OPENFLOW

OpenFlow – Տվյալների մշակման պրոցեսի ղեկավարման պրոտոկոլ է, որոնք փոխանցվում են ցանցում ռուտեռներով և սվիչներով, իրականացնելով SDN տեխնոլոգիան(տվյալների փոխանցման ցանց, որում ցանցի կառավարման մակարդակը առանձնացված է տվյալների փոխանցման սարքերից և իրականացվում է ծրագրային ապահովման մեջ` ցանցի վիրտուալիզացման ձևերից մեկը)։

SDN-ի հիմնական սկզբունքներն են՝ տվյալների ղեկավարման և փոխանցման պրոցեսների բաժանումը, ցանցի կենտրոնացված ղեկավարումը օգտագործելով միասնական ծրագրային գործիքներ, ֆիզիկական ցանցային ռեսուրսների վիրտուալիզացում(ցանցի վիրտուալիզացումը տեխնոլոգիա է, որը թույլ է տալիս ստեղծել վիրտուալ ցանցեր, սարքեր և ցանցային գործառույթներ ՝ առանց ծրագրային ապահովման ֆիզիկական ենթակառուցվածքի հետ կապված լինելու):

Կախված ցանցերի մասշտաբից և օգտագործվող միջավայրից` առանձնանում են հատուկ տեխնոլոգիաներ. SD-WAN (ծրագրակազմով սահմանված լայն ցանց), SD-LAN (տեղական), SDMN (բջջային):

Գոյություն ունի SDN-ի 3 մակարդակ՝

1. Ենթակառուցվածքի մակարդակը, որի վրա գործում են ցանցի սվիչները և տվյալների փոխանցման ուղիները.
2. Ղեկավարման մակարդակ – ծրագրային գործիքների ամբողջություն, որոնք ֆիզիկապես առանձնացված են ենթակառուցվածքային մակարդակից `ապահովելով ենթակառուցվածքային մակարդակում սարքերի կառավարման մեխանիզմների իրականացումը։
3. Ցանցային ծրագրերի մակարդակ- SDN հավելվածների ամբողջություն, որոնք SDN կոնտրոլլերի հետ փոխգործակցում են ծրագրային ապահովման պրոտոկոլի (API) միջոցով՝ կիրառական մակարդակում ցանցային ենթակառուցվածքը հավաքելու, վերլուծելու, տեղակայելու և կառավարման համար:

OpenFlow արձանագրությունը օգտագործվոււմ է կենտրոնական սարքի՝ կոնտրոլլերի օգնությամբ ցանցային սարքերի՝ ռոուտեր , սվիչ կենտրոնական ղեկավարման համար։Այս ղեկավարումը լրացնում կամ փոխարինում է ցանցային սարքում ներդրված ծրագիրը,որն իրականացնում է երթուղու կառուցում,կոմուտացիայի քարտեզի ստեղծում։

Կոնտրոլլերը օգտագործվում է սվիչի հոսքի աղյուսակները կառավարելու համար, որոնց հիման վրա որոշում է կայացվում ստացված փաթեթը փոխանցել սվիչի կոնկրետ պորտի: Այսպիսով, ցանցում ստեղծվում են ուղղակի ցանցային կապեր տվյալների փոխանցման նվազագույն ուշացումներով և անհրաժեշտ պարամետրերով:

**Ճարտարապետությունը**

Տվյալների ուղին բաղկացած է հոսքերի աղյուսակից և աղյուսակի յուրաքանչյուր գրառմանը նշանակված գործողություններից։ Աղյուսակներն իրենք կարող են վերաբերել ինչպես Ethernet-ին (կամ հղման շերտի այլ արձանագրություններին), այնպես էլ բարձր մակարդակի արձանագրություններին (IP, TCP): Գործողությունների ճշգրիտ ցուցակը կարող է տարբեր լինել,բայց սրաքն հիմնականներն են՝

**Forwarding- փաթեթները, ֆրեյմները փոխանցում է անհրաժեշտ պորտին,** **ուղարկում տվյալների մի կտոր կոնտրոլլերին անվտանգ երթուղով՝ հետագա հետազոտությունների համար, տվյալների կտորների՝ ֆրագմենտների գցում(drop):**

**Սարքերը որոնք համատեղում են openflow-ն և փաթեթների սովորական մշակումը սարքի միկրոծրագրերի գործիքներով, ավելացնում է գործողության 4-րդ տիպը՝տվյալնեի մշակումը սովորական գործիքներով։ Սարքը ,որը ներառում է այս 4 գործողությունները կոչվում է type0 սարքավորում։**

OpenFlow սարքը բաղկացած է առնվազն երեք բաղադրիչներից.

1. Հոսքի աղյուսակներ(անգլ. flow table);
2. Անվտանգ երթուղի(անգլ. secure channel), որն օգտագործվում է արտաքին «խելացի» սարքի (հսկիչի) միջոցով սվիչը կառավարելու համար;
3. Կառավարման համար օգտագործվող OpenFlow պրոտոկոլի աջակցություն: Այս արձանագրության օգտագործումը հնարավորություն է տալիս խուսափել վերահսկվող սարքի համար ծրագիր գրելու անհրաժեշտությունից.

Հոսքերի աղյուսակումյուրաքանչյուր գրառում ունի 3 դաշտ՝

1. PDU ([**Protocol Data Unit**](https://en.wikipedia.org/wiki/Protocol_Data_Unit)-տվյալների ֆրագմենտներ) վերնագիր, որը թույլ է տալիս որոշել PDU- ի համապատասխանությունը հոսքին,
2. գործողություն
3. դաշտ վիճակագրությամբ (հոսքին համապատասխան բայթերի և PDU– ների քանակ, հոսքին համապատասխանող վերջին PDU– ի տարանցման ժամանակ):

Վերնագիրը կարող է բաղկացած լինել տարբեր մակարդակների բազմաթիվ դաշտերից (օրինակ ՝ աղբյուրի և նպատակակետի MAC հասցեները, IP փաթեթի վերնագրի դաշտերը, TCP հատվածի վերնագրի դաշտերը): Հարկ է նշել, որ արձանագրության ներկա տարբերակը չի աջակցում ստուգելու, ​​օրինակ, TCP հատվածի վերնագրի մեջ SYN դրոշը: Յուրաքանչյուր դաշտ կարող է ունենալ հատուկ արժեք (աստղանիշ), ինչը նշանակում է, որ այն համապատասխանում է PDU- ի համապատասխան դաշտի ցանկացած արժեքին:

Type1 սարքերը, որոնք պետք է ապահովեն ցանցային հասցեների թարգմանությունը,կլասսների և առաջնահերթությունների աջակցությունը, պլանավորված են , սակայն դեռ դրանց օգտագորման պահանջները դեռ ճշգրտված չեն։

Կոնտրոլլերները ապահովում են հոսքերի աղյուսակի լրացում` փաթեթներ ստացում սարքից անվտանգ երթուղով: Դրանք կարող են իրականացվել որպես պարզ ալգորիթմ, որը հիշեցնում է փաթեթները բաժանող սվիչի վարքագիծը տրամաբանական ցանցերում (VLAN), կամ նրանք կարող են իրականացնել բարդ դինամիկ տրամաբանություն, որն ազդում է փաթեթների անցման վրա` ելնելով արտաքին պատճառներից (մուտքի իրավունքներ, սերվերի բեռ, ծառայության առաջնահերթություններ և այլն)):

|  |  |  |
| --- | --- | --- |
| **OpenFlow** | **SDN** | **Traditional networks** |
| **OpenFlow-ն պրոտոկոլ է, որը կարգավորում է ցանցի սվիչները` օգտագործելով API-ի նման գործընթացներ:** | **SDN-ը զարգացող ճարտարապետություն է, որը դինամիկ, կառավարելի, ծախսարդյունավետ և հարմարվող է՝ ինչն էլ նրան դարձնում է իդեալական այսօրվա բարձր թողունակության ծրագրերի դինամիկ բնույթի համար: Այս ճարտարապետությունն առանձնացնում է ցանցի կառավարման և տվյալների փոխանցման գործառույթները՝ թույլ տալով, որ ցանցի կառավարումն ուղղակիորեն ծրագրավորվի, և հիմքում ընկած ենթակառուցվածքը վերացվի դիմումների և ցանցային ծառայությունների համար:** **OpenFlow արձանագրությունը հիմնարար նշանակություն ունի SDN լուծումներ ստեղծելու համար:** | **SDN- ի և ավանդական ցանցերի միջև ամենաուշագրավ տարբերությունն այն է, որ SDN-ն հիմնված է ծրագրային ապահովման վրա, մինչդեռ ավանդական ցանցերը `սովորաբար ապարատային: Քանի որ SDN- ն հիմնված է ծրագրային ապահովման վրա, այն ավելի ճկուն է ՝ ինչն էլ թույլ տալիս գտվողներին ավելի մեծ վերահսկողություն և հեշտությամբ վերահսկել ռեսուրսները գործնականում ամբողջ կառավարման հարթությունում:** |

**Information About OpenFlow**

OpenFlow is a specification from the Open NetworkingFoundation (ONF) that defines a flow-based forwarding infrastructure (L2-L4 Ethernet switch model) and a standardized application programmatic interface (protocol definition) to learn capabilities, add and remove flow control entries and request statistics. OpenFlow allows a controller to direct the forwarding functions of a switch through a secure channel. Cisco ONE Platform Kit provides the ability to host Cisco internal or external third party applications on or adjacent to Cisco’s networking infrastructure, and enables programmatic access to networking services in a controlled and consistent manner. When hosting applications on Cisco routers or switches, the applications will run within a virtual-machine or container.

**OpenFlow Limitations**

The Cisco Nexus 5500 and Cisco Nexus 6000 switches do not support the OpenFlow action to rewrite the layer-2 destination MAC address. Therefore, the XNC controller use cases such as Topology Independent Configuring OpenFlow 1 Forwarding and Latency Optimized Forwarding may not be work correctly on the Cisco Nexus 5500 and Cisco Nexus 6000 switches.

**Supported Interface**

Types The following is a list of supported interface types:

• Regular Layer 2 physical ports (switchport)

• Layer 2 port channels

**Unsupported Interface**

Types The following is a list of unsupported interface types:

• Layer 3 ports (no switchport)

• Fabric extender ports

• Virtual Port-Channel (VPC) ports

• Layer 3 Port-Channel

**Supported Interface Modes**

The following is a list of supported interface modes:

• Access port

• Trunk port

**Supported Match Fields**

The following are lists of supported match fields:

• Layer 2 header

• Ethertype

• VLAN ID

• VLAN priority (PCP)

• Source MAC address

• Destination MAC address

• Layer 3 header

• Source IP address

• Destination IP address

• Layer 4 protocol

• Differentiated services Code Point (DSCP)

• Layer 4 header • Source port

• Destination port

• Ingress Interface

**Supported Actions**

The following is a list of supported actions:

• Redirect the packet to one output port

• Redirect the packet to multiple output ports

• Set the VLAN tag (vlan rewrite) on egress

• Strip the VLAN tag on egress

• Divert the datapath packet to the OpenFlow controller

• Drop the packet

**Scale Flow Numbers**

• The Cisco Nexus device supports a maximum of 65535 flowsin total. The device supports a combination of up to 3500 ACL-table flows and 62K MAC-table flows.

• The Cisco Nexus device supports up to 64 flows when the action is punt-to-controller. **Pipeline Support**

OpenFlow policies can be applied to the ACL-table and the MAC-table. OpenFlow relates tables by means of the ‘pipeline’ concept. The Cisco Nexus device supports two pipelines, 201 and 202. You can toggle the pipeline between 201and 202 by entering the pipeline id command in the openflow-agent logical switch configuration.

• Pipeline 201

• All the flows are added to the ACL-table. For example, ACL TCAM.

• ACL-table flows with the action as redirect or drop gets installed in the IFACL region of the ACL-TCAM. Configuring OpenFlow

• ACL-table flows with the action punt-to-controller are installed in theSUPregion of the ACL-TCAM.

• Pipeline 202

• Flows can be added to both the ACL-table(ACL TCAM) and the MAC-table(STM table).

• Flows with only L2-dest-mac and VLAN as the match criteria are installed in the MAC-table. The remaining flows are installed in ACL-table

• Supported actions for the MAC-table are redirect-to-port and drop

. • The MAC-table supports a higher scale number than the ACL-table.

**Prerequisites for OpenFlow**

The OpenFlow agent requires the Cisco Nexus device to be configured with OpenFlow specific commands in order to support topology discovery and the installation of flows. The Cisco Nexus device worksin a hybrid mode so that the default commands from the startup-config file are executed upon boot up. This might create an undesirable effect and therefore must be changed.

**Note** If you change or negate these required commands, it can lead to unpredictable system behavior.

**VLAN Creation**

The following command is used to create the necessary VLANs in an OpenFlow-controller switch. This command creates the OpenFlow specific VLANs in the VLAN database.

**vlan x[-y]**

Even with the hybrid-Ships-In-Night mode of operation, we recommend that you segregate the VLANs among the OpenFlow-controlled ports and the regular ports. You should take caution in ensuring that the VLANs are not shared among the OpenFlow and non-OpenFlow ports in order to prevent traffic leaks.

**Interface Level Configurations**

To make the interfaces connected to other switches receive spanned traffic, the interface is connected to the analyzer and configured to support OpenFlow. The interface ethernet command changes the parser to the interface submode. Before entering the mode openflow command which enables OpenFlow support on the interface, the following commands are required:

• switchport mode trunk

• switchport trunk allowed vlan x-y

In order for the strip-vlan functionality to work on the Cisco Nexus device, the trunk port must be configured with the native VLAN. Cisco One controllers can perform topology discovery of OpenFlow enabled ports. To allow topology discovery on trunk ports, the native VLANs must be configured on trunk ports

switchport trunk native vlan z

Configuring OpenFlow 4 Configuring OpenFlow Prerequisites for OpenFlow When an interface is added to the OpenFlow logical switch, the following commands are applied to the interface implicitly:

• mode openflow

• spanning-tree bpdufilter enable

• no lldp transmit

**Template Based TCAM Carving for OpenFlow**

The Cisco Nexus device supports template-based TCAM carving. To configure OpenFlow on the device, you must make a number of changes to the TCAM carving regions using the template based TCAM carving commands.

To support higher scale numbers for OpenFlow policies, the IFACL-region of the TCAM must be recarved accordingly. To apply TCAM carving for a maximum flow scale, enter the following commands

switch(config)# hardware profile tcam resource template openflow

switch(config-tcam-templ)# vacl 64

switch(config-tcam-templ)# ifacl 3520

switch(config-tcam-templ)# qos 128

switch(config-tcam-templ)# rbacl 64

switch(config-tcam-templ)# span 64

switch(config)# hardware profile tcam resource service-template openflow

Enter the following command to verify the TCAM carving: show hardwareprofile tcam resource template tmplt-name

**Note** Configuring TCAM carving requires that the Cisco Nexus device be reloaded.

**Setting Up an OpenFlow Virtual Service**

The virtual service manager allows you to enable the OpenFlow agent application to run as a virtual service on a container. To setup a virtual service for OpenFlow you must perform the following tasks:

• Download the application OVA package to your system.

• Install the OVA package for a named virtual service. For example:

switch#virtual-service install name openflow-agent package file-url

• Configure and activate the virtual service. For example:

switch(config)#virtual-service openflow-agent

switch(config-virt-serv)#activate To upgrade a software package installed on a virtual service you use the virtual-service upgrade name application-name package file-url command.

**Note** An active virtual service can not be updated.

**Enabling OpenFlow**

OpenFlow capability is enabled by entering thehardwareprofile openflow command to allocate the hardware resources required for the OpenFlow agent. Following a switch reload, the hardware profile command is used to configure ACL Feature Manager (AFM) and Forwarding Manager (FWM) modules for OpenFlow functionality.

Procedure

Command or Action

**Step 1** switch# configure terminal

**Purpose** Enters global configuration mode.

**Step 2** switch(config)# hardware profile openflow

**Purpose** Allocates the hardware resources required for the OpenFlow agent.

**Step 3** switch(config)# copy running-config startup-config

**Purpose** Saves the change persistently through reboots and restarts b copying the running configuration to the startup configuration․

**Step 4** switch#reload

**Purpose** Reloads the operating system on the switch

**Configuring the OpenFlow Switch**

You must enable OpenFlow on the switch, for the configuration to take effect.

Produce

Command or Action

Purpose

Step 1 switch# configure terminal

Enters global configuration mode.

Step 2 openflow

Example: switch(config)# openflow

Enters OpenFlow configuration mode.

Step 3 switch *switch-number*

Example:

switch(config-ofa)# switch 1

Specifies the OpenFlow logical switch and enters OpenFlow switch configuration mode

Step 4 pipeline {201|202}

Specifies the pipeline mode.

Example:

switch(config-ofa-switch)# pipeline 201

OpenFlow policies can be applied to the ACL-table and the MAC-table. Cisco Nexus devices support two pipelines, 201 and 202. This command allows you to switch between the supported pipeline modes.

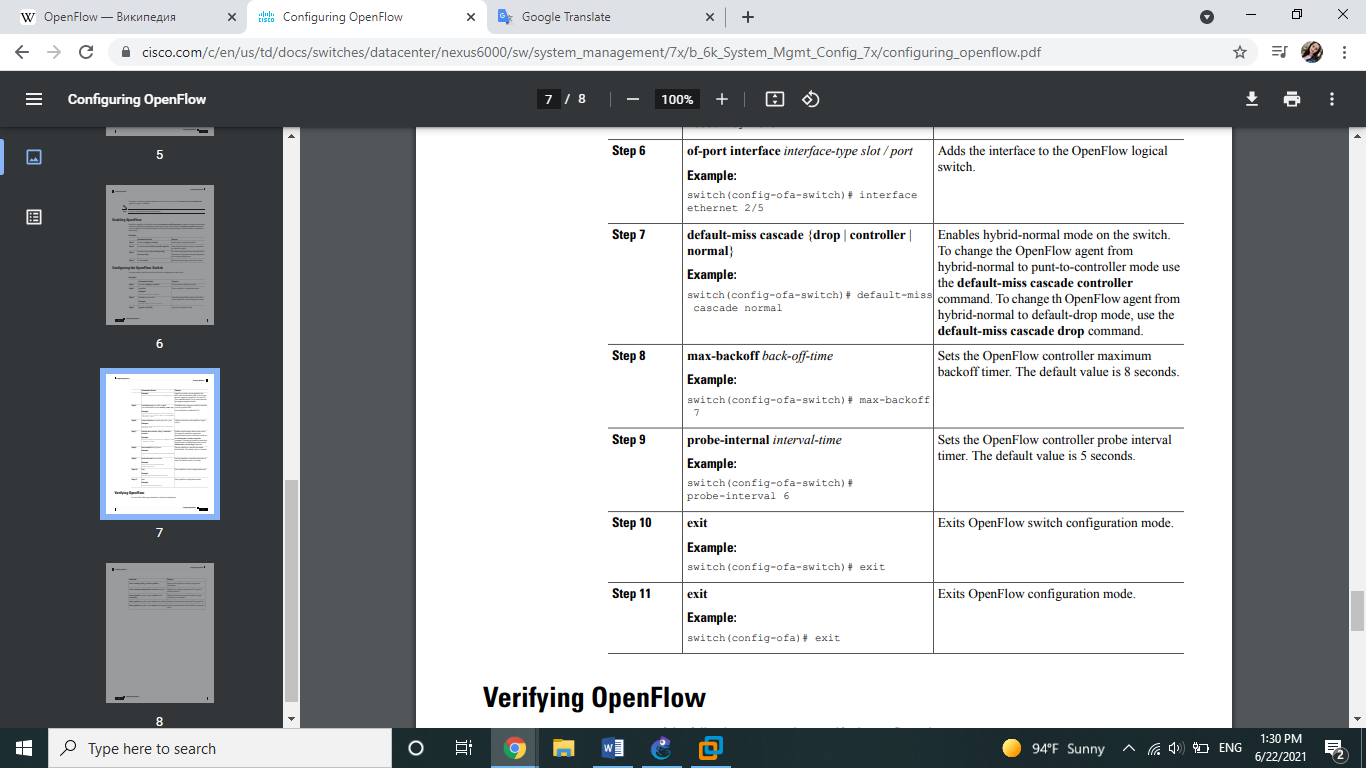
Step 5 controller ipv4 ipv4-address port port-numbervrf vrf-name security {none | tls}

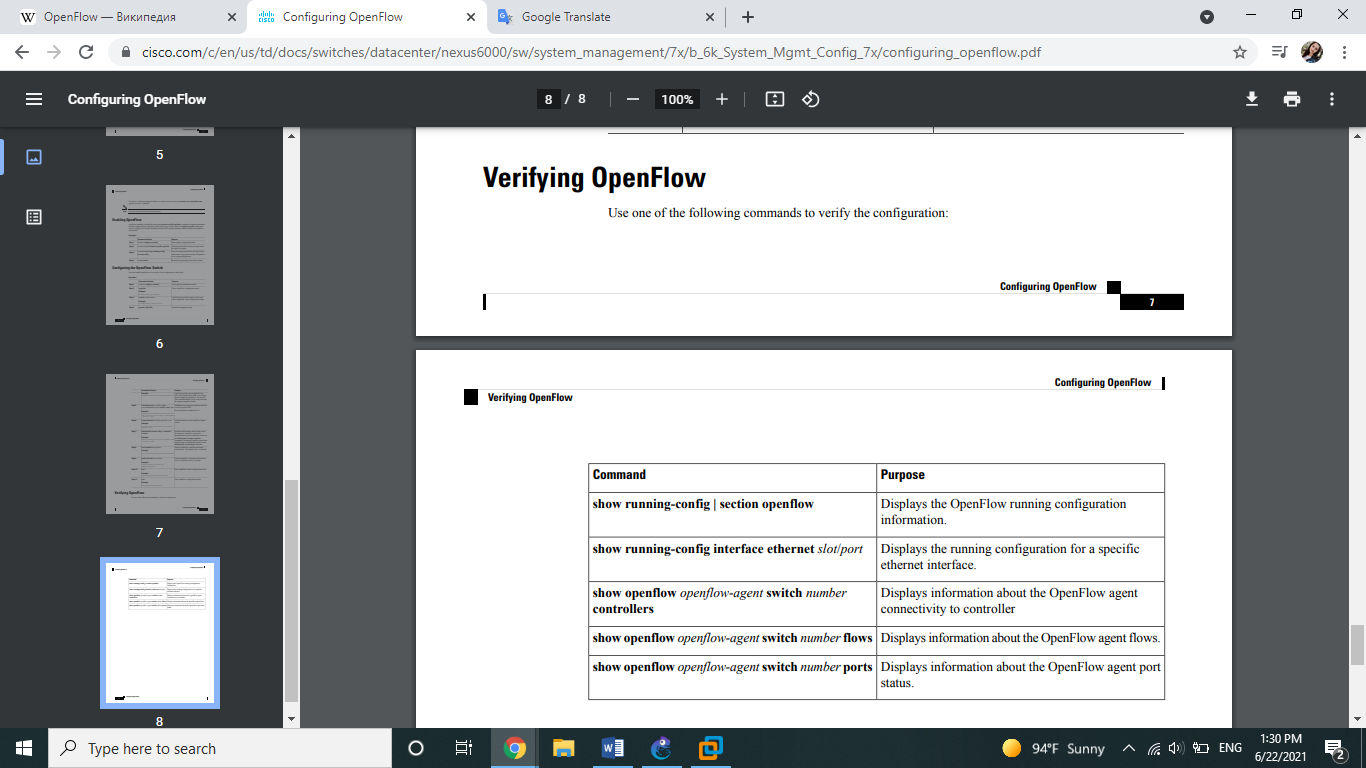
Example:

switch(config-ofa-switch)# controller ipv4 192.0.2.10 port 6653 vrf mnagement security none

Purpose Establishes the connection with the controller over the specified VRF.

You can disable or enable the TLS





<https://www.arccn.ru/media/1132/>

## В ОЖИДАНИИ ПЕРЕМЕН

Стремительный рост объемов трафика и изменение его структуры, необходимость поддержки растущей армии мобильных пользователей, формирования высокопроизводительных кластеров для обработки Больших Данных и хорошо масштабируемых виртуализированных сред для предоставления облачных сервисов — все это серьезно изменило требования к сетевым средам. И все чаще сеть превращается в ограничивающий фактор развития вычислительной инфраструктуры.

Главная проблема: традиционные сети слишком статичны и потому не соответствуют динамике, свойственной современному бизнесу, в отличие от серверов — чем последние обязаны технологиям виртуализации. Сегодня приложения распределены между множеством виртуальных машин, которые интенсивно обмениваются данными (что ведет к росту трафика запад — восток, который начинает доминировать над традиционным для архитектур клиент-сервер трафиком север — юг). Для оптимизации загрузки серверов виртуальные машины часто мигрируют, что меняет точки «привязки» трафика. Традиционные схемы адресации, логического деления сетей и способы назначения правил обработки трафика в таких динамичных средах становятся неэффективны.

Например, при запуске новой виртуальной машины, реконфигурирование списков контроля доступа (ACL) на всех сетевых устройствах в крупной сети может занять несколько дней. Причина — ориентация имеющихся инструментов управления на работу с отдельными устройствами: в лучшем случае автоматизация назначения параметров распространяется на группу устройств, в которую входят представители одного модельного ряда конкретного производителя. В результате администраторам приходится тратить массу времени на то, чтобы вручную перенастроить правила обработки трафика на каждом отдельном устройстве. Такие же проблемы возникают с переконфигурацией механизмов QoS при добавлении в мультисервисную сеть нового приложения, например видеосвязи. Недопустимо много времени в больших сетях занимают процедуры по изменению параметров защиты, что не позволяет оперативно реагировать на возникающие угрозы.

«Лоскутная» природа имеющихся средств управления значительно усложняет масштабирование современных сетей. Дополнительные сложности в части масштабирования создают и ограничения по числу логических групп. Например, как известно, стандартная технология VLAN обеспечивает поддержку только 4096 виртуальных локальных сетей, а при развертывании облачных сервисов IaaS коммерческим ЦОД уже сегодня требуется гораздо большее число виртуальных сетей. Представьте, что услуги IaaS предоставляются сотне предприятий, у каждого из которых по сотне VLAN, — уже в этом случае число логических сетей составляет 10 тыс.

Но еще большую озабоченность у ИТ-директоров вызывает неопределенность относительно поддержки будущих приложений и сервисов. Смогут ли инсталлируемые сегодня сетевые устройства обеспечить такую поддержку? В какой мере будущее развитие сети — а значит, и бизнес компании — будет привязано к продуктовой стратегии выбранного производителя коммутаторов? Архитектура традиционного сетевого оборудования делает эту «привязку» очень прочной. SDN обещает существенно ослабить, а то и полностью ликвидировать зависимость заказчиков от технологий конкретного вендора.

## РАЗДЕЛЯЙ И УПРАВЛЯЙ

|  |
| --- |
| **[Рисунок 1. Архитектура типичного коммутатора или маршрутизатора.](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_LAN/2012-12/12_12/13134618/ZHurnal_setevyh_reshenij_LAN_1_(3842).png)** |
| ***Рисунок 1.****Архитектура типичного коммутатора или маршрутизатора.* |

[**Главная идея SDN**](http://www.osp.ru/lan/2014/06/13041880/) заключается в отделении функций передачи трафика от функций управления (включая контроль как самого трафика, так и осуществляющих его передачу устройств). В традиционных коммутаторах и маршрутизаторах эти процессы неотделимы друг от друга и реализованы в одной «коробке»: специальные микросхемы обеспечивают пересылку пакетов с одного порта на другой, а вышележащее ПО определяет правила такой пересылки, выполняет необходимый анализ пакетов, производит изменение содержащейся в них служебной информации и т. д. (см. Рисунок 1). Для определения маршрута передачи или недопущения зацикливания трафика устройства, конечно, «общаются между собой», для чего разработано множество протоколов, таких как OSPF, BGP и Spanning Tree, но при этом каждое функционирует достаточно автономно.

Согласно концепции SDN, вся логика управления выносится в так называемые контроллеры, которые способны отслеживать работу всей сети (см. Рисунок 2). Нельзя сказать, что это революционно новая идея: связисты помнят, что подобную логику в свое время предполагалось реализовать при модернизации телефонных сетей, результатом чего стало появление «интеллектуальных сетей», а затем и коммутаторов класса Softswitch.

|  |
| --- |
| **[Рисунок 2. Архитектура SDN.](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_LAN/2012-12/12_12/13134618/ZHurnal_setevyh_reshenij_LAN_2_(4697).png)** |
| ***Рисунок 2.****Архитектура SDN.* |

Согласно замыслу разработчиков, SDN позволит программировать сеть как единое целое, а администраторам не придется заниматься отдельными устройствами. Главным становится контроллер: он все видит, все знает и раздает сетевым устройствам инструкции по обработке трафика. Самим устройствам больше не надо разбираться в сотнях замысловатых протоколов — достаточно следовать инструкциям контроллера, а значит, они могут быть простыми и дешевыми.

Реализация концепции SDN на практике позволит предприятиям и операторам связи получить вендоронезависимый контроль над всей сетью из единого места, что значительно упростит ее эксплуатацию. Что не менее важно, конфигурирование сети сильно упростится и администраторам не придется вводить сотни строчек кода отдельно для разных коммутаторов или маршрутизаторов. Характеристики сети можно будет оперативно изменять в режиме реального времени, соответственно, сроки внедрения новых приложений и сервисов значительно сократятся.

Основным элементом концепции SDN является протокол OpenFlow, который обеспечивает взаимодействие контроллера с сетевыми устройствами (см. Рисунок 2). На «северной» стороне контроллер предоставляет программные интерфейсы (API), наличие которых позволяет владельцу сети или сторонним разработчикам создавать приложения для управления сетью. Такие приложения могут выполнять самые разные функции в интересах бизнес-задач (например, контролировать доступ, управлять пропускной способностью и т. п.), причем их разработчикам не надо знать детали функционирования конкретных сетевых устройств. Благодаря контроллеру, вся сеть, состоящая из множества разнотипных устройств разных производителей, предстает для приложения как один логический коммутатор.

## НЕМНОГО ИСТОРИИ

Концепция архитектуры SDN и протокола OpenFlow зародилась в Стэнфордском университете, исследовательской группе которого потребовалось создать тестовую среду для экспериментов с новыми сетевыми протоколами. Строить отдельную сеть было дорого, поэтому решили задействовать имеющуюся университетскую сеть, в которой с помощью прообраза SDN были выделены ресурсы для испытаний.

Интерес к SDN со стороны неуниверситетских кругов первыми проявили крупные поставщики интернетсервисов, которым требовались высокопроизводительные инфраструктуры для организации взаимодействия между десятками и даже сотнями серверов в гигантских ЦОД. Традиционная трехуровневая архитектура (доступ — агрегация — ядро) и необходимость производить множество действий при обработке трафика в каждом узле представлялись для них избыточными и чрезмерными. Именно шесть крупных поставщиков услуг — компании Deutsche Telekom, Facebook, Google, Microsoft,Verizon и Yahoo — весной 2011 года сформировали организацию Open Networking Foundation (ONF) с целью развития технологий SDN в целом и протокола OpenFlow в частности. Сегодня членами ONF являются практически все основные поставщики сетевого оборудования, включая Alcatel-Lucent, Brocade, Ciena, Cisco, Dell, Ericsson, Extreme Networks, HP, Huawei, IBM, Infinera, Intel, Juniper Networks, Mellanox, Netgear, Nokia Siemens Networks, ZTE, а также лидеры рынка систем виртуализации VMware и Citrix.

## КАК РАБОТАЕТ[SDN](http://www.osp.ru/lan/2014/06/13041880/)

Как и следует из названия, протокол OpenFlow при идентификации трафика оперирует понятием «потока». Ключевым элементом коммутатора, поддерживающего этот протокол, является таблица потоков (Flow Table). Группа столбцов в левой части таблицы формирует поля соответствия, где указаны характеристики потоков: это могут быть различные параметры, включая МАС- и IP-адреса отправителя и получателя, идентификатор VLAN, номера протокольных портов TCP и UDP, а также другая информация (см. Рисунок 3). Эти данные с помощью протокола OpenFlow записывает в таблицу коммутатора контроллер, он же определяет приоритет разных потоков: чем выше приоритет, тем выше соответствующая запись в таблице потоков.

|  |
| --- |
| **[Рисунок 3. Типичная таблица потоков в сетевом устройстве, поддерживающем OpenFlow.](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_LAN/2012-12/12_12/13134618/ZHurnal_setevyh_reshenij_LAN_3_(9104).png)** |
| ***Рисунок 3.****Типичная таблица потоков в сетевом устройстве, поддерживающем OpenFlow.* |

Входящие пакеты проверяются на соответствие указанным в таблице параметрам. Если соответствие выявлено, к пакетам применяется действие, которое указано в следующем столбце таблицы. Типичным действием является пересылка пакета на один или несколько выходных портов. Кроме того, коммутатор может изменить содержимое служебных полей пакета, сбросить его, направить для анализа контроллеру и т. д. В случае если совпадение не найдено, пакет сбрасывается или направляется контроллеру, который определит, как следует обрабатывать данный поток, и добавит соответствующую запись в таблицу. Статистика по проходящему трафику — число пакетов, байтов и пр. — помещается в соответствующие поля (на Рисунке 3 они обозначены как Count).

Используя протокол OpenFlow, контроллер добавляет, модифицирует и удаляет записи в таблице потоков. Кроме того, он может запрашивать у коммутатора его характеристики и собранную статистику, конфигурировать коммутатор и его отдельные порты.

Хотя в названиях организации ONF и ее основного протокола присутствует слово «открытый», как вы понимаете, это совсем не синоним слову «бесплатный». Члены ONF могут использовать протокол OpenFlow в своих продуктах, но за само членство взимается ежегодная плата в 30 тыс. долларов.

## ТЕКУЩАЯ СИТУАЦИЯ НА РЫНКЕ

[**Интерес к SDN**](http://www.osp.ru/lan/2014/06/13041880/) со стороны крупных поставщиков интернет-сервисов, облачных услуг и владельцев мегаЦОД понятен: новые технологии позволят им решать свои задачи более эффективно и, главное, за меньшие деньги. А что же производители сетевого оборудования?

Наиболее рьяно за осваивание нового «поля» взялись стартапы. Один из пионеров в области SDN компания Nicira разработала контроллер и платформу виртуализации сетей (Network Virtualization Platform, NPV), а летом 2012 года она была куплена Vmware. По-видимому, следующий претендент на поглощение кем-нибудь из крупных игроков — компания Big Switch Networks, выпустившая контроллер Floodlight и средство тестирования OFTest. Упомянем еще один стартап — компанию Pica8, которая предлагает серию недорогих OpenFlowсовместимых коммутаторов.

Известные компании тоже не остаются в стороне. Так, например, NEC разработала контроллер и коммутаторы с поддержкой OpenFlow. IBM выпустила линию OpenFlow-совместимых коммутаторов, а также сотрудничает с NEC с целью обеспечения их совместимости с ее контроллером. НР добавила поддержку OpenFlow в большое число моделей коммутаторов и разрабатывает контроллер SDN. Многие ведущие производители коммутаторов и маршрутизаторов, включая Brocade, Cisco, Extreme Networks и Juniper Networks, либо уже объявили о выпуске OpenFlow-совместимых коммутаторов/маршрутизаторов, либо планируют сделать это в ближайшее время.

Но ни один из ведущих поставщиков сетевого оборудования не объявил SDN главной целью своего технологического развития. Этого можно было ожидать, ведь открываемая SDN возможность использования простых и дешевых коммутаторов в какой-то степени подрывает бизнес этих компаний. Они многие годы совершенствовали функционал своих коммутаторов и маршрутизаторов, именно он — их главное «ноу-хау», основной источник добавленной стоимости, а значит, и прибыли. А теперь что же, все «коту под хвост»? Отдать весь функционал в какой-то контроллер, тем более стороннего производителя?!

Интересно, зачем вообще эти компании ввязались в историю с SDN? Думаю, одна из причин заключается в том, что производители не хотят терять крупных заказчиков, которые «прониклись» идеями SDN. Еще раз взгляните на список основателей ONF, добавьте к ним других операторов связи и ЦОД, которые уже присоединились к этой организации (Colt, Deutsche Telekom, France Telecom Orange, Goldman Sachs, KDDI, Korea Telecom, Level 3 Communications, NTT Communications, SK Telecom, Swisscom, Telecom Italia), учтите заинтересованность, высказанную множеством других игроков, например «Ростелекомом», — и станет понятно, что вендорам просто нельзя дистанцироваться от столь представительной компании.

Другая, возможно, гораздо более важная причина — у сетевых вендоров просто нет альтернативы. Компании Broadcom и Marvell, ведущие производители микросхем для коммутаторов, — тоже члены ONF — уже объявили о поддержке OpenFlow. А собрать простенький коммутатор, работающий по протоколу OpenFlow, из имеющихся наборов микросхем может даже не слишком искушенная в вопросах сетевых технологий компания. Что, собственно, и продемонстрировала Google: сеть, связывающая ее центры обработки данных, построена на базе коммутаторов SDN собственной разработки. Если по этому пути пойдут и другие крупные игроки, бизнес Cisco и других традиционных производителей коммутаторов может оказаться под угрозой.

## ВИРТУАЛЬНЫЙ SDN

Разделение «плоскостей» передачи и управления можно реализовать вообще не затрагивая имеющуюся физическую сеть — задействуя виртуальные коммутаторы наподобие Cisco Nexus 1000v, VMware DVS, IBM 5000v или даже Open vSwitch с открытым исходным кодом (см. Рисунок 4). Программирование таких коммутаторов с помощью контроллера позволяет создать виртуальную сеть SDN поверх имеющейся физической инфраструктуры. Некоторые эксперты рассматривают этот подход как альтернативу развиваемому ONF, но на самом деле, поскольку описываемая схема не исключает возможности использования стандартного протокола OpenFlow, противопоставлять ее решениям ONF не стоит.

|  |
| --- |
| **[Рисунок 4. Реализация принципов SDN с использованием виртуальных коммутаторов vSwitch.](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_LAN/2012-12/12_12/13134618/ZHurnal_setevyh_reshenij_LAN_4_(7140).png)** |
| ***Рисунок 4.****Реализация принципов SDN с использованием виртуальных коммутаторов vSwitch.* |

Это далеко не умозрительная конструкция: подобное решение развернула компания «Крок» в своем ЦОД, где для предоставления облачных сервисов используются программные коммутаторы Open vSwitch, контроллеры компании Nicira и протокол OpenFlow. Все это функционирует поверх высокопроизводительной физической сети, построенной на базе оборудования Extreme Networks. Главная задача, которую успешно решила «Крок», — формирование более 1 млн изолированных сетей на втором уровне (L2). С помощью обычных VLAN такая задача не решается в принципе.

Если в такой сети обычные коммутаторы также будут поддерживать OpenFlow, то к виртуальной сети можно будет подключить и физические серверы. Управление такими коммутаторами тоже можно будет передать контроллеру, если это не войдет в противоречие с принципом разделения физической и виртуальных сетей. Этот пример показывает, что в модели SDN конкретная реализация коммутатора — будь то физическое устройство или программа на гипервизоре — не имеет принципиального значения, главное, чтобы он мог получать и исполнять инструкции контроллера.

## ДРУГОЙ ПОДХОД

В рамках своей стратегии SDN ряд производителей заявили об открытии функционала операционных систем своих устройств через API. До сих пор настройка сетевых устройств производилась преимущественно через командную строку или Webинтерфейс. Но эти инструменты ограничены той оболочкой программирования, которую предлагает производитель. При наличии API можно использовать более широкий набор инструментов программирования и создавать приложения не только для настройки сетевого устройства, но и для программирования сетевой среды в целом. По сути, этот подход является альтернативой SDN, при этом он обеспечивает доступ к более широкому набору функций сетевых устройств, что потенциально позволяет реализовать больше возможностей, чем заполнение таблицы потоков.

В частности, наличие доступа через API непосредственно к функционалу сетевых устройств позволяет системам типа VMware vCenter программировать сеть — например, задавать настройки VLAN в рамках общих задач по развертыванию и обслуживанию виртуальных машин. Для многих производителей коммутаторов интеграция с системой vCenter чрезвычайно важна, поскольку она упрощает и автоматизирует процедуры конфигурирования сетей для сред виртуализации VMware.

Одним из примеров описываемого подхода служит анонсированная летом 2012 года компанией Cisco инициатива Open Network Environment (ONE), в рамках которой она представила несколько библиотек API для своих основных операционных систем: IOS, IOS XR и NX-OS. Используя эти API, заказчики могут разрабатывать свои собственные приложения, которые устанавливаются непосредственно на устройствах или запускаются удаленно (более подробно о стратегии Cisco применительно к SDN см. интервью с Полом Пересом, вице-президентом Cisco и главным директором по технологиям подразделения по разработке решений для ЦОД). Подобные возможности предлагают и другие производители, скажем Juniper (интерфейс XML API) или F5 Networks (API-интерфейс iControl).

Предоставление производителями подобных API означает, что для реализации централизованного программирования и управления совсем необязательно использовать протокол OpenFlow. Кроме того, такие инициативы снимают с повестки дня тезис о том, что коммутаторы могут быть очень простыми и дешевыми.

## ТРУДНОСТИ ПЕРЕВОДА

Преимущества, которые дает SDN, очевидны, причем не только для сетей в центрах обработки данных, но и для других типов сетевых инфраструктур. Централизованное управление мультивендорной средой, значительное упрощение обслуживания и модернизации, сокращение времени на обновление программных кодов коммутаторов/ маршрутизаторов и внедрение новых сервисов — все перечисленное важно как для корпоративных сетей, так и для инфраструктур операторов связи. Однако это не повод разом отказываться от преимуществ развиваемого десятилетиями традиционного подхода, когда каждое сетевое устройство наделяется «интеллектом», достаточным для автономного функционирования.

Полагаю, элементы SDN будут внедряться постепенно, при этом только компании уровня Google смогут позволить себе перейти на коммутаторы, полностью управляемые из центрального контроллера. Скорее всего, часть функционала, например по конфигурированию и заданию политики безопасности, будут выносить в контроллеры, при этом сами коммутаторы останутся достаточно «разумными», по крайней мере для того чтобы участвовать в работе протоколов определения маршрутов или балансировки трафика.

Что же касается русскоязычного термина, то SDN — это, скорее, «централизованно программируемая сеть». И надо поскорее прийти к консенсусу относительно такого термина. Напомню, что ISDN неформально расшифровывался как «I Still Don’t Need it», то есть «мне этого пока не нужно», причем именно такое отношение и предопределило судьбу этой технологии в России. Надеюсь, очень разумные идеи SDN ждет лучшая доля.