рисунке 5 показано, как контроллер ABNO отправляет через диспетчер обеспечения запрос для организации сквозного LSP. Как описано в параграфе 2.3.2.6, для этого взаимодействия можно применить протокол NETCONF [RFC6241] или расширения PCEP из [PCE-Init-LSP]. В любом случае запрос на обеспечение передаётся головному маршрутизатору с коммутацией по меткам (Label Switching Router или LSR) а тот использует сигнализацию плоскости управления (например, с помощью RSVP-TE [RFC3209]) для инициирования организации LSP.



Рисунок 5. Предоставление сквозного LSP.

#### 3b Обеспечение путём программирования сетевых ресурсов

В этом варианте LSP организуется поэтапно (hop by hop) от диспетчера обеспечения с использованием таких механизмов, как ForCES [RFC5810] или OpenFlow [ONF] (см. параграф 2.3.2.6). Картина в этом случае совпадает с показанной на рисунке 5. Для взаимодействия между контроллером ABNO и диспетчером обеспечения применяется PCEP или NETCONF (как в варианте 3a), а за распределение запросов между отдельными элементами сети отвечает диспетчер обеспечения.

#### 3c Обеспечение с активным родительским PCE

Активный PCE (параграф 2.3.1.7) работает на основе концепций [PCE-Init-LSP]. В этом случае процесс варианта 3a изменяется так, что PCE напрямую передаёт команду PCEP в сеть без возврата отклика сначала контроллеру ABNO. Этот подход показан на рисунке 6 и может быть изменён так, чтобы диспетчер обеспечения продолжал программировать отдельные ресурсы сети как в варианте 3b.

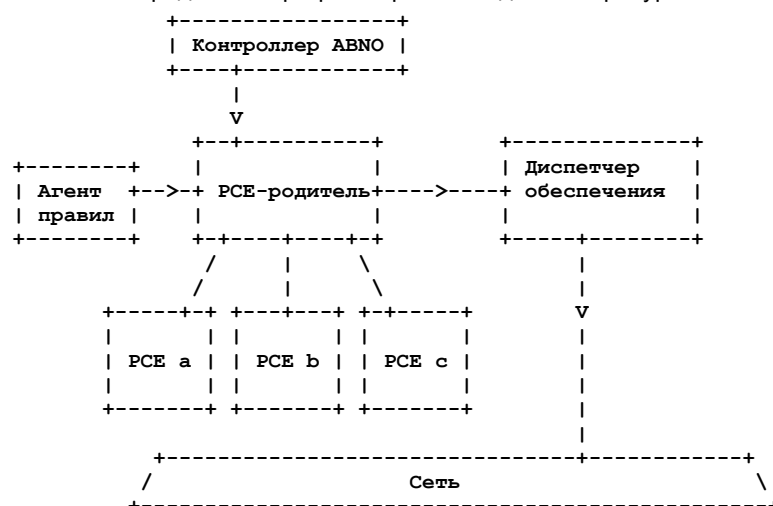


Рисунок 6. Разделение с активным PCE.

#### 3d Обеспечение с активными дочерними PCE и сборкой сегментов

Сочетание вариантов 3b и 3c может создавать комбинацию механизмов программирования сети для обеспечения сквозного LSP. На рисунке 7 показано, как каждый из дочерних PCE может стать активным и отвечать за организацию сквозного сегмента LSP через одну из AS. Затем контроллер ABNO использует диспетчер обеспечения для программирования соединений между AS, используя ForCES или OpenFlow, а сегменты LSP объединяются в соответствии с идеями из [RFC5150]. Можно спорить, является ли PCE-родитель в этой модели активным (указывает дочерним создать сегменты LSP) или пассивным (запрашивает сегменты пути у потомков).

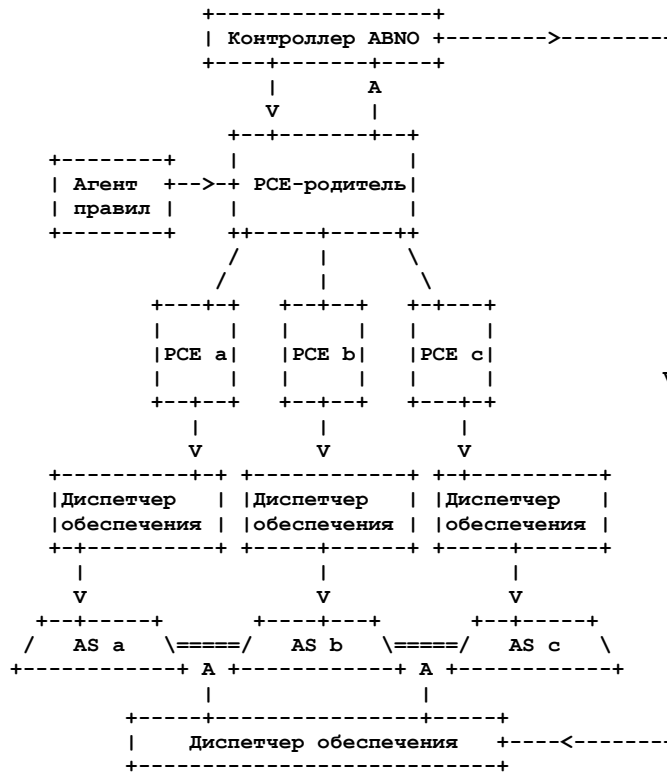


Рисунок 7. Подготовка LSP в активными дочерними PCE и шиванием.

#### 4. Проверка услуги

Контроллеру АВНО необходимо убедиться, что сквозной путь LSP организован в соответствии с запросом. При использовании для создания LSP плоскости управления головной LSR может передать уведомление (возможно, через PCEP) об успешной организации пути, но для проверки создания LSP контроллер АВНО будет запрашивать у обработчика OAM выполнение проверки непрерывности (Continuity Check) OAM в плоскости данных и возврат отчёта о готовности LSP к передаче трафика.

#### 5. Уведомление о предоставлении услуги

Когда контроллер АВНО убедился в готовности службы передавать трафик, он уведомляет об этом NMS/OSS. Предоставление услуги можно дополнительно проверить с помощью аудита сети (параграф 2.3.2.7).

## 3.2. Многоуровневая сеть

Сети обычно создаются с использованием нескольких уровней, представляющих разделение административных областей или технологий, а также по ролям узлов (клиенты и серверы).

Предпочтительно координировать управление сетевыми ресурсами и их использованием (учёт нескольких уровней и управления ими) вместо независимого управления и оптимизации ресурсов на каждом уровне. Это способствует повышению эффективности и автоматизации сети и может определяться как межуровневая организация трафика.

Архитектура PCE поддерживает межуровневую организацию трафика [RFC5623], а в сочетании с архитектурой АВНО обеспечивает набор возможностей для координации сетевых ресурсов между несколькими уровнями.

Приведённый ниже пример показывает использование АВНО для координации распределения сетевых ресурсов серверного уровня с целью создания виртуальной топологии в сети клиентского уровня для организации на нем сквозной связности. Многоуровневая сеть представлена на рисунке 8.

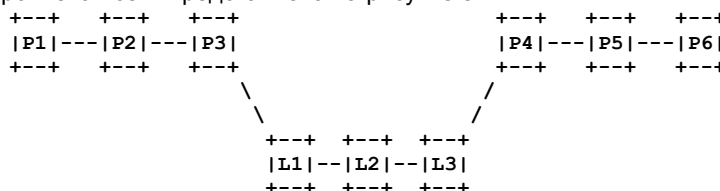


Рисунок 8. Многоуровневая сеть.

В сети имеется 6 маршрутизаторов пакетного уровня (P1 - P6) и три оптических коммутатора по длине волны ( $\lambda$ , L1 - L3). Между маршрутизаторами P1 - P3, а также между P4 - P6 имеется связность, но эти «островки» маршрутизаторов не связаны на уровне пакетов (возможно из-за отказа в сети или нехватки пропускной способности между островками). Однако имеется связность на оптическом уровне между коммутаторами L1 - L3 и оптическая сеть подключена к маршрутизаторам P3 и P4 (с оптическими линейными платами). В этом примере желательнее соединение на пакетном уровне (MPLS LSP) между P1 и P6, для организации которого в архитектуре АВНО используется ряд шагов.

#### 1. Управление запросами.

Как показано на рисунке 9, координатор прикладных услуг подаёт запрос на организацию связности между P1 и P6 в сети пакетного уровня, т. е. координатор запрашивает MPLS LSP с определённой пропускной способностью для доставки трафика приложений. Контроллер АВНО проверяет наличие у координатора прикладных служб прав на запрос услуги.



Рисунок 9. Управление запросами координатора прикладных служб.

- Расчёт пути для услуги на пакетном уровне.

Контроллер ABNO передаёт запрос на расчёт пути элементу PCE пакетного уровня для нахождения подходящего LSP, как показано на рисунке 10. PCE использует подходящие для запроса правила и обращается к базе TED для пакетного уровня. Выясняется, что доступных сейчас путей нет.

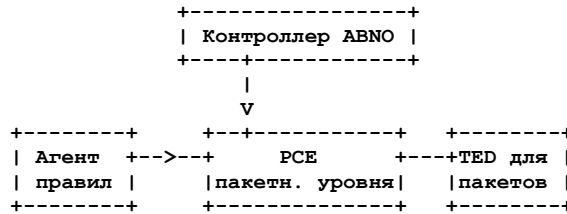


Рисунок 10. Запрос расчёта пути.

- Вызов VNTM и расчёт пути на оптическом уровне.

После неудачного расчёта пути в п. 2 PCE вместо уведомления об этом контроллера ABNO вызывает VNTM для определения возможности создать нужный канал в виртуальной топологии для преодоления разрыва.

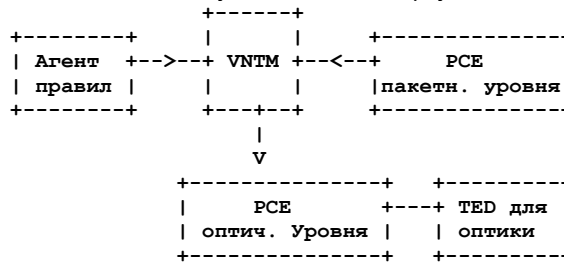


Рисунок 11. Вызов VNTM и расчёта пути оптического уровня.

Как показано на рисунке 11, PCE пакетного уровня сообщает диспетчеру VNTM о проблеме связности и тот обращается к правилам, чтобы проверить, разрешено ли ему находить путь. При положительном ответе VNTM запрашивает у PCE оптического уровня поиск пути через оптическую сеть для организации виртуального канала для пакетного уровня. При обработке этого запроса PCE обращается к базе TED оптического уровня.

- Предоставление оптического пути.

После нахождения пути через оптическую сеть его нужно предоставить. Варианты решения этой задачи описаны в п. 3 параграфа 3.1. Инициировать предоставление может PCE оптического уровня или его пользователь (VNTM). Команду можно отправить головному узлу оптического LSP (P3), чтобы можно было использовать плоскость управления (например, GMPLS RSVP-TE [RFC3473]) для предоставления LSP. Можно также предоставить ресурсы сети напрямую с помощью любого из механизмов, указанных в параграфе 2.3.2.6.

- Создание виртуальной топологии на пакетном уровне.

После создания LSP на оптическом уровне его можно сделать доступным пакетному уровню как виртуальный канал. При использовании сигнального механизма GMPLS [RFC6107] процесс можно автоматизировать в плоскости управления, в иных случаях может потребоваться специальная инструкция для головного маршрутизатора оптического LSP (например, через I2RS).

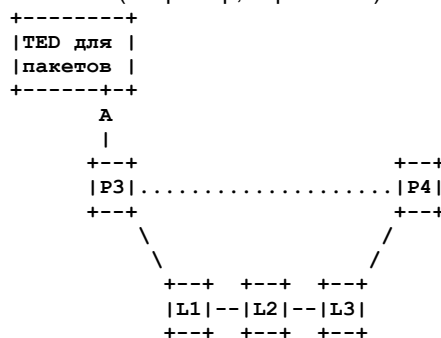


Рисунок 12. Анонсирование нового виртуального канала.

После создания виртуального канала (рисунок 12) он анонсируется в IGP для сети пакетного уровня и появляется в базе TED для этой сети.

- Завершение расчёта пути и его предоставление на пакетном уровне.

Сейчас пакетная сеть уже имеет требуемые ресурсы и PCE для пакетного уровня может завершить свою работу с предоставлением MPLS LSP как описано в параграфе 3.1.

## 7. Проверка услуги и уведомление об исполнении.

Как отмечено в параграфе 3.1, контроллеру ABNO нужно проверить возможность корректной организации сквозного LSP до уведомления о завершении организации услуги контроллера прикладных служб. Кроме того, весьма вероятно, что потребуется проверка услуги до того, как можно будет начать использование LSP оптического уровня, т. е. VNTM должен координироваться с обработчиком OAM для проверки готовности LSP.

## 3.2.1. Соединение ЦОД через многоуровневые сети

Для поддержки новых и создаваемых облачных приложений, таких как резервирование данных в реальном масштабе времени, перенос виртуальных машин, кластеризация серверов или реорганизация ресурсов, все чаще требуется динамическое предоставление или выделение ресурсов IT, а также соединение удалённых ЦОД. Эти операции требуют доставки трафика между ЦОД и соединения, обеспечивающие такую связь, обычно предоставляются через статические устройства (каналы) или выделенные арендованные линии, что ведёт к неэффективному использованию ресурсов. Кроме того, основным требованием является возможность управления удалёнными ЦОД как единым целым.

В таких средах плоскость данных зависит от оператора и провайдера. Их клиенты могут арендовать услуги с коммутацией длин волн (*lambda switch capable* или LSC) или пакетов (*packet switch capable* или PSC), а также мультиплексирования с разделением по времени (*time division multiplexing* или TDM), а применение архитектуры и контроллера ABNO позволяет обеспечить динамическое предоставление сетевых услуг независимо от базового сервиса и транспорта.

Соединение ЦОД может включать эксплуатацию, управление и поддержку гетерогенных сред для сайта каждого ЦОД и сегмента ядра сети (*metro-core*), соединяющий их, не только в части технологии базовой плоскости данных, но и в плоскости управления. Например, каждый сайт ЦОД или домен может централизованно управляться локально (скажем, через OpenFlow [ONF]), а транспорт в инфраструктуре ядра сети может управляться с помощью GMPLS. Хотя протокол OpenFlow специально приспособлен для работы в однодоменных сетях ЦОД (управление на уровне пакетов, множество маршрутных исключений), стандартизованная архитектура на базе GMPLS позволит динамически выделять и восстанавливать оптические ресурсы в многодоменных (например с оборудованием разных производителей) сетях, соединяющих ЦОД. Аспекты применения архитектуры ABNO и связанных с этим процедур описаны ниже.

## 1. Запрос от координатора прикладных служб или NMS.

Как показано на рисунке 13, контроллер ABNO получает от координатора прикладных служб или NMS запрос на создание нового сквозного соединения между парой конечных точек. Фактические адреса этих конечных точек рассматриваются ниже. Контроллер ABNO запрашивает у PCE путь между этими точками после рассмотрения применимой политики, заданной агентом правил (см. рисунок 1).

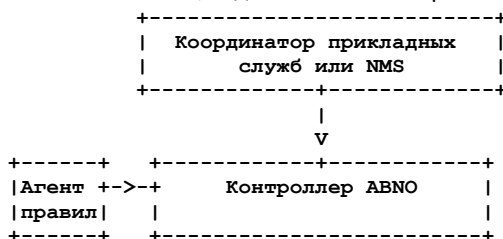


Рисунок 13. Управление запросами координатора прикладных служб.

## 2. Отображение адресов.

Для расчёта сквозного пути элементу PCE нужно иметь однородное представление всей топологии, что означает необходимость учёта и идентификации фактических конечных точек в части сетевых адресов клиентов. Контроллеру ABNO и/или PCE может потребоваться трансляция или отображение адресов из разных пространств. В зависимости от распространения и сбора топологических сведений возможны два варианта, описанных ниже.

## 2a Уровень приложений знает сетевой уровень клиента

Объекты прикладного уровня могут иметь интерфейс с TED или сервером ALTO, позволяющий им сопоставить высокоуровневые конечные точки с сетевыми адресами. Механизм такого сопоставления выходит за рамки этого документа, но основан на прямых интерфейсах с другими компонентами ABNO в дополнение к интерфейсу с контроллером ABNO.

В этом варианте запрос от NMS или координатора прикладных служб содержит адреса в сети клиентского уровня, поэтому при запросе контроллера ABNO к PCE для расчёта пути между двумя конечными точками PCE может использовать предоставленные адреса, рассчитать путь и продолжать рабочий процесс, взаимодействуя с диспетчером обеспечения.

## 2b Уровень приложений не знает сетевой уровень клиента

В этом случае контроллер ABNO получает от NMS или диспетчера прикладных служб запрос, содержащий лишь идентификаторы из пространства адресов уровня приложений. Чтобы PCE мог рассчитать сквозной путь, эти адреса нужно преобразовать в адреса сети клиентского уровня. Трансляцию выполняет контроллер ABNO, который может обращаться к базам данных TED и ALTO, что позволяет запросу, передаваемому элементу PCE, оставаться связанным с одной сетью и TED. Как вариант, запрос на расчёт может использовать адреса уровня приложений, предоставляя трансляцию адресов элементу PCE.

Отметим, что во избежание путаницы оба подхода в этом случае требуют чёткого указания используемого пространства адресов.

## 3. Процесс обеспечения.

После нахождения пути диспетчер обеспечения получает высокоуровневый запрос на предоставление услуги. Поскольку в рассматриваемом случае элементы сети не обязательно настраиваются по одному протоколу,

сквозной пути делится на сегменты, а контроллер ABNO координирует и организует их создание, адаптируя и/или транслируя абстрактный запрос предоставления услуги в запросы конкретных сегментов с помощью VNTM или PCE, выдающих соответствующие команды или инструкции. Обеспечение может включать настройку элементов сети напрямую или передачу организации базового соединения выделенному экземпляру плоскости управления, отвечающему за этот сегмент.

Диспетчер обеспечения может использовать для программирования элементов сети множество механизмов, как показано на рисунке 14. Он определяет технологию, применяемую для фактического обеспечения в каждом сегменте, из заданной вручную конфигурации или путём обнаружения.

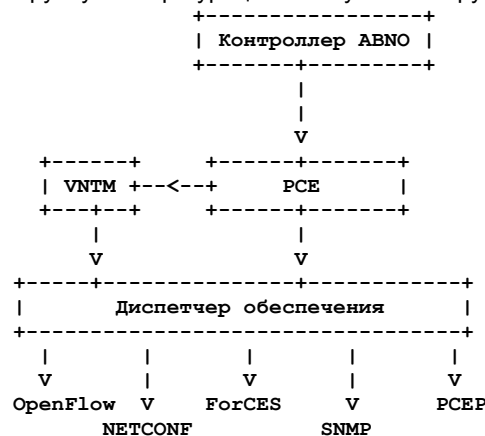


Рисунок 14. Процесс обеспечения.

#### 4. Проверка и уведомление об исполнении.

После предоставления услуги сквозной связности и проверки корректности её работы контроллеру ABNO нужно уведомить координатор прикладных служб или NMS.

### 3.3. Работа без прерывания

Многие службы зависят от организации нового LSP, чтобы можно было переключить трафик с имеющегося LSP с минимальными перебоями или без таковых. В этом параграфе рассматриваются эти варианты использования, представлена общая модель работы без прерывания (make-before-break) в архитектуре ABNO и показано как можно поддержать каждый из вариантов применения с помощью элементов базовой модели.

#### 3.3.1. Работа без прерывания при повторной оптимизации

Работа без прерывания - это механизм сигнализации RSVP-TE, организующий новый LSP до разрыва имеющегося LSP [RFC3209]. Это обеспечивает некоторые преимущества в таких ситуациях, как реоптимизация работающих LSP.

Процесс прост и в примере на рисунке 15 используется PCE с учётом состояния [Stateful-PCE] для мониторинга сети и принятия при необходимости мер повторной оптимизации. В этом процессе запрос на обслуживание поступает к контроллеру ABNO, например, от OSS. Запрос обслуживания указывает, что LSP нужно оптимизировать снова с учётом конкретных условий и правил. Это позволяет контроллеру ABNO управлять порядком и приоритетами реоптимизации множества LSP с использованием элементов глобальной одновременной оптимизации (Global Concurrent Optimization или GCO), как описано в параграфе 3.4, и применением правил во всей сети, например, для первоочередной реоптимизации LSP, обеспечивающих чувствительные к времени службы.

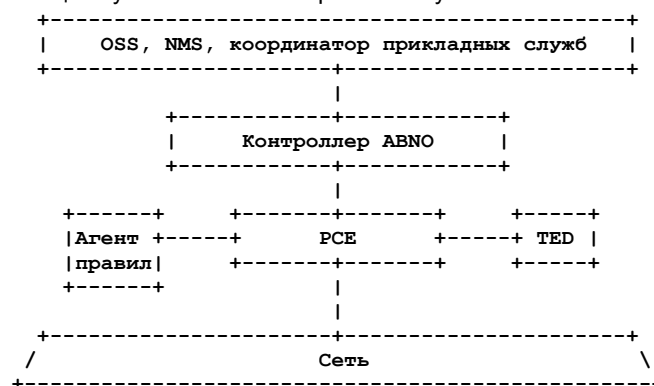


Рисунок 15. Процесс работы без прерывания.

Контроллер ABNO поручает элементу PCE расчёт и организацию начального пути. PCE отслеживает изменения в сети с течением времени, отражаемые в базе TED, и в соответствии с правилами может рассчитать и организовать путь на замену, используя в сети работу без прерывания. После организации нового пути о сообщении сети о его корректном использовании PCE уничтожает прежний путь и может сообщить о реоптимизации контроллеру ABNO.

#### 3.3.2. Работа без прерывания при восстановлении

Работа без прерывания может применяться для ремонта отказавшего LSP, когда есть желание сохранить ресурсы на части пути и имеется вероятность «кражи» ресурсов другими LSP, если сначала уничтожить отказавший LSP. В отличие от примера из параграфа 3.3.1, в данном случае рассматривается ситуация, когда обслуживание прервано в результате отказа в сети. Очевидно, что в случае LSP «один со многими» отказ может затрагивать лишь часть дерева и прерывание будет лишь для части листьев-адресатов, поэтому восстановление без прерывания не будет влиять на листья, не затронутые первоначальным отказом.

На рисунке 16 показаны компоненты, взаимодействующие в этом случае. Запрос на обслуживание приходит контроллеру ABNO, например, от OSS. Запрос указывает, что LSP может быть восстановлен после отказа и следует попытаться использовать как можно большую часть исходного пути.

Контроллер ABNO поручает PCE рассчитать и организовать исходный путь, а также просит обработчик OAM инициировать OAM на LSP и отследить результаты. В какой-то момент сеть сообщает об отказе обработчику OAM, который уведомляет контроллер ABNO. Контроллер ABNO поручает PCE рассчитать новый путь, используя как можно больше ресурсов исходного пути, и PCE создаёт новый LSP.

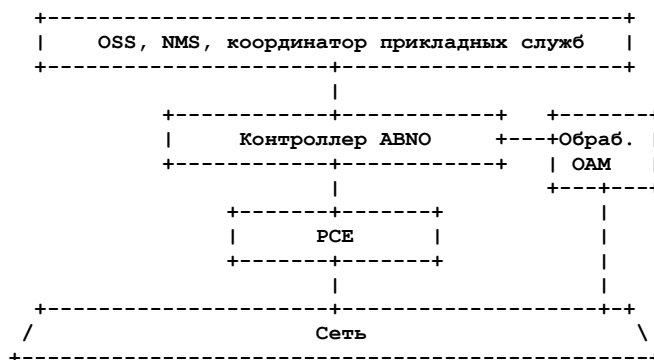


Рисунок 16. Процесс восстановления без прерывания работы.

После организации нового пути и сообщения из сети о его корректной работе контроллер ABNO даёт элементу PCE команду уничтожить старый путь.

### 3.3.3. Работа без прерывания при проверке и выборе пути

В более сложном варианте использования отслеживание LSP может выполняться по множеству атрибутов, таких как задержка и её вариации. Когда LSP выходит за установленные пороги, трафик может переноситься на другой LSP, обеспечивающий желаемое (или лучшее) качество обслуживания. Для этого нужно организовать новый LSP и протестировать его, а необходимость переноса трафика требует использовать работу без прерывания.

Возможно, что новый LSP не сможет обеспечить желаемые характеристики и потребуются проверить несколько LSP для выбора наилучшего. Даже при организации исходного LSP может быть желательным тестирование нескольких LSP для выбора используемого трафиком.

На рисунке 17 показаны компоненты, взаимодействующие в этом случае. Поскольку одновременно может существовать множество LSP, требуется отдельная операция для координации выбора пути и эту работу выполняет клиент I2RS под управлением контроллера ABNO.

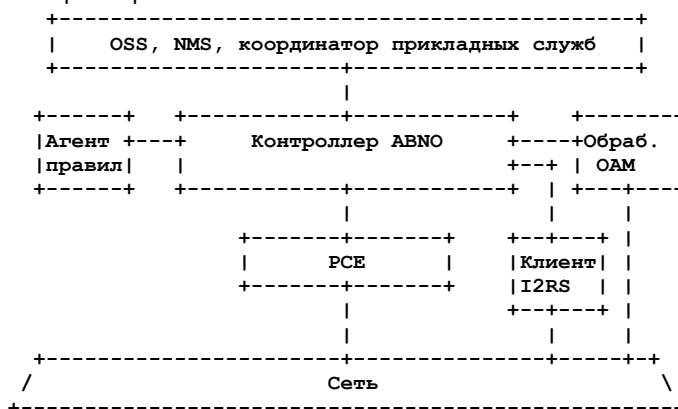


Рисунок 17. Процесс проверки и выбора пути без прерывания.

Обработчик OAM отвечает за инициирование тестов LSP и возврат результатов контроллеру ABNO. Обработчик OAM может также проверять результаты тестов сквозной связности через многодоменную сеть, где в каждом домене применяется своя технология. Например, сквозной путь может включать сегменты MPLS, Ethernet/VLAN, IP и т. п.

В остальном процесс похож на повторную оптимизацию, описанную в параграфе 3.3.1. Приведённый ниже псевдокод иллюстрирует взаимодействия между компонентами ABNO.

OSS запрашивает услугу с гарантией качества

```
:Label1
```

DoWhile пока недостаточно LSP (контроллер ABNO)

  Запросить у PCE расчёт и предоставление LSP (контроллер ABNO)

  Создать LSP (PCE)

```
EndDo
```

```
:Label2
```

DoFor для каждого LSP (контроллер ABNO)

  Проверить LSP (обработчик OAM)

  Сообщить результаты контроллеру ABNO (обработчик OAM)

```
EndDo
```

Оценить результаты всех тестов (контроллер ABNO)



```

Выбрать предпочтительный LSP и указать клиенту I2RS (контроллер ABNO)
Поместить трафик на предпочтительный LSP (I2RS Client)

DoWhile слишком много LSP (контроллер ABNO)
  Дать PCE команду уничтожения ненужного LSP (контроллер ABNO)
  Уничтожить ненужный LSP (PCE)
EndDo

DoUntil пока нет триггера (обработчик OAM, контроллер ABNO, Policy Agent)
  Сохранять передачу трафика (Network)
  Протестировать LSP (обработчик OAM)
EndDo

If уже есть подходящий LSP (контроллер ABNO)
  GoTo Label2
Else
  GoTo Label1
EndIf

```

### 3.4. Глобальная одновременная оптимизация

Глобальная одновременная оптимизация (GCO) определена в [RFC5557] и является ключевой технологией для максимизации эффективности сети путём одновременного расчёта набора путей с организацией трафика. Запрос на расчёт пути GCO учитывает всю топологию сети и полный набор LSP вместе с соответствующими ограничениями. GCO подходит также для перерасчёта путей набора имеющихся LSP.

Оптимизация GCO может запрашиваться в разных ситуациях, включая:

- маршрутизацию новых услуг, когда элементу PCE нужно учитывать другие службы или топологию сети;
- повторную оптимизацию имеющихся служб из-за фрагментации ресурсов сети или неоптимального размещения последовательно рассчитываемых услуг;
- восстановление связности для массовых услуг при катастрофических сбоях в сети.

Сервис-провайдер может также захотеть рассчитать и внедрить новые массовые службы на основе прогнозируемой матрицы трафика. Функциональность GCO и возможности одновременных расчётов обеспечивают значительные преимущества при оптимизации сети, позволяя оптимально использовать ресурсы сети и избегать блокировки.

В приведённом ниже примере показано как архитектура и компоненты ABNO применяются для одновременной оптимизации набора услуг.

#### 3.4.1. Пример использования GCO с MPLS LSP

При рассмотрении GCO для расчёта путей можно разделить целевые функции GCO на три категории:

- минимизация совокупного расхода пропускной способности (Minimize Bandwidth Consumption или MBC);
- минимизация загрузки наиболее занятого канала (Most Loaded Link или MLL);
- минимизация совокупных расходов для набора путей (Minimize Cumulative Cost или (MCC).

В приведённом примере использования предполагается, что GCO запрашивается по отдельному каналу (offline) системой NMS/OSS. Расчёт путей может занять значительное время, а пользователь может проверять рассчитанные пути перед их обеспечением в сети.

##### 1. Управление запросами

NMS/OSS запрашивает новую услугу связности для массовых служб. Контроллер ABNO проверяет наличие у NMS/OSS права подавать такой запрос и применяет атрибут GCO с запросом MBC (рисунок 18).

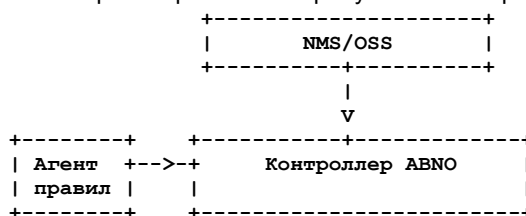


Рисунок 18. Запрос NMS к контроллеру ABNO.

- 2а В каждом запросе указывается источник, получатель и запрашиваемая пропускная способность. Запросы передаются контроллеру ABNO и классифицируются как запросы GCO. PCE применяет для каждого запроса подходящие правила и обращается к TED пакетного уровня.
2. Расчёт пути на уровне пакетов.

При расчёте набора услуг для приложения GCO протокол PCEP поддерживает списки векторов синхронизации (synchronization vector или SVEC) для синхронизированного расчёта путей, как задано в [RFC5440] и описано в [RFC6007].

2а Контроллер ABNO передаёт запрос массового обслуживания элементу PCE пакетного уровня с поддержкой GCO в сообщении PCEP. PCE применяет к запросу подходящие правила и обращается к TED пакетного уровня, как показано на рисунке 19.

2b При получении запроса массового обслуживания (GCO) элемент PCE применяет целевую функцию MBC и рассчитывает услуги одновременно.



Рисунок 19. Запрос расчёта пути для PCE с поддержкой GCO.

2с По завершении расчёта путей запрошенной услуги GCO элемент PCE передаёт эти пути контроллеру ABNO. Отклик включает полный явный путь для каждой услуги (TE LSP).

3. Одновременно рассчитанное решение, полученное от PCE контроллер ABNO отправляет NMS/OSS как отклик PCEP (рисунок 20). Пользователь NMS/OSS может проверить пути-кандидаты и предоставить их для новых услуг или сохранить решение на будущее.

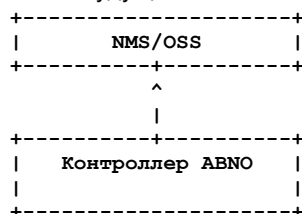


Рисунок 20. Передача решения от ABNO в NMS/OSS.

### 3.5. Адаптивное управление сетью (ANM)

Архитектура ABNO обеспечивает возможность реактивного управления сетевыми ресурсами на основе классификации, профилирования и предсказаний в соответствии с текущими потребностями и загрузкой ресурсов. Можно манипулировать ресурсами транспортной сети серверного уровня, такими как временная нарезка оптической транспортной сети (Optical Transport Network или OTN) [G.709] или сетка длин волн с переменной спектральной полосой (flexi-grid) [G.694.1], в соответствии с текущими и прогнозируемыми потребностями в рамках модели эластичных оптических сетей (Elastic Optical Networks или EON) [EON]. EON обеспечивает эффективный и расширяемый транспорт за счёт гибкой гранулярной сортировки трафика в домене оптических частот. Это достигается за счёт произвольного объединения (конкатенации) оптического спектра, позволяющего обеспечивать настраиваемую пропускную способность. Полоса выделяется с дискретностью 12,5 ГГц.

Адаптивное управление сетью (Adaptive Network Management или ANM) с EON позволяет выделять оптическую полосу нужного размера для сквозного оптического пути. При использовании flexi-grid выделение полосы происходит в соответствии с объёмом трафика, форматом оптической модуляции, дальностью связи или запросами пользователей, что позволяет добиться высокой спектральной эффективности и расширяемости. OTN обеспечивает гибкое и гранулярное выделение пропускной способности в сетях с коммутацией по длинам волн (Wavelength Switched Optical Network или WSON).

Для эффективного использования оптических ресурсов нужна система мониторинга сетевых ресурсов и выбора оптимальной конфигурации сети на основе состояния, доступности пропускной способности и пользовательских служб. Такую систему называют ANM.

#### 3.5.1. Включение ANM

Процесс адаптивного управления сетью может включаться по разным причинам, указанным ниже.

##### Измерения

Измерения трафика могут применяться для распределения полосы (пропускной способности) в соответствии с потребностями трафика, насколько это возможно. Триггером могут служить измерения потоков трафика на маршрутизаторах IP, проверка баз данных организации трафика и состояний каналов, пороговые значения загрузки критически важных каналов сети, запросы от внешних объектов. В настоящее время операторы используют в сетях активные зонды для мониторинга, которые сохраняют результаты в системе OSS. Компоненты OSS или обработчика OAM активируют основанный на измерениях триггер, поэтому контроллер ABNO в данном случае не принимает непосредственного участия в процессе.

##### Запуск человеком

Оператор может запросить у ABNO запуск процесса адаптивного управления сетью через NMS.

##### Периодический запуск

Процесс адаптивного планирования сети может запускаться периодически для поиска оптимальной конфигурации. Контроллер ABNO получает от OSS или NMS запрос на запуск процесса адаптивного управления сетью.

#### 3.5.2. Обработка запроса на расчёт GCO

По запросам человека или периодических триггеров, описанных в предыдущем параграфе, OSS или NMS передаёт контроллеру ABNO запрос на выполнение GCO на основе EON. Контроллер ABNO выбирает набор служб для повторной оптимизации и целевую функцию, обеспечивающую наилучшее использование ресурсов сети. При выборе контроллер руководствуется политикой использования ресурсов для всей сети, определением оптимизации и допустимой степенью нарушения работы имеющихся услуг.

Этот запрос GCO передаётся элементу PCE, как описано в параграфе 3.4. PCE может рассмотреть сквозные пути и оптимальное распределение спектра для каждого канала, чтобы удовлетворить потребности трафика и оптимизировать расход оптической полосы пропускания в сети.

PCE работает с базой TED, но, вероятно, будет учитывать состояния, чтобы знать, какие LSP соответствуют тому или иному выделению полосы и на каких каналах сети. Получив ответ, PCE возвращает набор возможных путей контроллеру ABNO, который передаёт их NMS или OSS для выбора/контроля последующего процесса установки или изменения пути. Этот обмен показан на рисунке 21, где не показаны взаимодействия, используемые OSS/NMS для организации или изменения LSP в сети.

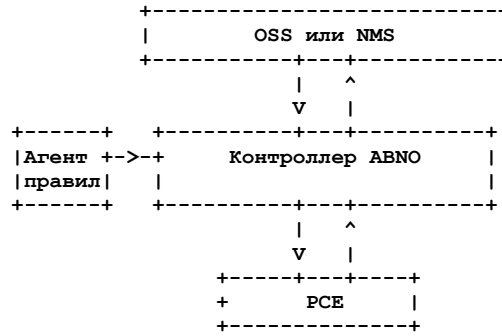


Рисунок 21. Адаптивное управление сетью с участием человека.

### 3.5.3. Автоматизированный процесс обеспечения

Хотя оператор контролирует большинство сетевых операций, для некоторых действий не требуется присмотра, например, для простой смены модуляции в транспондере с переменной битовой скоростью (Bit-rate Variable Transponder или BVT) с целью повышения оптической спектральной эффективности или снижения энергопотребления. Когда участие человека не требуется, PCE передаёт диспетчеру обеспечения новую конфигурацию для настройки элементов сети, как показано на рисунке 22.

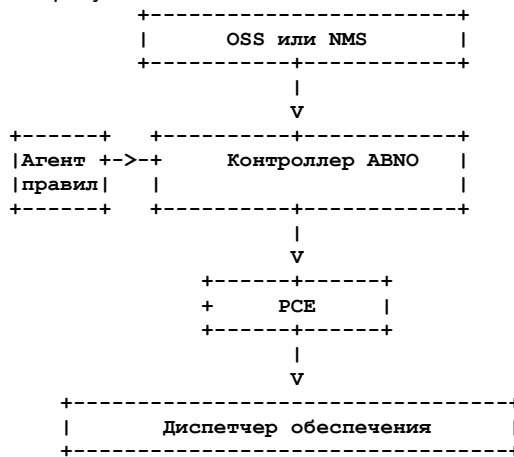


Рисунок 22. Адаптивное управление сетью без участия человека.

## 3.6. Операции псевдопроводов и управление ими

Псевдопровода в сети MPLS [RFC3985] работают как форма многоуровневой сети на основе связности, предоставляемой сетью MPLS. Псевдопровода организуются через LSP, работающие как транспортные туннели, и для размещения туннелей в сети и привязки их к псевдопроводам требуется планирование.

В этом параграфе рассмотрены 4 примера: многосегментные псевдопровода, псевдопровода с разными путями (в том числе многосегментные, и защита сегментов в псевдопроводе. В параграфе 3.6.5 рассматривается применимость архитектуры ABNO для этих случаев.

### 3.6.1. Многосегментные псевдопровода

В [RFC5254] описана архитектура многосегментных псевдопроводов. Сквозная услуга, как показано на рисунке 23, может состоять из последовательности «сшитых» сегментов, обозначенных на рисунке как AC, PW1, PW2, PW3, AC. Сегменты псевдопровода соединяются на границе «поставщика сшивки» (stitching Provider Edge или S-PE), например, PW1 сшивается с PW2 в S-PE1. Каждое устройство доступа (AC) сшивается с сегментом псевдопровода на завершающем PE (terminating PE или T-PE), например, PW1 сшивается с AC в T-PE1.

Каждый сегмент псевдопровода передаётся через сеть MPLS по пути LSP, работающему как транспортный туннель, например, PW1 передаётся через LSP1. LSP между узлами PE могут проходить через разные сети MPLS с PE в качестве граничных узлов или PE могут размещаться внутри сети так, что каждый LSP включает охватывает лишь часть сети.

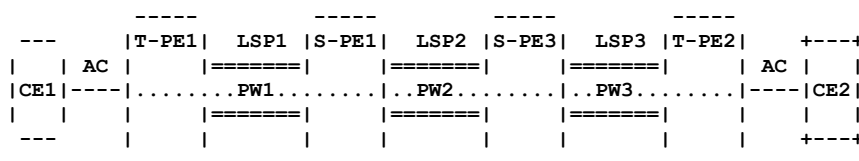


Рисунок 23. Многосегментный псевдопровод.

Хотя показанная на рисунке 23 топология проста для понимания в реальных сетях все может быть гораздо сложнее. На рисунке 24 показана небольшая многосвязная (mesh) сеть PE. Каналы между PE не являются физическими и представляют возможные MPLS LSP между PE.

При организации сквозной услуги между граничными узлами клиентов (Customer Edge или CE) CE1 и CE2 нужно выбрать используемые PE. Иными словами, нужно рассчитать путь для определения сегментов псевдопровода (hop), а затем организовать требуемые туннели LSP для передачи сегментов псевдопровода, которые будут сшиты воедино. Для каждого LSP может потребоваться свой расчёт пути через сеть MPLS между PE. Выбор путей для многосегментного псевдопровода будет зависеть от:

- связности MPLS;
- доступности пропускной способности MPLS;
- возможностей шивки псевдопроводов и пропускной способности PE;
- правил и соображений конфиденциальности при использовании PE.

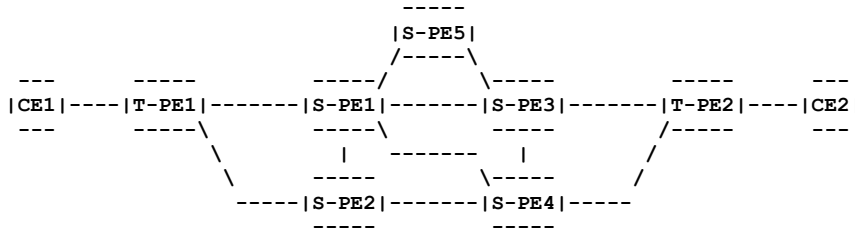


Рисунок 24. Топология сети с многосегментными псевдопроводами.

### 3.6.2. Псевдопровода с разными путями

Для предоставляемой псевдопроводом услуги связности может потребоваться устойчивость к отказам. Во многих случаях это обеспечивается путём организации пары псевдопроводов, передаваемых через сеть по разным LSP, как показано на рисунке 25 (обозначения заимствованы из [RFC3985]). Очевидно, что в этом случае задача состоит в постоянном разделении двух LSP (LSP1 и LSP2) внутри сети MPLS. Это не отличается от обычной проблемы разных путей в сети MPLS.

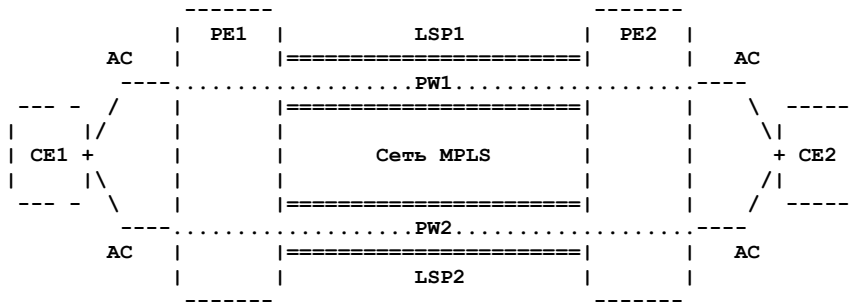


Рисунок 25. Псевдопровода с разными путями.

Псевдопровод с разными путями на рисунке 26 организован за счёт «двудомности» каждого CE через несколько PE. Требование к разным путям LSP сохраняется, но оно усложняется разными конечными точками LSP. В этом случае головной маршрутизатор (например, PE1) не может полагаться для поддержки разных путей на сигнальный протокол, поскольку ему известен путь лишь одного LSP. Поэтому нужна координация при расчёте путей.

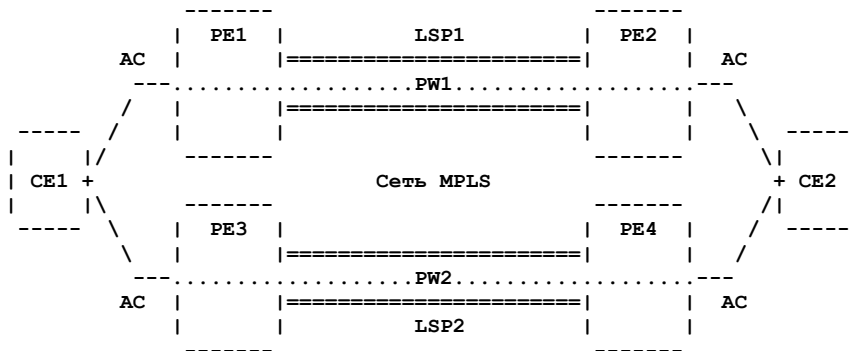


Рисунок 26. Псевдопровода с разными путями и развязанными PE.

### 3.6.3. Многосегментные псевдопровода с разными путями

На рисунке 27 показано как можно объединить услуги из двух предыдущих параграфов для предоставления сквозной услуги с разными путями в многосегментной среде. Для обеспечения вквозной отказоустойчивости нужна два полностью разные многосегментные провода.

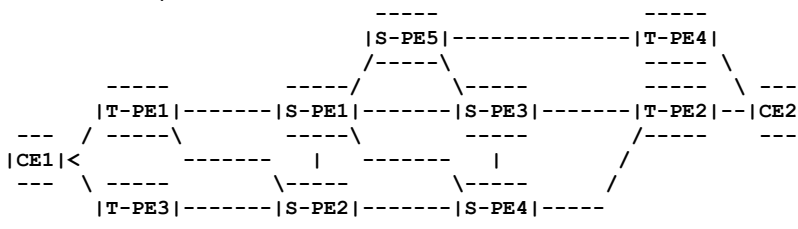


Рисунок 27. Топология с многосегментными проводами по разным путям.

Как при любом расчёте разных путей, первый путь выбирается с учётом необходимости второго, полностью иного пути. Если применить последовательный расчёт к топологии рисунка 27, первый путь CE1,T-PE1,S-PE1,S-PE3,T-PE2,CE2

сделает невозможным расчёт полностью отличающегося второго пути. Проблема усложняется многоуровневой природой пути. Выбора разных PE недостаточно, поскольку туннели LSP между ними могут совместно использовать каналы сети MPLS. По этому для обеспечения желаемого уровня сервиса требуется многоуровневое планирование.

### 3.6.4. Защита сегментов псевдопровода

Альтернативой услуге сквозной защиты псевдопровода, обеспечиваемой механизмом, описанным в параграфе 3.6.3, может быть защита отдельных сегментов псевдопровода или PE. Например, псевдопровод между S-PE1 и S-PE5 на рисунке 27 можно защитить парой шитых сегментов S-PE1 - S-PE5 и S-PE5 - S-PE3, как показано на рисунке 28.

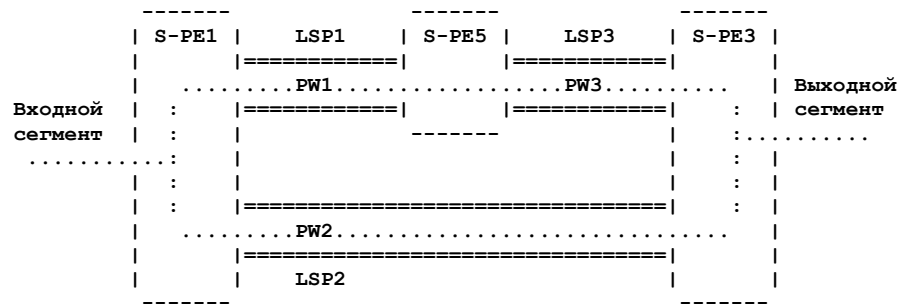


Рисунок 28. Фрагмент многосегментного псевдопровода с защитой сегмента.

Определение сегментов для защиты псевдопроводов требует координации и планирования и, как указано в параграфе 3.6.5, это планирование должно учитывать пути, обеспечиваемые LSP через базовые сети MPLS.

### 3.6.5. Применимость ABNO для псевдопроводов

Архитектура ABNO хорошо подходит для планирования и управления псевдопроводами в описанных выше вариантах применения. Пользователю или приложению нужна единая точка для запроса услуг - контроллер ABNO. Контроллер может запросить у PCE топологию поддерживающих шивку псевдопроводов PE, а также дополнительные сведения о возможностях PE, такие как нагрузка на них и административные правила, способность PCE использовать несколько TED или других PCE в базовых сетях MPLS для определения путей туннелей LSP. На момент создания этого документа протокол PCER не поддерживал запросы и отклики расчёта путей, связанные с псевдопроводами, но общие концепции этого очень похожи на уже используемые и значительного расширения не потребуется.

После расчёта путей можно использовать разные системы обеспечения для организации LSP и предоставления псевдопроводов под контролем диспетчера обеспечения. Контроллер ABNO с помощью клиента I2RS будет инструктировать сетевые устройства в части размещения трафика в псевдопроводах и в сочетании с обработчиком OAM будет обеспечивать корректную обработку отказов, надлежащее качество обслуживания и уровень защиты.

Во многих случаях сеть псевдопроводов образует наложенную сеть (со своей базой TED и механизмами обеспечения) на основе базовых пакетных сетей, предоставляя возможность передачи для сетей клиентов (данные псевдопроводов). Таким образом, пространство задачи, решаемой ABNO в этом случае представляет классическую многоуровневую сеть.

## 3.7. Межслойная оптимизация (CSO)

Если рассматривать термин «слой» (stratum) как широкое разделение уровней, наиболее важных для приложения и сети в целом, необходимость межслойной оптимизации (Cross-Stratum Optimization или CSO) возникает, когда нужно координировать страты приложений и сети для эффективной работы и оптимизации ресурсов в обоих слоях.

Приложения в ЦОД могут предоставлять широкий набор услуг, таких как видеоигры, облачные вычисления и grid-приложения. Появляются также широкополосные видео-приложения, такие как удалённая хирургия, живые концерты и спортивные события.

Этот вариант применения архитектуры ABNO связан в основном с приложениями для ЦОД, предъявляющими высокие требования к пропускной способности в агрегате и индивидуально. Кроме того, этим приложениям могут потребоваться особые ограничения на параметры, связанные с QoS, такие как задержки и их вариации.

### 3.7.1. Работа ЦОД

Центры обработки данных (ЦОД) имеют разные размеры и конфигурации, но все они содержат вычислительные серверы, хранилища и средства управления приложениями. ЦОД предоставляют конечным пользователям прикладные услуги, такие как видеоигры, облачные вычисления и т. п. Поскольку ЦОД, используемые для предоставления прикладных услуг, могут быть распределены в сети, решения о контроле и управлении прикладными услугами, например, при создании нового экземпляра услуги или выборе ЦОД для клиента, могут существенно влиять на состояние сети, а возможности и состояние сети могут существенно влиять на производительность приложений. Эти решения обычно принимаются приложениями, у которых очень мало или нет совсем сведений о базовой сети. В результате решения могут быть неоптимальными с точки зрения приложений или в части использования ресурсов сети и качества обслуживания.

Процесс CSO оптимизирует работу приложений и загрузку (использование) сети за счёт координации решений в слоях приложений и сети. Ресурсы приложений можно условно разделить на вычислительные ресурсы (серверы разных типов, виртуальные машины, память и хранилища) и содержимое (видео, аудио, базы данных, большие наборы данных). Под сетевым слоем понимается уровень IP и нижележащие уровни (MPLS, SDH, OTN, WDM). Ресурсы сетевого слоя включают маршрутизаторы, коммутаторы и каналы. Очень важно раскрыть потенциал плоскостей управления MPLS и GMPLS на нижних сетевых уровнях для удовлетворения значительных агрегатных и индивидуальных потребностей прикладного уровня.

Этот пример показывает, что архитектура ABNO поддерживает межслойную (приложения-сеть) оптимизацию для ЦОД. Другие формы CSO (например, для одноранговых приложений) здесь не рассматриваются.

**3.7.1.1. Перенос виртуальных машин**

Ключевым фактором снижения затрат на ЦОД, консолидацию, гибкость и масштабируемость приложений стала технология виртуализации вычислений за счёт применения виртуальных машин (Virtual Machine или VM). Для программного приложения VM выглядит как выделенный процессор со своей памятью и операционной системой. VM не только предоставляют блок вычислительных возможностей, но и обеспечивают среду приложений, которую можно реплицировать, резервировать и перенести. Могут предлагаться разные конфигурации VM, оптимизированные для различных типов обработки (например, с большим объёмом памяти или высокой пропускной способностью). VM можно перемещать между вычислительными ресурсами в одном или разных ЦОД. Перенос VM служит для распределения нагрузки между ресурсами ЦОД и может выполняться в нескольких режимах.

- (i) Плановый или динамический перенос.
- (ii) Массовый или последовательный перенос.
- (iii) Перенос между парой точек (point-to-point) или между точкой и несколькими точками (point-to-multipoint).

Перенос VM может решать задачи распределения нагрузки или планового обслуживания в ЦОД, а также эффективно снижать нагрузку на сеть вокруг ЦОД. Однако процесс переноса VM, особенно между ЦОД, может оказывать существенное влияние на сеть и предоставляемые услуги.

Для некоторых ситуаций, таких как восстановление при аварии, может потребоваться массовый перенос «на лету», который может потребовать одновременного расчёта и динамической организации путей.

Операции прикладного слоя должны учитывать ситуацию в сетевом слое, даже когда действия в сетевом слое определяются состоянием сетевого слоя.

**3.7.1.2. Распределение нагрузки**

Серверы приложений могут размещаться во многих ЦОД в разных частях сети. При запросе конечного пользователя к приложению нужно принять решение о выборе ЦОД, где следует разместить обработку и хранение данных для выполнения запроса. Одним из основных факторов работы приложений в нескольких ЦОД (вместо одного крупного центра) является желание организовать работу приложения на машине, расположенной как можно ближе к конечному пользователю, что обеспечивает повышение качества обслуживания за счёт сокращения задержек в сети. Однако при перегрузке в сти или ЦОД это может давать обратный эффект.

Таким образом, ключевыми факторами выбора сервера при организации VM для приложения являются:

- загрузка серверов в ЦОД;
- загрузка сети в ЦОД;
- загрузка сетей между ЦОД;
- условия в сети между конечным пользователем и ЦОД.

При выборе в слое приложений требуется учитывать ситуацию в сетевом слое.

**3.7.2. Применение архитектуры ABNO**

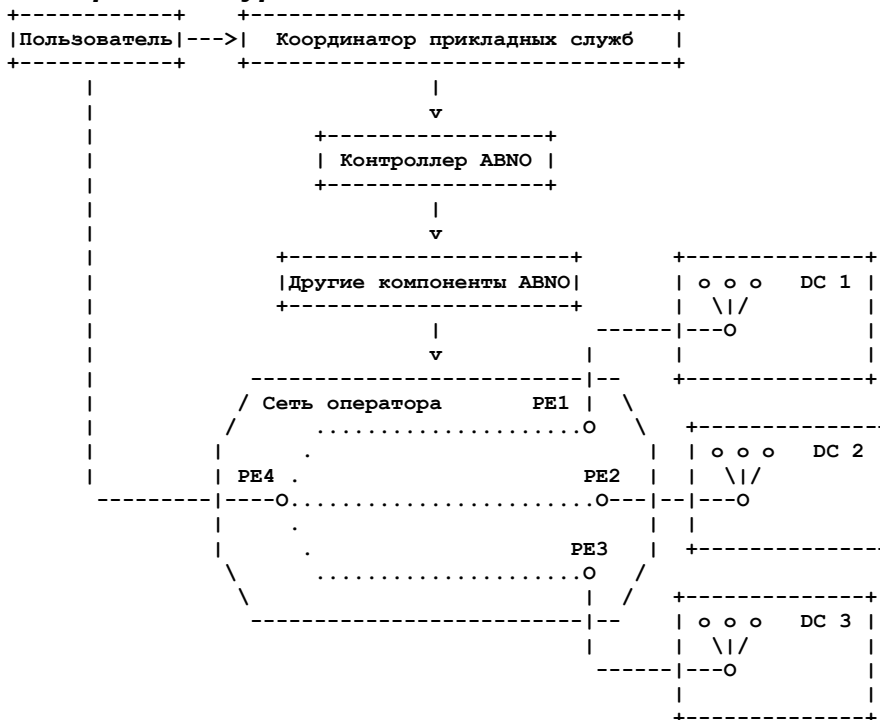


Рисунок 29. Архитектура ABNO в контексте кросс-стратовой оптимизации для ЦОД.

В этом параграфе показано, как можно применить архитектуру ABNO при решении задач межслойного взаимодействия в ЦОД, отмеченных в параграфе 3.7.1.

На рисунке 29 схематически показан пример приложения на основе ЦОД. Сеть оператора обеспечивает доступ конечных пользователей через PE4. Доступ в трёх ЦОД (DC1, DC2, DC3) осуществляется по разным путям через PE1, PE2 и PE3.

Координатор прикладных служб получает от конечного пользователя сведения о желаемых услугах и преобразует их в запрос сервиса, который он передаёт контроллеру ABNO. Конечные пользователи могут уже знать, с каким ЦОД они хотят работать, или координатор прикладных служб может определить такой ЦОД. В ином случае задачу выбора ЦОД должен рашать контроллер ABNO и он может для этого использовать дополнительную базу данных (см. 2.3.1.8) со сведениями о загрузке серверов и другими параметрами ЦОД.

Контроллер ABNO проверяет ресурсы сети, используя сведения от других компонентов ABNO и применяя эти компоненты при настройке сети для удовлетворения потребностей конечных пользователей.

### 3.7.2.1. Внедрённые приложения, службы и продукция

Контроллер ABNO должен использовать ряд компонентов для реализации функций CSO, описанных в параграфе 3.7.1.

Сервер ALTO предоставляет сведения о топологической близости и подходящем географическом местоположении серверов по отношению к базовым сетям. Эти сведения могут применяться для оптимизации выбора партнёра, помогающей сократить путь трафика IP, или указания сети конкретного сервис-провайдера. ALTO в сочетании с контроллером ABNO и координатором прикладных служб могут решать общие проблемы, такие как выбор серверов приложений на основе доступности ресурсов и использования базовых сетей.

Контроллер ABNO может также формировать представление о текущей нагрузке в сети по сведениям из TED или от обработчика OAM (например, путём запуска диагностических инструментов для измерения потерь, задержки и её вариаций). Это представление влияет, очевидно, на предоставление путей от конечных пользователей к ЦОД, а также на выбор ЦОД для предоставления услуг и, возможно, точек присоединения конечных пользователей к выбранному ЦОД. Представление об использовании PCE в CSO приведено в [CSO-PCE] (см. рисунок 29).

Как уже отмечено, комбинация контроллера ABNO и координатора прикладных служб должна быть способна выбирать (возможно, и переносить) местоположение VM для предоставления услуг конечному пользователю. Поскольку для направления конечного пользователя к нужному серверу (VM) применяется перенаправление DNS, важна способность контроллера ABNO помогать должным образом программировать серверы DNS.

Кроме того, как уже отмечалось выше, контроллер ABNO может координировать размещение трафика в сети для распределений нагрузки и обеспечения устойчивости к отказам. Эти свойства могут применяться в сочетании с рассмотренными выше функциями для обеспечения того, что размещение новых VM, создаваемый ими трафик и нагрузка при переносе не будут нарушать работу имеющихся служб.

## 3.8. Сервер ALTO

Архитектура ABNO поддерживает совместную оптимизацию на сетевом и прикладном уровне. В таких случаях приложению представляется абстрактная топология сети, содержащая лишь сведения, относящиеся к нему. Приложение рассчитывает маршрутизацию прикладного уровня в соответствии со своими задачами и может взаимодействовать с контроллером ABNO для организации явных LSP, поддерживающих эту маршрутизацию.

Ниже указаны проиллюстрированы этапы действий, выполняемых в этом варианте применения.

1. Запрос приложением топологии прикладного уровня.

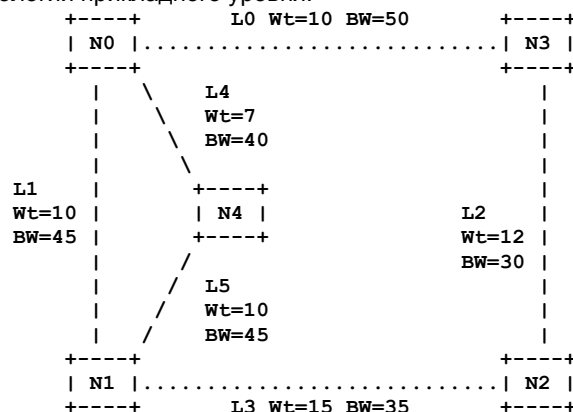


Рисунок 30. Необработанная топология сети.

На рисунке 30 показана сеть из 5 узлов и 6 каналов. Приложение с конечными точками в узлах N0, N1, N2 запрашивает топологию сети для расчёта маршрутов на прикладном уровне, например, для максимизации пропускной способности при репликации содержимого между конечными точками на трёх сайтах. Запрос приходит на контроллер ABNO, который пересылает его серверу ALTO. Сервер ALTO обращается к агенту правил, TED и PCE для возврата абстрактной топологии прикладного уровня.

Например, политика может задавать предоставление приложению пропускной способности не выше 40 Мбит/с. Сеть предварительно определила, что маршруту от N0 к N2 следует использовать путь N0->N3->N2 в соответствии с такими целями, как GCO (см. параграф 3.4). Сервер ALTO может создать для приложения сокращённую топологию, например, показанную на рисунке 31.

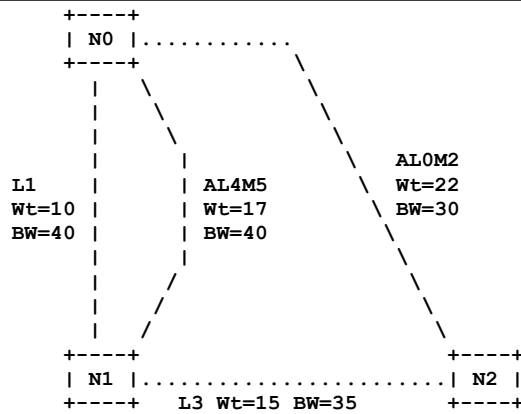


Рисунок 31. Сокращённый граф для конкретного приложения.

Сервер ALTO использует топологию и имеющиеся маршруты для расчёта абстрактной карты сети, состоящей из трёх PID. Парная пропускная способность, а также общие узкие места в сети могут быть вычислены по топологии внутренней сети и отражены в картах стоимости.

## 2. Расчёт приложением наложенной сети.

На основе абстрактной топологии приложение рассчитывает маршрутизацию на прикладном уровне. Приложение может рассчитать связующее дерево (spanning tree) для максимизации общей пропускной способности от N0 к N2. На рисунке 32 показан пример маршрутизации прикладного уровня с использованием путей N0->N1->N2 (35 Мбит/с) и N0->N2 (30 Мбит/с), что даёт в результате 65 Мбит/с.

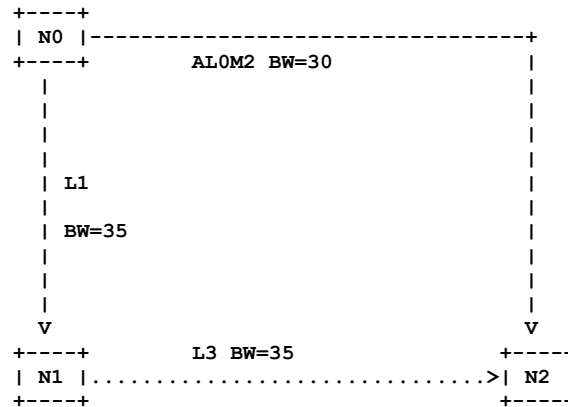


Рисунок 32. Связующее дерево прикладного уровня.

## 3. Организация контроллером ABNO путей для приложения.

Приложение может представить свои маршруты контроллеру ABNO для организации явных LSP, поддерживающих работу. Контроллер ABNO обращается к картам ALTO для сопоставления маршрутов прикладного уровня с топологией внутренней сети и даёт указание диспетчеру обеспечения организовать пути. Контроллер ABNO может снова вызвать GCO для повторной оптимизации организации сетевого трафика.

## 3.9. Другие возможные варианты применения

В этом параграфе указаны другие возможные варианты применения, которые могут быть описаны в будущем.

### 3.9.1. Обслуживание трафика

Этот вариант может включать несколько сценариев:

- вложенные LSP;
- классификация пакетов (потoki IP в LSP на граничных маршрутизаторах);
- «наполнение ведра» (Bucket Stuffing);
- потоки IP в «хеш-ведро» ECMP (Hash Bucket).

### 3.9.2. Планирование пропускной способности

Планирование пропускной способности заключается в настройке LSP на основе заданного расписания. Это может служить для поддержки графиков технического обслуживания и эксплуатации для регулировки пропускной способности сети на основе наблюдаемой картины трафика. Схема ABNO предоставляет компоненты для решения задач планирования пропускной способности.

## 4. Живучесть и резервирование в рамках архитектуры ABNO

Архитектура ABNO представлена в этом документе как функциональные блоки, каждый из которых реализуется отдельно или в связке с другими блоками как единая программа или продукция. Кроме того, каждый реализованный блок или набор блоков может размещаться на отдельном устройстве (например, сетевом сервере) или виртуальной машине (например, в ЦОД), а также возможно развёртывание группы программ на одном процессоре. С точки зрения архитектурной модели эти варианты реализации и внедрения не различаются.



Аналогично, реализация функционального компонента архитектуры ABNO может обеспечиваться не одним экземпляром или разными экземплярами разных реализаций, обеспечивающими одну или похожие функции. Например, компонент PCE может иметь несколько реализаций для распределенной обработки большого числа расчётных запросов и в разных экземплярах могут применяться различные алгоритмы, чтобы один экземпляр мог обеспечивать расчёт развязанных (параллельных) путей, в другой - оптимальных путей к группе получателей (point-to-multipoint).

Возможность иметь несколько экземпляров компонентов ABNO обеспечивает отказоустойчивость модели, когда при отказе одного экземпляра отдельного компонента (в результате программного или аппаратного сбоя или проблем связности) его работу принимают на себя другие экземпляры. В некоторых случаях может потребоваться синхронизация между экземплярами компонентов для обеспечения бесперебойно отказоустойчивости.

Вопросы обеспечения таких свойств реализаций ABNO выходят за рамки этого документа.

## 5. Вопросы безопасности

Архитектура ABNO описывает сетевую систему и безопасность должна играть важную роль. Прежде всего, нужно обеспечить надлежащую защиту внешних протоколов (входы и выходы блока, показанного точками на рисунке 1). Это включает проверку подлинности и полномочий при контроле доступа к различным функциям, выполняемым системой ABNO для поддержки различных правил на основе идентификации и для управления сетевыми устройствами. Во-вторых, внутренние протоколы, используемые между компонентами ABNO, также должны иметь надлежащую защиту, особенно при реализации этих компонентов на разных узлах сети.

С учётом наличия в системе ABNO большого объёма данных о сети, предоставляемых в ней услугах, а также услугах, предоставляемых клиентам, доступ к хранящейся в системе информации должен тщательно контролироваться. Поскольку такой доступ осуществляется в основном через внешние протоколы, контроль с аутентификацией на основе правил будет очень эффективным. Но следует отметить, что сведения, передаваемые из баз данных системы ABNO, могут раскрывать детали сети, поэтому следует рассмотреть вопрос об их шифровании. Поскольку компоненты ABNO могут иметь доступ к базам данных, требуются меры по проверке подлинности этих компонентов, а также целесообразно рассмотреть вопрос шифрования данных между компонентами при реализации тех на разных узлах.

Всем протоколам, применяемым для реализации архитектуры ABNO, следует иметь развитые средства защиты.

## 6. Вопросы управляемости

Архитектура ABNO в целом относится к управлению сетью. ABNO обеспечивает механизм сбора и обобщения сведений о сети, сообщает результаты управляющим приложениям, хранит их для будущего использования и запускает действия в соответствии с заданными правилами.

Система ABNO сама нуждается в мониторинге и управлении, которые можно разделить на несколько категорий.

- Управление внешними протоколами.
- Управление внутренними протоколами.
- Мониторинг компонентов ABNO и управление ими.
- Настройка правил, применяемых в системе ABNO.

## 7. Литература

- [BGP-LS] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", Work in Progress<sup>1</sup>, draft-ietf-idr-ls-distribution-10, January 2015.
- [CSO-PCE] Dhody, D., Lee, Y., Contreras, LM., Gonzalez de Dios, O., and N. Ciulli, "Cross Stratum Optimization enabled Path Computation", Work in Progress, draft-dhody-pce-cso-enabled-path-computation-07, January 2015.
- [EON] Gerstel, O., Jinno, M., Lord, A., and S.J.B. Yoo, "Elastic optical networking: a new dawn for the optical layer?", IEEE Communications Magazine, Volume 50, Issue 2, ISSN 0163-6804, February 2012.
- [Flood] Project Floodlight, "Floodlight REST API", <<http://www.projectfloodlight.org>>.
- [G.694.1] ITU-T Recommendation G.694.1, "Spectral grids for WDM applications: DWDM frequency grid", February 2012.
- [G.709] ITU-T Recommendation G.709, "Interface for the optical transport network", February 2012.
- [I2RS-Arch] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", Work in Progress<sup>2</sup>, draft-ietf-i2rs-architecture-09, March 2015.
- [I2RS-PS] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Interface to the Routing System Problem Statement", Work in Progress<sup>3</sup>, draft-ietf-i2rs-problem-statement-06, January 2015.
- [ONF] Open Networking Foundation, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)", October 2013.
- [PCE-Init-LSP] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", Work in Progress<sup>4</sup>, draft-ietf-pce-pce-initiated-lsp-03, March 2015.
- [RESTCONF] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", Work in Progress<sup>5</sup>, draft-ietf-netconf-restconf-04, January 2015.

<sup>1</sup>Опубликовано в [RFC 7752](#). Прим. перев.

<sup>2</sup>Опубликовано в [RFC 7921](#). Прим. перев.

<sup>3</sup>Опубликовано в [RFC 7920](#). Прим. перев.

<sup>4</sup>Опубликовано в [RFC 8281](#). Прим. перев.

<sup>5</sup>Опубликовано в [RFC 8040](#). Прим. перев.

- [RFC2748] Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000, <<http://www.rfc-editor.org/info/rfc2748>>.
- [RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", [RFC 2753](#), January 2000, <<http://www.rfc-editor.org/info/rfc2753>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC3292] Doria, A., Hellstrand, F., Sundell, K., and T. Worster, "General Switch Management Protocol (GSMP) V3", RFC 3292, June 2002, <<http://www.rfc-editor.org/info/rfc3292>>.
- [RFC3412] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002, <<http://www.rfc-editor.org/info/rfc3412>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003, <<http://www.rfc-editor.org/info/rfc3473>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003, <<http://www.rfc-editor.org/info/rfc3630>>.
- [RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", [RFC 3746](#), April 2004, <<http://www.rfc-editor.org/info/rfc3746>>.
- [RFC3985] Bryant, S., Ed., and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", [RFC 3985](#), March 2005, <<http://www.rfc-editor.org/info/rfc3985>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", [RFC 4655](#), August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC5150] Ayyangar, A., Kompella, K., Vasseur, JP., and A. Farrel, "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)", RFC 5150, February 2008, <<http://www.rfc-editor.org/info/rfc5150>>.
- [RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", [RFC 5212](#), July 2008, <<http://www.rfc-editor.org/info/rfc5212>>.
- [RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, October 2008, <<http://www.rfc-editor.org/info/rfc5254>>.
- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", [RFC 5277](#), July 2008, <<http://www.rfc-editor.org/info/rfc5277>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008, <<http://www.rfc-editor.org/info/rfc5305>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008, <<http://www.rfc-editor.org/info/rfc5394>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", [RFC 5424](#), March 2009, <<http://www.rfc-editor.org/info/rfc5424>>.
- [RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", [RFC 5440](#), March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009, <<http://www.rfc-editor.org/info/rfc5520>>.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009, <<http://www.rfc-editor.org/info/rfc5557>>.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009, <<http://www.rfc-editor.org/info/rfc5623>>.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009, <<http://www.rfc-editor.org/info/rfc5693>>.
- [RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", [RFC 5810](#), March 2010, <<http://www.rfc-editor.org/info/rfc5810>>.
- [RFC6007] Nishioka, I. and D. King, "Use of the Synchronization VECtor (SVEC) List for Synchronized Dependent Path Computations", RFC 6007, September 2010, <<http://www.rfc-editor.org/info/rfc6007>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6107] Shiomoto, K., Ed., and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011, <<http://www.rfc-editor.org/info/rfc6107>>.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011, <<http://www.rfc-editor.org/info/rfc6120>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, September 2012, <<http://www.rfc-editor.org/info/rfc6707>>.
- [RFC6805] King, D., Ed., and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012, <<http://www.rfc-editor.org/info/rfc6805>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, [RFC 7011](#), September 2013, <<http://www.rfc-editor.org/info/rfc7011>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014, <<http://www.rfc-editor.org/info/rfc7285>>.
- [RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, July 2014, <<http://www.rfc-editor.org/info/rfc7297>>.
- [Stateful-PCE] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", Work in Progress<sup>1</sup>, draft-ietf-pce-stateful-pce-10, October 2014.
- [TL1] Telcorida, "Operations Application Messages - Language For Operations Application Messages", GR-831, November 1996.
- [TMF-MTOSI] TeleManagement Forum, "Multi-Technology Operations Systems Interface (MTOSI)", <<https://www.tmforum.org/MTOSI/2319/home.html>>.
- [YANG-Rtg] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", Work in Progress<sup>2</sup>, draft-ietf-netmod-routing-cfg-17, March 2015.

## Приложение А. Не определённые интерфейсы

В этом приложении кратко перечислены интерфейсы, которые ещё не были определены на момент написания документа. Интерфейсы, для которых имеется возможность выбора, не включены в список.

- Интерфейс для добавления в TED дополнительных сведений описан в параграфе 2.3.2.3. В настоящее время для этого интерфейса протокол не определён, но есть кандидаты:
  - протокол, разработанный или адаптированный в соответствии с требованиями I2RS [I2RS-Arch];
  - NETCONF [RFC6241].
- Протокол, используемый интерфейсом в систему маршрутизации (I2RS), описан в параграфе 2.3.2.8. Рабочая группа I2RS определила, что этот протокол будет основан на сочетании NETCONF [RFC6241] и RESTCONF [RESTCONF] с дополнениями и изменениями, которые будут сочтены необходимыми для поддержки желаемой функции. Детали протокола ещё предстоит определить.
- Как указано в параграфе 2.3.2.10, менеджеру топологии виртуальных сетей (VNTM) нужен интерфейс, который PCE или контроллер ABNO может применять для информирования о потребности клиентского уровня в дополнительной виртуальной топологии. Возможно, что для этого подойдёт протокол, выбранный для использования с I2RS, или это можно сделать с использованием расширения сообщений PCEP Notify (PCNtf).
- Северный интерфейс контроллера ABNO используется NMS, OSS и координатором прикладных служб для запроса услуг в сети в поддержку приложений, как описано в параграфе 2.3.2.11.
  - Возможно, для этого подойдёт протокол, выбранный в соответствии с требованиями I2RS.
  - Возможный подход к интерфейсу этого типа описан в [RFC7297] для простого варианта применения.
- Как указано в параграфе 2.3.2.14, возможны независимые от уровня модели данных для предоставления базовых интерфейсов управления, настройки и отчётов OAM.
- Как отмечено в параграфе 3.6, модель ABNO подходит для включения многосегментных псевдопроводов в топологию сети, состоящей из туннелей S-PE и MPLS. Текущее определение протокола PCEP [RFC5440] и связанные с ним расширения, работа над которыми ещё не завершена, не включают всех деталей запроса таких путей, поэтому может потребоваться доработка, хотя общие концепции остаются применимыми. Такая работа может потребоваться для расширения применений PCE в различных вариантах сетей.

## Благодарности

Спасибо за обсуждения и рецензии Ken Gray, Jan Medved, Nitin Bahadur, Diego Caviglia, Joel Halpern, Brian Field, Ori Gerstel, Daniele Ceccarelli, Cyril Margaria, Jonathan Hardwick, Nico Wauters, Tom Taylor, Qin Wu, Luis Contreras. Спасибо George Swallow за сообщение о наличии базы данных SRLG. Tomonori Takeda и Julien Meuric предоставили ценные комментарии в рамках обзора Routing Directorate, Tina Tsou - в рамках обзора Operational Directorate.

Эта работа финансировалась по программе Евросоюза Seventh Framework по исследованию, технологической разработке и демонстрации в рамках проекта PACE по грантовому соглашению 619712 и в рамках проекта IDEALIST по грантовому соглашению 317999.

<sup>1</sup>Опубликовано в RFC 8231. Прим. перев.

<sup>2</sup>Опубликовано в RFC 8022. Прим. перев.

## **Участники работы**

**Quintin Zhao**

Huawei Technologies  
125 Nagog Technology Park  
Acton, MA 01719  
United States  
EMail: [qzhao@huawei.com](mailto:qzhao@huawei.com)

**Victor Lopez**

Telefonica I+D  
EMail: [vlopez@tid.es](mailto:vlopez@tid.es)

**Ramon Casellas**

CTTC  
EMail: [ramon.casellas@cttc.es](mailto:ramon.casellas@cttc.es)

**Yuji Kamite**

NTT Communications Corporation  
EMail: [y.kamite@ntt.com](mailto:y.kamite@ntt.com)

**Yosuke Tanaka**

NTT Communications Corporation  
EMail: [yosuke.tanaka@ntt.com](mailto:yosuke.tanaka@ntt.com)

**Young Lee**

Huawei Technologies  
EMail: [leeyoung@huawei.com](mailto:leeyoung@huawei.com)

**Y. Richard Yang**

Yale University  
EMail: [yry@cs.yale.edu](mailto:yry@cs.yale.edu)

## **Адреса авторов**

**Daniel King**

Old Dog Consulting  
EMail: [daniel@olddog.co.uk](mailto:daniel@olddog.co.uk)

**Adrian Farrel**

Juniper Networks  
EMail: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

## **Перевод на русский язык**

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)