

Internet Research Task Force (IRTF)
Request for Comments: 7426
Category: Informational
ISSN: 2070-1721

E. Haleplidis, Ed.
University of Patras
K. Pentikousis, Ed.
EICT
S. Denazis
University of Patras
J. Hadi Salim
Mojatatu Networks
D. Meyer
Brocade
O. Koufopavlou
University of Patras
January 2015

Программно-определяемые сети (SDN) - уровни и архитектура Software-Defined Networking (SDN): Layers and Architecture Terminology

Тезисы

Программно-определяемые сети (SDN¹) определяют новый подход к программируемости сетей, т. е. возможности динамической инициализации, управления, изменения и администрирования поведения сети через открытые интерфейсы. SDN усиливает роль программ в работе сетей за счет введения абстракции для уровня пересылки данных (data forwarding plane) и связанного с этим отделения уровня пересылки от уровня управления (control plane). Такое разделение ускоряет инновационные циклы для обоих уровней, как показывает имеющийся опыт. Однако накапливается и путаница в части понимания, что такое SDN, какова структура уровней в архитектуре SDN и как эти уровни взаимодействуют между собой. Этот документ, являющийся результатом работы группы IRTF SDN RG², разрешает эти вопросы и содержит краткий справочник для исследовательского сообщества SDN на основе рецензирования литературы, RFC и документов, выпущенных другими органами стандартизации.

Статус документа

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

Документ является результатом работы IRTF³. Группа IRTF публикует результаты связанных с Internet исследований и разработок. Эти результаты не обязательно пригодны для реализации. Данный RFC представляет согласованное мнение исследовательской группы Software-Defined Networking в составе IRTF. Документ был одобрен для публикации IRTF и не претендует на статус стандарта Internet (см. раздел 2 документа RFC 5741).

Информация о статусе этого документа, обнаруженных ошибках и способах обратной связи доступна по ссылке <http://www.rfc-editor.org/info/rfc7426>.

Авторские права

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

Этот документ является субъектом прав и ограничений, перечисленных в BCP 78 и IETF Trust Legal Provisions и относящихся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно, поскольку в них описаны права и ограничения, относящиеся к данному документу.

Оглавление

1. Введение.....	2
2. Терминология.....	2
3. Уровни и архитектура SDN.....	3
3.1. Обзор.....	3
3.2. Сетевые устройства.....	5
3.3. Уровень управления.....	6
3.4. Уровень администрирования.....	6
3.5. Обсуждение уровней управления и администрирования.....	7
3.5.1. Временной масштаб.....	7
3.5.2. Продолжительность состояний.....	7
3.5.3. Расположение.....	7
3.5.4. Теорема CAP.....	7
3.6. Уровень абстракции сетевых служб.....	7
3.7. Уровень приложений.....	8
4. Представление модели SDN.....	8
4.1. ForCES.....	8
4.2. NETCONF/YANG.....	8
4.3. OpenFlow.....	9
4.4. Интерфейс с системой маршрутизации.....	9

¹Software-Defined Networking.

²Software-Defined Networking Research Group - группа по исследованию программно-определяемых сетей.

³Internet Research Task Force - группа по исследованию технологий Internet.

4.5. SNMP.....	9
4.6. PCEP.....	9
4.7. BFD.....	10
5. Заключение.....	10
6. Вопросы безопасности.....	10
7. Литература.....	10
Благодарности.....	13
Участники работы.....	13
Адреса авторов.....	13

1. Введение

Термин «программно-определяемые сети» (SDN) обозначает парадигму программируемых сетей [PNSurvey99] [OF08]. Вкратце, SDN означает способность программных приложений динамически управлять (программировать) отдельными сетевыми устройствами и за счет этого управлять поведением сети в целом [NV09]. Boucadair и Jacquet [RFC7149] показали, что SDN представляет собой набор технологий для упрощения разработки, доставки и использования сетевых служб динамичным, детерминированным и масштабируемым способом.

Ключевым элементом SDN является разделение (традиционных) уровней пересылки (forwarding) и управления (control) для того, чтобы обеспечить приложениям возможности, требуемые для программного управления сетью. Цель состоит в дальнейшем разделении и связанной с этим программируемости для снижения уровня сложности и ускорения развития обоих уровней [A4D05].

Эволюция исследований и разработок в сфере программируемых сетей подробно рассмотрена в [SDNHistory] [SDNSurvey], начиная с 1980-х годов. Как отмечено в [SDNHistory], многие идеи, концепции и проблемы применимы и к более поздним исследованиям и разработкам SDN (и стандартизации SDN), а также остаются активными направлениями исследования и обсуждения в современном сообществе исследователей. Например, Rooney и др. [Tempest] рассматривают вопросы предоставления третьим сторонам доступа в сеть без создания угрозы ее целостности, а также приспособления унаследованных сетевых решений в их (тогда) новую программируемую среду. Кроме того, концепция разделения уровней управления и пересылки, являющаяся важной частью SDN, широко обсуждалась еще до 1998 года [Tempest] [P1520] в сетях SS7 [ITUSS7], Ipsilon Flow Switching [RFC1953] [RFC2297] и ATM [ITUATM].

Исследования SDN часто сосредоточены на разных аспектах программируемости и нередко возникают противоречивые точки зрения на саму природу SDN. Отмечено, что в силу разных причин (например, концентрация работ на одной области, когда результаты могут оказаться не применимыми в других областях), некоторые общепринятые определения не коррелируют между собой. Например OpenFlow [OpenFlow] и NETCONF¹ [RFC6241] считаются интерфейсами SDN, но относятся к управлению и администрированию, соответственно.

Это диктует необходимость консолидации определений SDN в литературе и сопоставления их с ранними работами в IETF и сообществе исследователей. Особый интерес представляет, например, определение уровней, входящих в архитектуру SDN, а также интерфейсов и связанных с ними атрибутов, которые обеспечат лучшее взаимодействие между этими уровнями. Таким образом, целью настоящего документа является не стандартизация какого-либо конкретного уровня или интерфейса, а скорее предоставление кратких справок, отражающих текущие подходы к многоуровневой архитектуре SDN. Предполагается, что этот документ будет полезен в предстоящих работах SDN RG, а также обсуждениях в рамках всего сообщества SDN.

В этом документе рассматривается направление работы SDN RG, названное «Обзор архитектуры и систематика SDN²», способствующее лучшему пониманию основных технологий SDN без привязки к технологиям и бизнесу, не задающее нового стандарта IETF. Это служит основой для дальнейшего обсуждения. Таким образом, мы не делаем каких-либо важных заявлений и не обсуждаем применимость любой из рассмотренных в документе моделей для решения тех или иных конкретных задач. Вместо этого рассматриваются характеристики, атрибуты и классификация моделей для обеспечения их систематики. Документ не содержит исчерпывающего списка направлений исследования SDN, интересующихся этим вопросом читателей следует обратиться к обзорам [SLTSDN] и [SDNACS]. В частности, Jaggau и др. в [SLTSDN] представили обзор связанных с SDN исследований, например, разделение управления, связанное с теоремой CAP³, рассмотренной в параграфе 3.5.4.

Этот документ был представлен для широкого обзора, обсуждения и комментариев большинству членов SDNRG, число которых превышало 100 человек. В SDN RG было достигнуто соглашение о целесообразности публикации документа в потоке IRTF в рамках серии RFC [RFC5743].

Оставшаяся часть документа организована следующим образом - раздел 2 посвящен используемой в документе терминологии, в разделе 3 представлен обзор современных абстракций архитектуры SDN, а раздел 4 рассматривает связь многоуровневой архитектуры SDN с известными технологиями поддержки SDN.

2. Терминология

Ниже приведены определения используемых в документе терминов.

Software-Defined Networking (SDN) – программно-определяемые сети

Модель программируемых сетей, которая поддерживает разделение уровней пересылки и управления через стандартизованные интерфейсы.

Resource - ресурс

Физическая или виртуальная компонента, доступная в системе. Ресурсы могут быть очень простыми и четко ограниченными (например, порт или очередь) или комплексными, состоящими из множества других ресурсов (например, сетевое устройство).

Network Device – сетевое устройство

Устройство, выполняющее одну или множество сетевых операций, относящихся в пересылке и манипуляциям с пакетами. Эталонная модель не различает физические и виртуальные сетевые устройства. Устройство может рассматриваться как контейнер для ресурсов или как ресурс.

¹Network Configuration Protocol - протокол настройки сети.

²Оригинальное название направления - Survey of SDN approaches and Taxonomies.

³Consistency, Availability and Partitioning - согласованность, доступность и разделение.

Interface - интерфейс

Точка взаимодействия между двумя элементами (сущностями). При размещении элементов в разных местах интерфейс между ними обычно реализуется через сетевой протокол. Если элементы расположены в одном месте, интерфейс может быть реализован с помощью программного API¹, межпроцессных коммуникаций (IPC²) или сетевого протокола.

Application (App) – приложение (программа)

Приложением в контексте SDN считается часть программы, использующая услуги нижележащего уровня для выполнения своих функций. Работа приложения может быть параметризована, например, путем передачи неких аргументов при вызове, но это относится к автономным частям программ. Приложение не предоставляет каких-либо интерфейсов для других приложений или служб.

Service – сервис (служба)

Часть программы, которая выполняет одну или множество функций и предоставляет один или множество интерфейсов API для приложений или других служб на том же или других уровнях, позволяя вызывать функции и возвращая один или множество результатов. Службы могут объединяться с другими службами или вызываться последовательно для создания нового сервиса.

Forwarding Plane (FP) – уровень пересылки

Набор ресурсов в масштабе всей сети, отвечающих за пересылку трафика.

Operational Plane (OP) – операционный уровень

Набор ресурсов, отвечающих за общее управление работой отдельных сетевых устройств.

Control Plane (CP) – уровень управления

Набор функций, отвечающих за управление одним или множеством сетевых устройств. CP задает поведение сетевых устройств в части обработки и пересылки пакетов. Уровень управления взаимодействует, прежде всего, с уровнем пересылки и, в меньшей степени, с операционным уровнем.

Management Plane (MP) – уровень администрирования

Набор функций, отвечающих за мониторинг, настройку и администрирование одного или множества сетевых устройств или их частей. Уровень администрирования связан, прежде всего, с операционным уровнем и в меньшей степени с уровнем пересылки.

Application Plane – уровень приложений

Набор приложений и служб, программирующих поведение сети.

Device and resource Abstraction Layer (DAL) – уровень абстракции устройств и ресурсов

Уровень абстракции ресурсов устройства, основанный на одной или множестве моделей. Для физических устройств этот уровень может называться уровнем абстракции оборудования (HAL³). DAL обеспечивает единую точку отсчета для ресурсов уровней пересылки и операций в устройстве.

Control Abstraction Layer (CAL) – уровень абстракции управления

Уровень абстракции управления. CAL обеспечивает доступ к интерфейсу «южной границы» уровня управления (Control-Plane Southbound Interface).

Management Abstraction Layer (MAL) – уровень абстракции администрирования

MAL обеспечивает доступ к интерфейсу «южной границы» уровня администрирования (Management-Plane Southbound Interface).

Network Services Abstraction Layer (NSAL) – уровень абстракции сетевых служб

Обеспечивает абстракции сервиса, которые могут применяться службами и приложениями.

3. Уровни и архитектура SDN

На рисунке 1 показаны абстрактные компоненты архитектуры SDN в форме достаточно подробной схемы верхнего уровня. Отметим, что в конкретной реализации разные уровни (plane) могут размещаться совместно или разделяться физически, как будет описано ниже.

SDN базируется на концепции разделения управляющих и управляемых объектов. Контроллер манипулирует управляемым объектом через некий интерфейс. Локальные интерфейсы обеспечиваются в основном путем вызова функций API из той или иной библиотеки или системных вызовов. Однако такие интерфейсы можно расширить с использованием протоколов, которые могут применять локальные коммуникации между процессами (IPC), или протоколов, работающих удаленно. Эти протоколы могут быть реализованы как открытые стандарты или фирменные (proprietary) решения.

Day в книге [PiNA] исследовал применение IPC в качестве основы для определения вариантов рекурсивной сетевой архитектуры с разной областью действия и набором операций. Модель [RINA⁴] описывает рекурсивную сетевую архитектуру на основе IPC, которая использует повторяющиеся шаблоны и структуры. Данный документ не предлагает новую архитектуру, здесь лишь рассматриваются предшествующие работы для систематизации. Хотя рекурсия выходит за рамки этой работы, на рисунке 1 показана иерархическая модель, в которой уровни могут образовывать стек над каждым другим уровнем, реализуя при необходимости рекурсию.

3.1. Обзор

В этом документе используется модель, центром которой является сетевое устройство. Уровень управления относится в основном к возможностям обработки пакетов устройством, а уровень администрирования - к общим аспектам работы устройства. Сетевое устройство рассматривается, как комплексный ресурс, который включает в себя или является частью множества ресурсов типа [DIOPR]. Ресурсы могут быть простыми, однокомпонентными сетевыми устройствами (например, порт или очередь в устройстве) или объединяться в комплексные ресурсы (например, сетевой адаптер или полное сетевое устройство).

Читателю следует принимать во внимание, что в этом документе не делается различий между физическими и виртуальными ресурсами, а также между программными и аппаратными реализациями, поскольку мы не углубляемся в вопросы реализации или производительности. Иными словами, ресурс может быть полностью реализован аппаратно или программно, а может быть гибридным. Кроме того, мы не различаем, реализуется ли ресурс как наложение (overlay) или часть (компонента) некоего другого устройства. В общем случае программы сетевого устройства могут

¹Application programming interface - интерфейс с прикладными программами.

²Inter-process communication.

³Hardware Abstraction Layer.

⁴Recursive InterNetwork Architecture - рекурсивная межсетевая архитектура.

работать на «голом железе» (bare metal) или виртуализованной подложке. И, наконец, в документе не рассматривается выделение, организация и освобождение ресурсов.

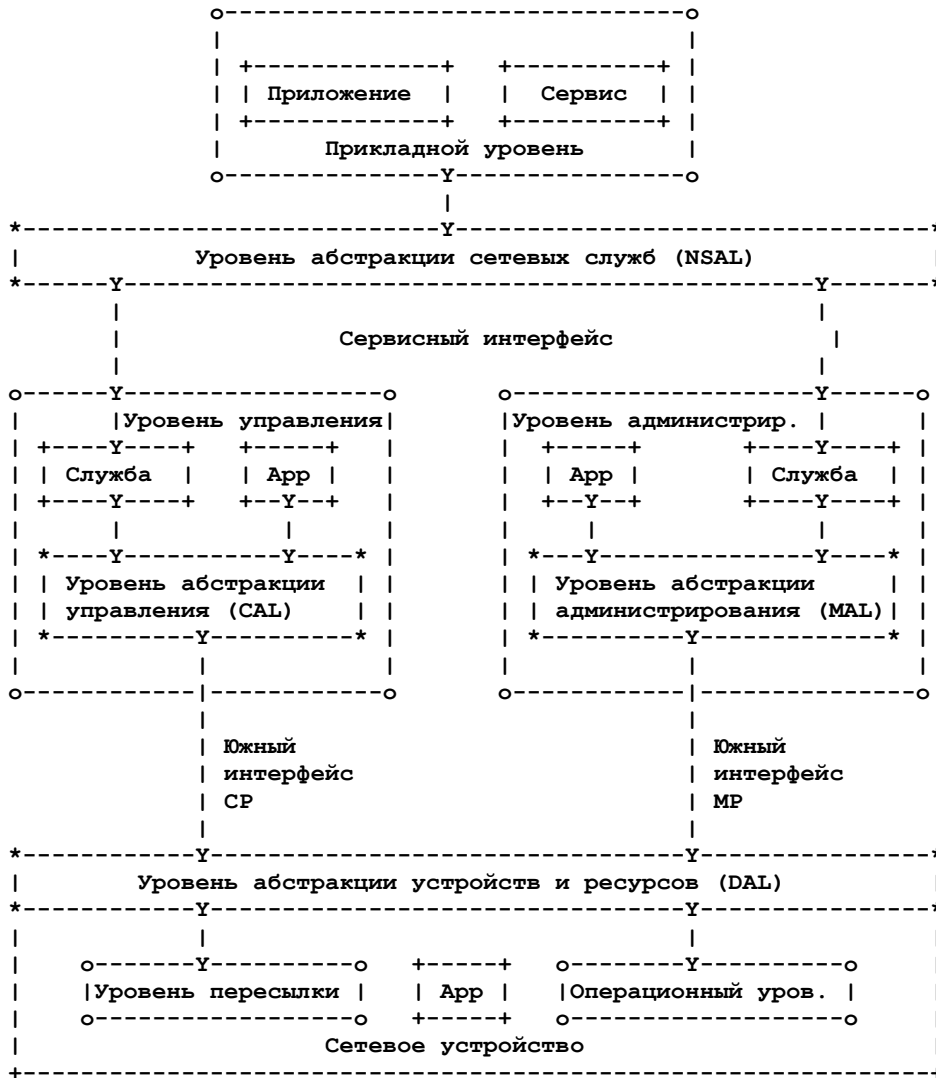


Рисунок 1. Многоуровневая архитектура SDN.

SDN включает множество уровней, показанных на рисунке 1. Начиная с нижней части рисунка и перемещаясь вверх можно видеть перечисленные ниже уровни.

- **Уровень пересылки** отвечает за обработку пакетов в путях данных на основе инструкций, полученных от уровня управления. Операции уровня пересылки включают пересылку, изменение и отбрасывание пакетов (могут включаться и другие операции). Уровень пересылки обычно является точкой завершения (termination point) для служб и приложений уровня управления. Уровень пересылки может включать ресурсы типа классификаторов. Этот уровень зачастую называют уровнем данных (data plane) или путем данных (data path).
- **Операционный уровень** отвечает за управление рабочим состоянием сетевого устройства, например, его активностью, числом доступных портов, статусом каждого порта и т. п. Операционный уровень обычно служит точкой завершения для служб и приложений уровня администрирования. Операционный уровень связан с ресурсами сетевого устройства типа портов, памяти и т. п. Отметим, что у некоторых членов группы IRTF SDN RG имеется другое мнение по части определения операционного уровня. Т. е. можно сказать, что операционный уровень не самостоятелен и на практике является набором функций уровня пересылки. С другой стороны, этот уровень позволяет выделить область действия операций и в этом смысле операционный уровень указан отдельно на рисунке 1. В документе используется именно такой подход.
- **Уровень управления** отвечает за принятие решений о том, как пакеты должны пересылаться одним или множеством сетевых устройств, и «выталкивает» эти решения вниз для выполнения их сетевыми устройствами. Уровень управления обычно связан с уровнем пересылки и в меньшей степени с операционным уровнем устройства. Уровень управления может быть заинтересован в информации операционного уровня, которая может включать, например, текущее состояние конкретного порта или его возможности. Основная работа уровня управления заключается в «тонкой настройке» таблиц, относящихся к уровню пересылки, на основе топологии сети или запросов внешних служб.
- **Уровень администрирования** отвечает за мониторинг, настройку и администрирование сетевых устройств (например, принятие решений относительно состояния сетевого устройства). Уровень администрирования обычно взаимодействует с операционным уровнем и в меньшей степени с уровнем пересылки. Уровень администрирования может использоваться для настройки уровня пересылки, но делает это нечасто и более масштабно (на уровне устройства в целом), нежели операционный уровень. Например, уровень администрирования может разом установить правила пересылки для всех портов, хотя предполагается, что такие действия будут производиться редко.
- **Уровень приложений**, на котором размещаются приложения и службы, определяющие поведение устройства. Приложения, которые непосредственно (или в первую очередь) поддерживают операции уровня пересылки

(типа процесса маршрутизации на уровне управления), не считаются частью уровня приложений. Отметим, что приложения могут быть реализованы в форме модулей и распределены, поэтому зачастую могут охватывать разные уровни, показанные на рисунке 1.

[RFC7276] определяет уровни данных, управления и администрирования в терминах OAM¹. Этот документ пытается расширить действие терминов, определенных в [RFC7276], с учетом всех аспектов архитектуры SDN.

Все отмеченные выше уровни соединяются через интерфейсы (Y на рисунке 1). Интерфейс может играть множество ролей в зависимости от того, размещаются ли соединяемые уровни на одном (физическом или виртуальном) устройстве. Если соответствующие уровни устройств так, что не размещаются на одном устройстве, интерфейс может принимать лишь форму протокола. Если уровни совмещаются на одном устройстве, интерфейс может быть реализован на основе открытого или фирменного протокола, открытого или фирменного API межпроцессного взаимодействия, а также вызовов ядра операционной системы.

Приложения, т. е. программы, выполняющие конкретные расчеты и использующие службы без предоставления доступа к другим приложениям, могут быть реализованы естественным путем внутри одного уровня или охватывать несколько уровней. Например, приложения или службы могут охватывать уровни управления и администрирования и, таким образом, иметь возможность использования обоих интерфейсов CPSI² и MPSI³, хотя это не указано явно на рисунке 1. Примером этого может служить приложение, использующее сразу [OpenFlow] и [OF-CONFIG].

Службы, т. е. программы, обеспечивающие API для приложений или других служб, также могут быть естественным образом реализованы на конкретных уровнях. Службы, охватывающие несколько уровней, относятся также к уровню приложений.

Хотя это и не показано явно на рисунке 1, службы, приложения и целые уровни могут размещаться рекурсивно, обеспечивая в этой модели семантику наложения (перекрытия). Например, услуги уровня приложений могут обеспечиваться для других услуг или приложений через NSAL. Другие примеры включают виртуальные ресурсы, реализованные на основе физических ресурсов и иерархических контроллеров уровня управления [KANDOO].

Отметим, что этот документ посвящен коммуникациям «север-юг» между разными уровнями. Очевидно, что это не исключает коммуникаций между элементами внутри одного уровня.

Однако нужно отметить, что на рисунке 1 представлен абстрактный вид разных уровней без деталей реализации. Многие реализации прошлых лет размещали уровень администрирования над уровнем управления. Это можно интерпретировать, как использование уровня управления в качестве сервиса для уровня администрирования. Более того, во многих сетях, особенно на маршрутизаторах Internet и коммутаторах Ethernet уровень управления был реализован в тесной связи с сетевым устройством. При этом уровень управления в целом был распределен по всей сети. С другой стороны, уровень администрирования обычно был централизованным и отвечал за администрирование уровня управления и всех устройств. Однако с принятием принципов SDN это различие утратило четкость.

В дополнение к перечисленным выше уровням данный документ рассматривает четыре абстрактных уровня.

- Абстракция DAL представляет ресурсы уровней пересылки и операций для уровней управления и администрирования. Варианты DAL могут абстрагировать оба уровня или любой из двух уровней устройства для любого из уровней администрирования и управления.
- Абстракция CAL представляет интерфейс южной границы уровня управления и DAL от приложений и служб для уровня управления.
- Абстракция MAL представляет интерфейс южной границы уровня администрирования и DAL от приложений и служб для уровня администрирования.
- Абстракция NSAL представляет службы для использования приложениями и другими службами.

На момент написания этого документа связанные с SDN работы начались и в других органах стандартизации (SDO). Например, в ITU разработаны архитектурные [ITUSG13] и сигнальные требования и протоколы [ITUSG11], но соответствующие исследовательские группы еще не опубликовали свои работы, за исключением [ITUY3300]. Взгляды, представленные в [ITUY3300] и [ONFArch], согласуются с этим документом.

3.2. Сетевые устройства

Сетевое устройство - это объект, принимающий пакеты на своих портах и выполняющий по отношению к ним одну или множество функций. Например, сетевое устройство может пересылать принятые пакеты, отбрасывать их, менять заголовок (или данные) и т. п. Сетевое устройство представляет собой агрегат множества ресурсов, таких как порты, CPU, память и очереди. Ресурсы могут быть простыми или агрегированными для формирования комплексного ресурса, воспринимаемого как целое. Сетевое устройство само по себе является комплексным ресурсом. Примерами сетевых устройств служат коммутаторы и маршрутизаторы. Другими примерами могут служить элементы сети, выполняющие функции выше (межсетевые экраны, балансировщики нагрузки, видеотранскодеры) или ниже уровня IP (коммутаторы L2, оптические и радиочастотные элементы сетей).

Сетевые устройства могут быть реализованы в виде оборудования или программ, а также быть физическими или виртуальными. Как уже было отмечено выше, в этом документе не делается различий между этими типами. В каждом сетевом устройстве имеется уровень пересылки и операционный уровень.

Уровень пересылки, зачастую называемый путем данных (data path), отвечает за обработку и пересылку пакетов. Этот уровень обеспечивает коммутацию и маршрутизацию, преобразование и фильтрацию пакетов. Ресурсы уровня пересылки включают фильтры, измерители, маркеры, классификаторы и могут включать иные типы ресурсов.

Операционный уровень отвечает за рабочее состояние сетевого устройства, например, за статус сетевых портов и интерфейсов. Ресурсы операционного уровня включают память, CPU, порты, интерфейсы, очереди и могут включать иные типы ресурсов.

Уровни пересылки и операций представляются через абстракцию DAL, которая может быть выражена одной или несколькими моделями. Примерами моделей абстракции уровня пересылки являются ForCES⁴ [RFC5812], OpenFlow [OpenFlow], YANG [RFC6020] и SNMP MIB [RFC3418]. Примеры моделей абстракции операционного уровня включают ForCES [RFC5812], YANG [RFC6020], SNMP MIB [RFC3418].

¹Operations, Administration, and Maintenance - операции, администрирование, обслуживание.

²Control-Plane Southbound Interface - интерфейс южной границы уровня управления.

³Management-Plane Southbound Interface - интерфейс южной границы уровня администрирования.

⁴Forwarding and Control Element Separation - разделение элементов пересылки и управления.

Отметим, что приложения также могут размещаться в сетевых устройствах. Примеры таких приложений включают мониторинг событий, определение топологии или обслуживание ARP [RFC0826] в самом устройстве вместо пересылки такого трафика на уровень управления.

3.3. Уровень управления

Уровень управления обычно распределен и отвечает главным образом за настройку уровня пересылки с использованием интерфейса CPSI с DAL в качестве опорной точки. Уровень управления (CP) отвечает за инструктирование уровня пересылки (FP) по части обработки сетевых пакетов.

Коммуникации между объектами уровня управления, которые иногда называют интерфейсом «восток-запад», обычно реализуются через протоколы шлюзов типа BGP [RFC4271] или PCEP¹ [RFC5440]. Сообщения соответствующего протокола обычно передаются по основному каналу (in-band) и затем перенаправляются уровнем пересылки на уровень управления для дальнейшей обработки. Примеры этой категории включают [RCP], [SoftRouter] и [RouteFlow].

Функциональность уровня управления обычно включает:

- определение и поддержку топологии;
- выбор и организацию маршрутов для пакетов;
- механизмы восстановления путей при отказах.

CPSI обычно определяется со следующими характеристиками:

- критичный ко времени интерфейс, требующий малой задержки и в некоторых случаях высокой пропускной способности для выполнения множества операций за короткое время;
- ориентация на эффективность передачи через «провод» и представление устройству, а не человеку.

Примеры включают быстрые и частые обновления потоков или таблиц, высокую пропускную способность и отказоустойчивость для обработки пакетов и событий.

CPSI может быть реализован с помощью протокола, API или межпроцессных коммуникаций. Если уровень управления и сетевое устройство не размещены вместе, этот интерфейс, безусловно, является протоколом. Примерами CPSI являются ForCES [RFC5810] и OpenFlow [OpenFlow].

Абстракция CAL обеспечивает доступ управляющим приложениям и службам к разным CPSI. Уровень управления может поддерживать более одного интерфейса CPSI.

Управляющие приложения могут использовать CAL для управления сетевым устройством без предоставления услуг вышележащим уровням. Примеры включают приложения, выполняющие функции управления, типа OSPF, IS-IS, BGP.

Примерами служб уровня управления служат виртуальные частные ЛВС, туннели, топологические службы и т. п.

3.4. Уровень администрирования

Уровень администрирования обычно централизован и предназначен для обеспечения оптимальной работы сети в целом за счет взаимодействия с операционным уровнем сетевых устройств с использованием интерфейса MPSI с DAL в качестве опорной точки.

Функциональность уровня администрирования обычно начинается с представления сети в целом и традиционно рассчитана на восприятие человеком. Однако в последнее время действия человека все больше заменяют алгоритмами. Функциональность уровня администрирования [FCAPS] обычно включает:

- администрирование отказов и мониторинга;
- управление конфигурацией.

В дополнение к этому функциональность уровня администрирования может также включать управление функциями виртуальных сетей (VNF Manager²) и виртуализованной инфраструктуры как описано в [NFVArch]. Такие элементы могут использовать интерфейсы администрирования к ресурсам уровня управления для запроса и предоставления ресурсов виртуальным функциям, а также для организации виртуальных функций пересылки на базе аналогичных физических функций. Возможность создания общей абстрактной модели для SDN и виртуализации сетевых функций (NFV³) рассмотрена в [SDNNFV]. Отметим однако, что имеются лишь примеры приложений и служб на уровне управления, а формальных определений объектов в этом документе не дано. Как было отмечено выше, организация и определение любых связанных с этим объектов выходят за рамки документа.

Интерфейс MPSI, в отличие от CPSI, обычно не критичен ко времени и не разделяет предъявляемых CPSI требований.

MPSI ближе к человеку, чем CPSI (см. [RFC3535]), поэтому MPSI обычно имеет приведенные ниже характеристики:

- интерфейс ориентирован в основном на удобство, а эффективность «в проводе» не столь важна;
- сообщения передаются не столь часто, как в CPSI.

В качестве примера баланса между производительностью и удобством отметим соглашение, принятое на 2002 IAB Workshop [RFC3535] - ключевым требованием для технологии сетевого администрирования является простота использования, а не производительность. Как отмечено в [RFC6632], текстовые конфигурационные файлы должны поддерживать возможность включения символов разных языков. В предназначенных для человека строках следует применять кодировку UTF-8, а протокольные элементы должны быть строками ASCII без учета регистра символов, который требует более сложной обработки.

MPSI может варьироваться от протокола до API или даже межпроцессных коммуникаций. Если уровень администрирования не встроен в сетевое устройство, MPSI обычно является протоколом. Примерами MPSI являются ForCES [RFC5810], NETCONF [RFC6241], IPFIX⁴ [RFC7011], Syslog [RFC5424], Open vSwitch Database (OVSDB) [RFC7047] и SNMP [RFC3411].

Абстракция MAL обеспечивает службам и приложениям доступ к разным MPSI. Уровень администрирования может поддерживать более одного MPSI.

¹Path Computation Element (PCE) Communication Protocol - коммуникационный протокол элементов расчета пути.

²Virtual Network Function Manager.

³Network Function Virtualization.

⁴IP Flow Information Export - экспорт информации о потоках IP.

Приложения для администрирования могут использовать MAL для управления сетевым устройством, не предоставляя своих услуг вышележащим уровням. Примеры таких приложений включают сетевой мониторинг, обнаружение отказов и восстановление.

Службы уровня администрирования обеспечивают доступ для других служб или приложений выше административного уровня.

3.5. Обсуждение уровней управления и администрирования

Определение четкого различия между управлением (control) и администрированием (management) в контексте SDN привлекло внимание сообщества в процессе подготовки этого документа. Было отмечено, что раньше уровень администрирования по большей части игнорировался или выносился за пределы экосистемы SDN. Далее в этом разделе описаны характеристики различий между уровнями управления и администрирования для лучшего понимания механизмов, возможностей и потребностей соответствующих интерфейсов.

3.5.1. Временной масштаб

Был поднят вопрос о временном масштабе для уровней управления и администрирования в плане скорости реакции и требуемой быстротой изменений, вносимых на уровне пересылки или операций. В общем случае уровню управления требуется передавать свои обновления «часто» - в интервале миллисекунд, что требует каналов с высокой пропускной способностью и малыми задержками. Уровень администрирования, напротив, обычно работает в более продолжительных временных рамках (минуты, часы и даже дни), поэтому высокая скорость не так важна. Хорошим примером может служить изменение конфигурации устройства.

3.5.2. Продолжительность состояний

Другим различием между уровнями управления и администрирования является продолжительность сохранения состояний. Состояние считается краткосрочным (эфемерным), если оно сильно ограничено во времени и не требует записи в энергонезависимую память. Хорошим примером служит картина маршрутизации, которая обычно связана с уровнем управления. Долгосрочное состояние существует в течение достаточно продолжительного времени (от часов до дней и даже месяцев), предназначено для использования за пределами срока действия создавшего его процесса и может сохраняться при перезагрузке устройства. Долгосрочные состояния обычно связаны с уровнем администрирования.

3.5.3. Расположение

Как было отмечено выше, уровень управления обычно выполняется на сетевых устройствах и по своей природе распределен между ними, а уровень администрирования обычно выполняется централизованно на удалении от устройств. Однако с появлением централизации SDN (логической централизации) обнаружилась тенденция контроллеров путать различие между уровнями управления и администрирования в части их расположения.

3.5.4. Теорема CAP

Теорема CAP рассматривает распределенную вычислительную систему, состоящую из множества вычислительных ресурсов (CPU, память, хранилища), соединенных коммуникационной сетью и совместно решающих задачи. Теорема (иногда ее называют гипотезой) указывает три характеристики распределенных систем, которые нужны всегда.

- Согласованность означает, что система одинаково реагирует на запрос, независимо от принявшего его узла (или не отвечает совсем).
- Доступность означает, что система всегда отвечает на запрос (хотя запросы могут быть несогласованными или некорректными).
- Устойчивость к разделению обеспечивает функционирование системы даже при отказе узлов или коммуникационной сети.

В 2000 году Eric Brewer [CAPBR] предположил, что распределенная система может удовлетворять любым двум из этих требований, но не всем трем. Это предположение позднее было подтверждено Gilbert и Lynch [CAPGL], а сейчас его обычно называют теоремой CAP [CAPFN].

Пересылку пакетов через сеть корректно будет считать вычислительной задачей. Одна из основных абстракций SDN заключается в том, что все элементы сети считаются вычислительными ресурсами, выполняющими простую расчетную задачу просмотра полей входящего пакета и принятия решения о его пересылке. Поскольку задача пересылки пакета со входа сети на ее выход обычно реализуется большим числом пересылающих элементов, сеть пересылающих элементов является распределенной вычислительной системой. Следовательно, теорема CAP применима к пересылке пакетов.

Если в контексте теоремы CAP считать устойчивость к разделению первостепенно важной, традиционные операции уровня управления обычно будут локальными и быстрыми (доступность), тогда как операции уровня администрирования обычно централизованы (согласованность) и могут быть достаточно медленными.

Теорема CAP обеспечивает также понимание архитектуры SDN. Например, централизованный контроллер SDN действует как согласованная глобальная база данных, а конкретные механизмы SDN обеспечивают согласованную обработку входящих в сеть пакетов всеми коммутаторами SDN. Вопрос устойчивости к потере связности с контроллером не решается в базовой модели SDN. Когда контроллер недоступен коммутатору SDN, поток становится недоступным, пока соединение не будет восстановлено. Предложено было использовать множество контроллеров SDN, расположенных в разных местах (например, указав в коммутаторе SDN список контроллеров), что может повысить уровень устойчивости к разделению, но за счет потери абсолютной согласованности. Panda и др. [CAPFN] выполнили первое исследование применимости теоремы CAP к SDN.

3.6. Уровень абстракции сетевых служб

Уровень NSAL обеспечивает доступ к службам уровня управления, администрирования и операций со стороны других служб и приложений. Отметим, что термин SAL слишком перегружен и часто применяется в разном контексте от разработки систем до ориентированной на услуги архитектуры, поэтому к нему явно добавлены сети (N - Network), чтобы подчеркнуть связь с рисунком 1, а в разделе 4 он отображен на известные решения SDN.

Интерфейсы служб могут принимать разные формы в зависимости от конкретных требований. Примеры сервисных интерфейсов включают RESTful API, открытые протоколы типа NETCONF, межпроцессные коммуникации, интерфейсы

CORBA [CORBA] и т. п. Двумя основными подходами являются интерфейсы RESTful и интерфейсы вызова удаленных процедур (RPC¹). Оба подхода используют архитектуру «клиент-сервер» и XML или JSON для передачи сообщений, но имеют слегка различающиеся характеристики.

Характеристики интерфейсов RESTful, разработанных в соответствии с парадигмой передачи репрезентативного состояния [REST], перечислены ниже.

- Идентификация ресурсов позволяет указывать отдельные ресурсы, например, с URI.
- Манипуляции с ресурсами через представления в форматах типа JSON, XML или HTML.
- Описывающие себя сообщения, каждое из которых содержит информацию, достаточную для его обработки.
- Гиперсреда в качестве конечного автомата приложения. Клиенту не нужно заранее знать как взаимодействовать с приложением, поскольку интерфейс API динамически предоставляется сервером.

Вызовы удаленных процедур (RPC) [RFC5531], типа XML-RPC и т. п., имеют следующие характеристики:

- отдельные процедуры указываются идентификаторами;
- клиент должен знать имя процедуры и связанные с ней параметры.

3.7. Уровень приложений

Приложения и службы, использующие услуги уровня управления и/или администрирования, образуют уровень приложений.

Кроме того, службы уровня приложений могут предоставлять свои услуги другим службам и приложениям того же уровня через сервисные интерфейсы.

Примеры приложений включают определение топологии сети, предоставление сети, резервирование пути и т. п.

4. Представление модели SDN

Мы считаем, что в интерфейс «южной границы» SDN следует включать как CPSI, так и MPSI.

Контроллеры SDN типа [NOX] и [Beacon] представляют собой набор приложений и служб уровня управления, реализующих CPSI ([NOX] и [Beacon] используют OpenFlow) и обеспечивающих интерфейс «северной границы» для приложений. Интерфейс северной границы SDN для контроллеров реализован в уровне абстракции сетевых служб (NSAL) рисунка 1.

Указанная выше модель может применяться для описания всех известных технологий с поддержкой SDN, как будет показано ниже.

4.1. ForCES

Схема IETF ForCES² [RFC3746] включает одну модель и два протокола. ForCES отделяет уровень пересылки от уровня управления открытым интерфейсом в виде протокола ForCES [RFC5810], который работает с объектами уровня пересылки, используя модель ForCES [RFC5812].

Модель ForCES [RFC5812] основана на том, что элементы сети состоят из множества логически разделенных объектов, работающих совместно для обеспечения заданной функциональности (например, маршрутизации или коммутации IP), а для внешнего наблюдателя они представляются цельными элементами сети.

ForCES моделирует уровень пересылки с использованием логических функциональных блоков (LFB³), которые, будучи связанными в граф, образуют элемент пересылки (FE⁴). LFB описываются в формате XML на основе схемы XML.

Определения LFB включают базовые и «заказные» (custom-defined) типы данных, определения метаданных, входные и выходные порты, операционные параметры и компоненты, а также определения возможностей и событий.

Модель ForCES может использоваться для определения LFB с требуемым уровнем детализации, независимо от физической или виртуальной природы устройств.

Протокол ForCES не зависит от модели и может служить для мониторинга, настройки и управления любым элементом, смоделированным в ForCES. Протокол включает простые команды Set, Get и Del. Протокол ForCES разрабатывался для обеспечения высокой пропускной способности и быстрых обновлений.

Применительно к рисунку 1, модель ForCES [RFC5812] подходит для DAL, а также уровней операций и пересылки, использующих LFB. Протокол ForCES [RFC5810] был разработан с расчетом на CPSI и подходит для этого интерфейса, но может применяться и для MPSI.

4.2. NETCONF/YANG

NETCONF⁵ [RFC6241] является протоколом сетевого управления IETF [RFC6632] и обеспечивает механизмы для создания, изменения и удаления конфигурации сетевых устройств.

Операции протокола NETCONF реализованы в форме вызова удаленных процедур (RPC). Протокол NETCONF использует представление данных в формате XML как для конфигурации, так и для протокольных сообщений. Недавние исследования, в частности, [ESNet] и [PENet] показали, что NETCONF работает лучше SNMP [RFC3411].

В дополнение к этому был разработан язык моделирования данных YANG [RFC6020] для задания моделей данных и протокольных операций NETCONF. Язык моделирования данных YANG используется для моделирования данных конфигурации и состояния в протоколе NETCONF, а также вызовах удаленных процедур и уведомлениях NETCONF.

Модели YANG организуют иерархию данных в форме дерева, где каждый узел имеет значение или набор дочерних узлов. Кроме того, YANG структурирует модели данных в модули и submodule, что обеспечивает их многократное применение и расширяемость. YANG также включает набор базовых типов данных и позволяет определять свои типы.

YANG обеспечивает возможность определять NETCONF RPC, которые позволяют включать в протокол расширяемые наборы команд. Для определения RPC имена операций, входные и выходные параметры задаются с использованием операторов определения данных YANG.

¹Remote Procedure Call.

²Forwarding and Control Element Separation - разделение элементов пересылки и управления.

³Logical Functional Block.

⁴Forwarding Element.

⁵Network Configuration Protocol — протокол настройки конфигурации сети.

Применительно к рисунку 1, модель YANG [RFC6020] подходит для задания DAL на уровнях операций и пересылки. NETCONF [RFC6241] подходит для MPSI. NETCONF является протоколом управления [RFC6632], который не был (изначально) рассчитан на быстрое обновление CP и может не соответствовать требованиям CPSI.

4.3. OpenFlow

Модель OpenFlow изначально разрабатывалась в Stanford University, а сейчас активно развивается в качестве стандарта [OpenFlow] консорциумом ONF¹. Исходной целью было предоставление исследователям возможности применять экспериментальные протоколы в действующей сети [OF08]. Модель OpenFlow претерпела множество изменений и очевидно появление новых версий. Приведенное описание соответствует версии 1.4 [OpenFlow]. Вкратце, OpenFlow определяет протокол, с помощью которого логически централизованный контроллер может управлять коммутатором OpenFlow. Каждый коммутатор, соответствующий OpenFlow, поддерживает одну или множество таблиц потоков, которые служат для поиска (lookup) пакетов. При поиске и пересылке пакетов применяются различные операции. Таблица групп и канал OpenFlow к внешнему контроллеру также входят в спецификацию коммутатора.

Применительно к рисунку 1, спецификации коммутатора OpenFlow [OpenFlow] определяют DAL для уровня пересылки, а также для CPSI. Протокол OF-CONFIG [OF-CONFIG], основанный на модели YANG [RFC6020], обеспечивает DAL для уровней пересылки и операций коммутатора OpenFlow, а также задает NETCONF [RFC6241] в качестве MPSI. OF-CONFIG перекрывается с OpenFlow DAL, но при использовании NETCONF [RFC6241] в качестве транспортного протокола, ему присущи ограничения, описанные в предыдущем параграфе.

4.4. Интерфейс с системой маршрутизации

Интерфейс с системой маршрутизации (I2RS) обеспечивает стандартный путь взаимодействия с системой маршрутизации в реальном масштабе времени с управлением по событиям через набор определяемых протоколами интерфейсов управления и администрирования. По сути, одной из основных целей I2RS является обеспечение возможности программирования RIB², что позволит по новому организовать работу сетей.

Интерфейс I2RS изначально не был предназначен для создания новых интерфейсов и лишь усиливал или расширял существующие, а также определял информационные модели для системы маршрутизации. Например, в недавнем обсуждении проблем I2RS [I2RSProb] рассматривались ранее определенные IETF протоколы типа ForCES [RFC5810], NETCONF [RFC6241] и SNMP [RFC3417]. Для определения информационных моделей и моделей данных рабочая группа I2RS выбрала язык моделирования YANG [RFC6020].

В настоящее время группа I2RS занята разработкой информационной модели [I2RSInfo] применительно к уровню абстракции сетевых служб (NSAL) для агента I2RS.

Применительно к рисунку 1, архитектура I2RS [I2RSArch] охватывает уровни управления и приложений и использует любые доступные CPSI и DAL, будь то ForCES [RFC5810], OpenFlow [OpenFlow] или другой интерфейс. Агент I2RS является службой уровня управления. Все службы и приложения «поверх» этого агента относятся к уровням управления, администрирования или приложений. В документах I2RS административный доступ к агенту может быть обеспечен протоколами управления типа SNMP и NETCONF. Протокол I2RS можно также отобразить на сервисный интерфейс для обеспечения доступа службам и приложениям, не относящимся к уровню управления.

4.5. SNMP

Простой протокол сетевого управления SNMP³ является протоколом управления, стандартизованным IETF, и в настоящее время имеется третья версия протокола (SNMPv3) [RFC3417] [RFC3412] [RFC3414]. Протокол включает набор стандартов управления сетью, в том числе протокол уровня приложений, схему баз данных и набор объектов данных. SNMP представляет данные администрирования (управляемые объекты) в форме переменных управляемой системы, которые описывают конфигурацию этой системы. Переменные могут считываться и устанавливаться управляющими приложениями.

SNMP использует расширяемую модель для описания данных, определяемого базами MIB⁴, которые описывают структуру данных управления для подсистемы устройства. В MIB используется иерархическое пространство имен, содержащее идентификаторы объектов (OID⁵). Каждый идентификатор OID указывает переменную, которая может быть прочитана и установлена по протоколу SNMP. В базах MIB используются обозначения, определенные структурой управляющей информации (Structure of Management Information) версии 2 [RFC2578].

Ранний пример использования SNMP в контексте SDN рассмотрен в работе [Peregrine].

Применительно к рисунку 1, базы SNMP MIB могут служить для описания DAL уровней пересылки и операций. Подобно YANG, базы SNMP MIB могут описывать DAL для уровня пересылки. Протокол SNMP, подобно NETCONF, предназначен для MPSI.

4.6. PCEP

Архитектура PCE⁶ [RFC4655] определяет сущность, способную рассчитывать пути для одной или множества служб. В качестве PCE может служить узел сети, станция управления сетью или специализированная расчетная платформа, знающие о ресурсах и способные учесть множество ограничений для разных вариантов расчета путей и технологий коммутации. Коммуникационный протокол PCE (PCEP⁷) [RFC5440] применяется для обмена информацией между клиентами PCC⁸ и PCE или между множеством устройств PCE.

Архитектура PCE представляет сеть с разделением расчета путей для служб, сквозной сигнализации и реальной пересылки пакетов. Определения интерактивного (online) или автономного (offline) расчета пути зависят от доступности PCE из сети и узлов системы управления сетью (NMS⁹), а также от типа оптимизации запроса, который может существенно влиять на время отклика PCE клиентам PCC.

¹Open Networking Foundation.

²Routing Information Base - база данных маршрутизации.

³Simple Network Management Protocol.

⁴Management Information Base - база данных управления.

⁵Object identifier.

⁶Path Computation Element - элемент расчета пути.

⁷PCE Communication Protocol.

⁸Path Computation Client.

⁹Network Management System.

Механизм обмена сообщениями PCEP упрощает задание конечных точек расчета (адреса отправителя и получателя), целевых функций (запрошенный алгоритм и критерии оптимизации) и связанных с ними ограничений типа параметров трафика (например, запрашиваемой полосы), возможностей коммутации и типа кодирования.

Применительно к рисунку 1, PCE является службой уровня управления, предоставляющей услуги приложениям этого уровня. PCEP может служить в качестве интерфейса «восток-запад» между PCE, которые могут выступать в качестве элементов управления доменом (службы и приложения). Рабочая группа PCE определяет расширения [PCEActive], которые позволяют активному PCE работать с использованием PCEP, MPLS или LSP¹, что делает его применимым в качестве CPSI для коммутаторов MPLS и GMPLS.

4.7. BFD

Обнаружение двухсторонней пересылки BFD² [RFC5880] - это стандартизованный IETF сетевой протокол, разработанный для детектирования отказов на пути между двумя элементами пересылки, включая физические интерфейсы, субинтерфейсы, каналы данных и (для расширения возможностей) сами устройства пересылки, с потенциально очень малой задержкой. BFD может обеспечивать детектирование с малыми издержками для всех типов путей между системами, включая прямые физические соединения, виртуальные устройства (каналы), MPLS LSP, пути с множеством интервалов маршрутизации и односторонние соединения, для которых имеется также обратный путь. Протокол часто реализуется в неких компонентах машины пересылки в системе, когда пересылка и управление разделены.

Применительно к рисунку 1, агент BFD может быть реализован в виде службы уровня управления, которая будет использовать CPSI в направлении уровня пересылки для отправки и приема пакетов BFD. Однако агенты BFD обычно реализуются в форме приложения на устройстве и используют уровень пересылки для отправки и получения пакетов BFD, обновляя соответствующим образом ресурсы уровня управления. Службы и приложения уровней администрирования и управления, которые отслеживают изменение ресурсов или подписаны на уведомления о таких изменениях, будут узнавать об изменениях через соответствующие интерфейсы и выполнять требуемые действия.

5. Заключение

Этот документ был создан после тщательного и детального анализа литературы, RFC и документов, выпущенных другими органами стандартизации. Документ был представлен на рассмотрение широкого сообщества SDN и мы надеемся, что он сможет послужить удобным инструментом для сетевых исследователей, инженеров и практиков в ближайшие годы.

Мы завершаем документ кратким обзором терминологии многоуровневой архитектуры SDN. В общем случае мы рассматриваем элемент сети как некий набор ресурсов. Каждый элемент сети включает уровень пересылки (FP), отвечающий за обработку пакетов в пути данных, и уровень операций (OP), отвечающий за управление рабочим состоянием устройства. Ресурсы элемента сети представляются абстракцией DAL, управляются и администрируются службами или приложениями, которые относятся к уровню управления или администрирования. Уровень управления (CP) отвечает за принятие решений о пересылке пакетов. Уровень администрирования (MP) отвечает за мониторинг, настройку и администрирование сетевых устройств. Интерфейсы служб представляются абстракцией NSAL, через которую другие сетевые службы или приложения могут использовать их. Введенная этим документом систематика (таксономия) определяет разные уровни SDN, уровни абстракции и интерфейсы для того, чтобы прояснить терминологию SDN и задать общепринятые определения для сообщества SDN, безотносительно к конкретным реализациям.

6. Вопросы безопасности

Этот документ не предлагает новой архитектуры или протокола, поэтому он не оказывает какого-либо влияния на безопасность Internet. При этом безопасность имеет первостепенное значение для сетей, поэтому ее следует принимать во внимание при разработке сетевой архитектуры и развертывании сетей. Безопасность SDN рассматривается в литературе, например, в [SDNSecurity], [SDNSecServ] и [SDNSecOF]. Вопросы безопасности, относящиеся к конкретным интерфейсам (например, ForCES, I2RS, SNMP или NETCONF), рассмотрены в соответствующих документах, а также в [RFC7149].

7. Литература

- [A4D05] Greenberg, A., Hjalmtysson, G., Maltz, D., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and H. Zhang, "[A Clean Slate 4D Approach to Network Control and Management](#)", ACM SIGCOMM Computer Communication Review, Volume 35, Issue 5, pp. 41-54, 2005.
- [ALIEN] Parniewicz, D., Corin, R., Ogirodowczyk, L., Fard, M., Matias, J., Gerola, M., Fuentes, V., Toseef, U., Zaalouk, A., Belter, B., Jacob, E., and K. Pentikousis, "Design and Implementation of an OpenFlow Hardware Abstraction Layer", In Proceedings of the ACM SIGCOMM Workshop on Distributed Cloud Computing (DCC), Chicago, Illinois, USA, pp. 71-76, doi 10.1145/2627566.2627577, August 2014.
- [Beacon] Erickson, D., "[The Beacon OpenFlow Controller](#)", In Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking, pp. 13-18, 2013.
- [CAPBR] Brewer, E., "[Towards Robust Distributed Systems](#)", In Proceedings of the Symposium on Principles of Distributed Computing (PODC), 2000.
- [CAPFN] Panda, A., Scott, C., Ghodsi, A., Koponen, T., and S. Shenker, "[CAP for Networks](#)", In Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking, pp. 91-96, 2013.
- [CAPGL] Gilbert, S. and N. Lynch, "[Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services](#)", ACM SIGACT News, Volume 33, Issue 2, pp. 51-59, 2002.
- [CORBA] Object Management Group, "CORBA Version 3.3", November 2012, <<http://www.omg.org/spec/CORBA/3.3/>>.
- [DIOPR] Denazis, S., Miki, K., Vicente, J., and A. Campbell, "[Designing Interfaces for Open Programmable Routers](#)", In "Active Networks", Springer Berlin Heidelberg, pp. 13-24, 1999.

¹GMPLS Label Switched Path - путь с коммутацией по меткам GMPLS.

²Bidirectional Forwarding Detection.

- [ESNet] Yu, J. and I. Al Ajameh, "An Empirical Study of the NETCONF Protocol", Sixth International Conference on Networking and Services, pp. 253-258, 2010.
- [FCAPS] ITU, "Management Framework For Open Systems Interconnection (OSI) For CCITT Applications", ITU Recommendation X.700, September 1992, <<http://www.itu.int/rec/T-REC-X.700-199209-I/en>>.
- [I2RSArch] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", Work in Progress¹, draft-ietf-i2rs-architecture-07, December 2014.
- [I2RSInfo] Bahadur, N., Folkes, R., Kini, S., and J. Medved, "Routing Information Base Info Model", Work in Progress, draft-ietf-i2rs-rib-info-model-04², December 2014.
- [I2RSProb] Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", Work in Progress³, draft-ietf-i2rs-problem-statement-05, January 2015.
- [ITUATM] ITU, "B-ISDN ATM Layer Specification", ITU Recommendation I.361, 1990, <<http://www.itu.int/rec/T-REC-I.361-199902-I/en>>.
- [ITUSG11] ITU, "ITU-T Study Group 11: Protocols and test specifications", <<http://www.itu.int/en/ITU-T/studygroups/2013-2016/11/Pages/default.aspx>>.
- [ITUSG13] ITU, "ITU-T Study Group 13: Future networks including cloud computing, mobile and next-generation networks", <<http://www.itu.int/en/ITU-T/studygroups/2013-2016/13/Pages/default.aspx>>.
- [ITUSS7] ITU, "Introduction to CCITT Signalling System No. 7", ITU Recommendation Q.700, 1993, <<http://www.itu.int/rec/T-REC-Q.700-199303-I/e>>.
- [ITUY3300] ITU, "Framework of software-defined networking", ITU Recommendation Y.3300, June 2014, <<http://www.itu.int/rec/T-REC-Y.3300-201406-I/en>>.
- [KANDOO] Yeganeh, S. and Y. Ganjali, "[Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications](#)", In Proceedings of the first ACM SIGCOMM workshop on Hot Topics in Software Defined Networks, pp. 19-24, 2012.
- [NFVArch] ETSI, "Network Functions Virtualisation (NFV): Architectural Framework", ETSI GS NFV 002, October 2013, <http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf>.
- [NOX] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and S. Shenker, "[NOX: Towards an Operating System for Networks](#)", ACM SIGCOMM Computer Communication Review, Volume 38, Issue 3, pp. 105-110, July 2008.
- [NV09] Chowdhury, N. and R. Boutaba, "Network Virtualization: State of the Art and Research Challenges", Communications Magazine, IEEE, Volume 47, Issue 7, pp. 20-26, 2009.
- [OF-CONFIG] Open Networking Foundation, "OpenFlow Management and Configuration Protocol (OF-Config 1.1.1)", March 2013, <<https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1-1-1.pdf>>.
- [OF08] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and J. Turner, "[OpenFlow: Enabling Innovation in Campus Networks](#)", ACM SIGCOMM Computer Communication Review, Volume 38, Issue 2, pp. 69-74, 2008.
- [ONFArch] Open Networking Foundation, "SDN Architecture, Version 1", June 2014, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf>.
- [OpenFlow] Open Networking Foundation, "The OpenFlow Switch Specification, Version 1.4.0", October 2013, <<https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>>.
- [P1520] Biswas, J., Lazar, A., Huard, J., Lim, K., Mahjoub, S., Pau, L., Suzuki, M., Torstensson, S., Wang, W., and S. Weinstein, "The IEEE P1520 standards initiative for programmable network interfaces", IEEE Communications Magazine, Volume 36, Issue 10, pp. 64-70, 1998.
- [PCEActive] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", Work in Progress⁴, draft-ietf-pce-stateful-pce-10, October 2014.
- [PENet] Hedstrom, B., Watwe, A., and S. Sakthidharan, "[Protocol Efficiencies of NETCONF versus SNMP for Configuration Management Functions](#)", Master's thesis, University of Colorado, 2011.
- [PNSurvey99] Campbell, A., De Meer, H., Kounavis, M., Miki, K., Vicente, J., and D. Villela, "[A Survey of Programmable Networks](#)", ACM SIGCOMM Computer Communication Review, Volume 29, Issue 2, pp. 7-23, September 1992.
- [Peregrine] Chiueh, D., Tu, C., Wang, Y., Wang, P., Li, K., and Y. Huang, "Peregrine: An All-Layer-2 Container Computer Network", In Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing, pp. 686-693, 2012.
- [PiNA] Day, J., "Patterns in Network Architecture: A Return to Fundamentals", Prentice Hall, ISBN 0132252422, 2008.
- [RCP] Caesar, M., Caldwell, D., Feamster, N., Rexford, J., Shaikh, A., and J. van der Merwe, "[Design and Implementation of a Routing Control Platform](#)", In Proceedings of the 2nd conference on Symposium on

¹Работа завершена и опубликована в RFC 7921. Прим. перев.

²Работа завершена и опубликована в RFC 8430. Прим. перев.

³Работа завершена и опубликована в RFC 7920. Прим. перев.

⁴Работа завершена и опубликована в RFC 8231. Прим. перев.

Networked Systems Design & Implementation Volume 2, pp. 15-28, 2005.

- [REST] Fielding, Roy, "Chapter 5: Representational State Transfer (REST)", in Dissertation "[Architectural Styles and the Design of Network-based Software Architectures](#)", 2000.
- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, [RFC 826](#), November 1982, <<http://www.rfc-editor.org/info/rfc826>>.
- [RFC1953] Newman, P., Edwards, W., Hinden, R., Hoffman, E., Ching Liaw, F., Lyon, T., and G. Minshall, "Ipsilon Flow Management Protocol Specification for IPv4 Version 1.0", RFC 1953, May 1996, <<http://www.rfc-editor.org/info/rfc1953>>.
- [RFC2297] Newman, P., Edwards, W., Hinden, R., Hoffman, E., Liaw, F., Lyon, T., and G. Minshall, "Ipsilon's General Switch Management Protocol Specification Version 2.0", RFC 2297, March 1998, <<http://www.rfc-editor.org/info/rfc2297>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999, <<http://www.rfc-editor.org/info/rfc2578>>.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002, <<http://www.rfc-editor.org/info/rfc3411>>.
- [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>>.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002, <<http://www.rfc-editor.org/info/rfc3414>>.
- [RFC3417] Presuhn, R., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002, <<http://www.rfc-editor.org/info/rfc3417>>.
- [RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002, <<http://www.rfc-editor.org/info/rfc3418>>.
- [RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003, <<http://www.rfc-editor.org/info/rfc3535>>.
- [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>>.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), January 2006, <<http://www.rfc-editor.org/info/rfc4271>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", [RFC 5424](#), March 2009, <<http://www.rfc-editor.org/info/rfc5424>>.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, May 2009, <<http://www.rfc-editor.org/info/rfc5531>>.
- [RFC5743] Falk, A., "Definition of an Internet Research Task Force (IRTF) Document Stream", RFC 5743, December 2009, <<http://www.rfc-editor.org/info/rfc5743>>.
- [RFC5810] Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., 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>>.
- [RFC5812] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", [RFC 5812](#), March 2010, <<http://www.rfc-editor.org/info/rfc5812>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6632] Ersue, M. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, June 2012, <<http://www.rfc-editor.org/info/rfc6632>>.
- [RFC7011] Claise, B., Trammell, B., 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>>.
- [RFC7047] Pfaff, B. and B. Davie, "The Open vSwitch Database Management Protocol", [RFC 7047](#), December 2013, <<http://www.rfc-editor.org/info/rfc7047>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, March 2014, <<http://www.rfc-editor.org/info/rfc7149>>.

- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", [RFC 7276](#), June 2014, <<http://www.rfc-editor.org/info/rfc7276>>.
- [RINA] Day, J., Matta, I., and K. Mattar, "[Networking is IPC: A Guiding Principle to a Better Internet](#)", In Proceedings of the 2008 ACM CoNEXT Conference, Article No. 67, 2008.
- [RouteFlow] Nascimento, M., Rothenberg, C., Salvador, M., Correa, C., de Lucena, S., and M. Magalhaes, "[Virtual Routers as a Service: The RouteFlow Approach Leveraging Software-Defined Networks](#)", In Proceedings of the 6th International Conference on Future Internet Technologies, pp. 34-37, 2011.
- [SDNACS] Kreutz, D., Ramos, F., Verissimo, P., Rothenberg, C., Azodolmolky, S., and S. Uhlig, "[Software-Defined Networking: A Comprehensive Survey](#)", Networking and Internet Architecture (cs.NI), arXiv:1406.0440, 2014.
- [SDNHistory] Feamster, N., Rexford, J., and E. Zegura, "[The Road to SDN: An Intellectual History of Programmable Networks](#)", ACM Queue, Volume 11, Issue 12, 2013.
- [SDNNFV] Haleplidis, E., Hadi Salim, J., Denazis, S., and O. Koufopavlou, "Towards a Network Abstraction Model for SDN", Journal of Network and Systems Management: Special Issue on Management of Software Defined Networks, pp. 1-19, 2014.
- [SDNSecOF] Kloti, R., Kotronis, V., and P. Smith, "[OpenFlow: A Security Analysis](#)", 21st IEEE International Conference on Network Protocols (ICNP) pp. 1-6, October 2013.
- [SDNSecServ] Scott-Hayward, S., O'Callaghan, G., and S. Sezer, "[SDN Security: A Survey](#)", In IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1-7, 2013.
- [SDNSecurity] Kreutz, D., Ramos, F., and P. Verissimo, "[Towards Secure and Dependable Software-Defined Networks](#)", In Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking, pp. 55-60, 2013.
- [SDNSurvey] Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K., and T. Turletti, "[A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks](#)", IEEE Communications Surveys and Tutorials, DOI:10.1109/SURV.2014.012214.00180, 2014.
- [SLTSDN] Jarraya, Y., Madi, T., and M. Debbabi, "[A Survey and a Layered Taxonomy of Software-Defined Networking](#)", IEEE Communications Surveys and Tutorials, Volume 16, Issue 4, pp. 1955-1980, 2014.
- [SoftRouter] Lakshman, T., Nandagopal, T., Ramjee, R., Sabnani, K., and T. Woo, "[The SoftRouter Architecture](#)", In Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networking, 2004.
- [Tempest] Rooney, S., van der Merwe, J., Crosby, S., and I. Leslie, "[The Tempest: A Framework for Safe, Resource Assured, Programmable Networks](#)", Communications Magazine, IEEE, Volume 36, Issue 10, pp. 42-53, 1998.

Благодарности

Авторы благодарны Salvatore Loreto и Sudhir Modali за их вклад в первоначальное обсуждение в рамках почтовой конференции SDNRG, а также связанные с документом замечания, которые помогли улучшить форму документа.

Кроме того, мы выражаем признательность (в алфавитном порядке) Shivleela Arlimatti, Roland Bless, Scott Brim, Alan Clark, Luis Miguel Contreras Murillo, Tim Copley, Linda Dunbar, Ken Gray, Deniz Gurkan, Dave Hood, Georgios Karagiannis, Bhunip Khasnabish, Sriganesh Kini, Ramki Krishnan, Dirk Kutscher, Diego Lopez, Scott Mansfield, Pedro Martinez-Julia, David E. Mcdysan, Erik Nordmark, Carlos Pignataro, Robert Raszuk, Bless Roland, Francisco Javier Ros Munoz, Dimitri Staessens, Yaakov Stein, Eve Varma, Stuart Venters, Russ White и Lee Young за критические замечания о и дискуссии в рамках IETF 88, IETF 89 и IETF 90, а также в почтовой конференции SDNRG, которые были приняты во внимание при пересмотре этого документа.

Мы благодарим (в алфавитном порядке) Spencer Dawkins и Eliot Lear за рецензирование в IRSG, которое внесло уточнения в документ.

Наконец, мы благодарим Nobo Akiya за рецензирование раздела BFD, Julien Meuric за рецензирование раздела PCE, а также Adrian Farrel и Benoit Claise за рецензирование документа в IESG.

Поддержка Kostas Pentikousis осуществлялась [ALIEN], исследовательским проектом, частично финансируемым Европейской Комиссией по программе Seventh Framework (грант 317880). Выраженное здесь мнение принадлежит лишь автору. Европейская Комиссия не несет никакой ответственности за любое использование содержащейся в этом документе информации.

Участники работы

Авторы благодарны перечисленным ниже (в алфавитном порядке) людям, внесшим свой вклад в создание этого документа. Все они предоставили тексты, ссылки и комментарии, которые позволили сделать документ более полным.

- Daniel King предоставил текст, относящийся к PCEP.
- Scott Mansfield предоставил информацию по текущим работам ITU в сфере SDN.
- Yaakov Stein предоставил текст, относящийся к теореме CAP и связанную с SDO информацию.
- Russ White внес предложения по тексту определений управления, администрирования и приложений.

Адреса авторов

Evangelos Haleplidis (редактор)

University of Patras

Department of Electrical and Computer Engineering

Patras 26500

Greece

EMail: ehalep@ece.upatras.gr

Kostas Pentikousis (редактор)

EICT GmbH
Torgauer Strasse 12-15
10829 Berlin
Germany
EMail: k.pentikousis@eict.de

Spyros Denazis

University of Patras
Department of Electrical and Computer Engineering
Patras 26500
Greece
EMail: sdena@upatras.gr

Jamal Hadi Salim

Mojatatu Networks
Suite 400, 303 Moodie Dr.
Ottawa, Ontario K2H 9R4
Canada
EMail: hadi@mojatatu.com

David Meyer

Brocade
EMail: dmm@1-4-5.net

Odysseas Koufopavlou

University of Patras
Department of Electrical and Computer Engineering
Patras 26500
Greece
EMail: odysseas@ece.upatras.gr

Перевод на русский язык

Николай Малых
nmalykh@gmail.com