

# Conceitos e Tecnologias para Dispositivos Conectados (C115)

Prof. Samuel Baraldi Mafra



## Histórico:

- O termo SDN (Software-Defined Networking) foi originalmente cunhado para representar as ideias e o trabalho relacionado ao projeto OpenFlow, em Stanford.

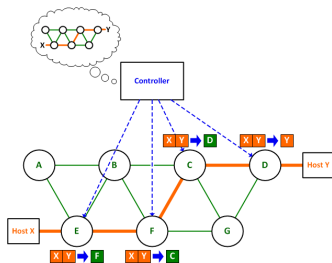


## Origens do SDN

- 1. Necessidade de tratar o encaminhamento de pacotes de forma personalizada. Ex: rotear um grupo de pacotes por um caminho, outro grupo por outro caminho. Como testar novos protocolos de roteamento na prática? As funções de roteamento eram gravadas em hardware.
- 2. A rede é um enorme sistema distribuído. A maioria dos protocolos são distribuídos. Isso não é eficiente para gerenciar redes com equipamentos próximos e em grande quantidade: data centers.
- 3. Link cair ou mudança ocorrer: tempo para convergência da rede elevado. Nesse período, as tabelas de encaminhamento são atualizadas.

## Fluxos (Flows)

- Um fluxo descreve um conjunto de pacotes transferidos de um ponto de extremidade da rede (ou conjunto de pontos de extremidade) para outro ponto de extremidade (ou conjunto de pontos de extremidade).
- Os pontos de extremidade podem ser definidos como pares de endereço IP-porta TCP / UDP, pontos de extremidade de VLAN, pontos de extremidade de túnel da camada três e porta de entrada, entre outras coisas.
- Um conjunto de regras descreve as ações que o dispositivo deve realizar para todos os pacotes pertencentes a esse fluxo.

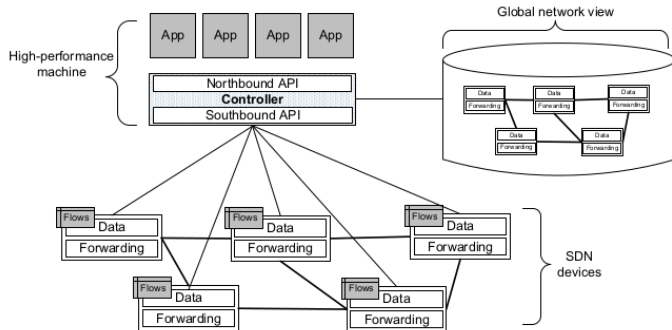


- Um fluxo é unidirecional, ou seja, os pacotes fluindo entre os mesmos dois pontos de extremidade na direção oposta podem constituir um fluxo separado;
- Os fluxos são representados em um dispositivo como uma entrada de fluxo.

Três abstrações básicas formam a base para SDN:

- Abstrações de estado distribuído: fornece ao programador de rede uma visão de rede global que protege o programador das realidades de uma rede que, na verdade, é composta de muitas máquinas, cada uma com seu próprio estado, colaborando para resolver problemas de toda a rede;
- Encaminhamento: permite que o programador possa especificar os comportamentos de encaminhamento necessários sem nenhum conhecimento do hardware específico do fornecedor;
- Configuração: o programador deve ser capaz de expressar os objetivos desejados da rede geral sem se perder nos detalhes de como a rede física implementará esses objetivos.

# Arquitetura de rede SDN



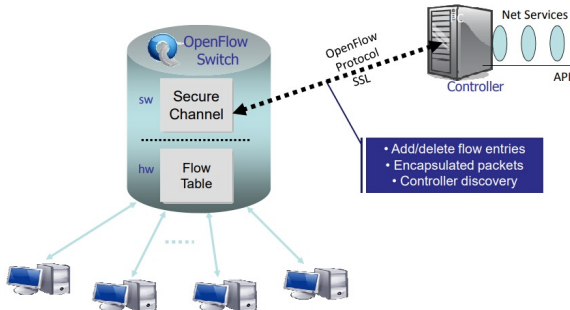
- A Rede Definida por Software é uma nova arquitetura que oportuniza redes mais ágeis e rentáveis, e é composta por três camadas distintas que são acessíveis através de APIs abertas:
  - A camada de aplicação é composta pelas aplicações de negócio do usuário final que consomem os serviços de comunicações SDN. A fronteira entre o Camada de Aplicação e a camada de controle é atravessado pela northbound API.
  - A camada de controle oportuniza o controle consolidado que supervisiona o comportamento de encaminhamento de rede através de uma interface aberta.
  - A camada de infraestrutura consiste de elementos de rede (NE) e dispositivos que proporcionam comutação e encaminhamento de pacotes.



## Northbound e Southbound APIs

- No contexto do SDN, os termos northbound e southbound são frequentemente usados para distinguir se a interface é para os aplicativos ou para os dispositivos.
- Esses termos derivam do fato de que na maioria dos diagramas as aplicações são representadas acima (ou seja, ao norte) do controlador, enquanto os dispositivos são representados abaixo (ou seja, ao sul) do controlador.

- Um exemplo de southbound API é a interface OpenFlow que o controlador usa para programar os dispositivos de rede.

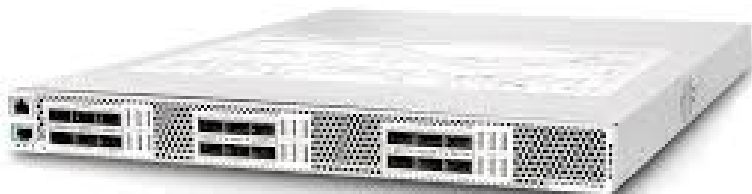


- O controlador oferece uma northbound API, permitindo que aplicativos de software sejam conectados ao controlador e, dessa forma, permitindo que o software forneça os algoritmos e protocolos que podem executar a rede de maneira eficiente.
- Esses aplicativos podem fazer alterações na rede de forma rápida e dinâmica, conforme a necessidade.

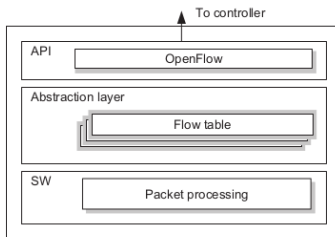
- A northbound API tem como objetivo fornecer uma abstração dos dispositivos de rede e da topologia. Existem três benefícios principais que o desenvolvedor de aplicativos deve derivar da Northbound API :
  - (1) converte para uma sintaxe que é mais familiar para os desenvolvedores, por exemplo, REST ou JSON;
  - (2) fornece abstração da topologia e da camada de rede, permitindo ao programador de aplicativos lidar com a rede como um todo ao invés de nós individuais;
  - (3) fornece abstração dos próprios protocolos de rede, ocultando o desenvolvedor do aplicativo dos detalhes do OpenFlow ou BGP.

- Desta forma, podem ser desenvolvidos aplicativos que funcionam em uma ampla gama de equipamentos de fabricantes que podem diferir substancialmente em seus detalhes de implementação.

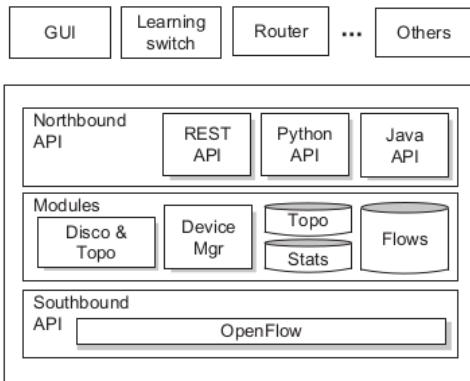
Camada de infraestrutura:



- A camada de infraestrutura (plano de dados) é responsável pelo encaminhamento de tráfego e funções de processamento.
- Dispositivos simples controlados por um sistema centralizado que executa um software de gerenciamento e controle;
- Dispositivos Openflow são equipamentos que conseguem se comunicar com controladores através do protocolo Openflow.



## Controlador- Plano de controle





- O controlador mantém uma visão de toda a rede, implementa decisões de política, controla todos os dispositivos SDN que compõem a infraestrutura de rede e fornece um Northbound API para aplicativos.
- O controlador implementa decisões de política relacionadas a roteamento, encaminhamento, redirecionamento, balanceamento de carga, essas instruções referem-se ao controlador e aos aplicativos que fazem uso desse controlador.
- Os controladores geralmente vêm com seu próprio conjunto de módulos de aplicativos comuns, como um switch de aprendizagem, um roteador, um firewall básico e um balanceador de carga simples.
- O controlador abstrai os detalhes do protocolo de comunicação controlador-dispositivo SDN para que os aplicativos acima sejam capazes de se comunicar com esses dispositivos SDN sem conhecer as nuances desses dispositivos.

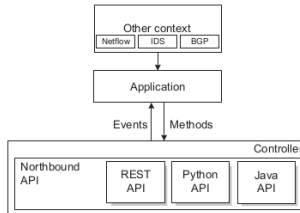
- O controlador pode gerenciar a rede com base em políticas de nível superior;
- O controlador fornece instruções a fim de permitir que os dispositivos tomem decisões rápidas sobre como lidar com os pacotes de entrada.

Os principais recursos do controlador incluirão:

- **Descoberta de dispositivo do usuário final:** descoberta de dispositivos do usuário final, como laptops, desktops, impressoras, dispositivos móveis, etc.
- **Descoberta de dispositivos de rede:** descoberta de dispositivos de rede que constituem a infraestrutura da rede, como switches, roteadores e pontos de acesso sem fio.
- **Gerenciamento da topologia do dispositivo de rede:** mantém informações sobre os detalhes de interconexão dos dispositivos de rede entre si e com os dispositivos do usuário final aos quais estão diretamente conectados.
- **Gerenciamento de fluxo:** mantém um banco de dados dos fluxos sendo gerenciados pelo controlador e executa toda a coordenação necessária com os dispositivos para garantir a sincronização das entradas de fluxo do dispositivo com esse banco de dados.

- O controlador mantém um cache de fluxo que espelha as tabelas de fluxo nos diferentes comutadores que controla. O controlador mantém localmente as estatísticas de fluxo que ele reuniu de seus comutadores.
- O controlador informa a aplicação dos eventos que ocorrem na rede. Os eventos são comunicados do controlador para o aplicativo.

- Os eventos podem pertencer a um pacote individual que foi recebido pelo controlador ou a alguma mudança de estado na topologia da rede, como uma queda de link. Os aplicativos usam métodos diferentes para mudar a operação da rede.
- Tais métodos podem ser invocados em resposta a um evento recebido e podem resultar em um pacote recebido sendo descartado, modificado e / ou encaminhado, ou a adição, exclusão ou modificação de um fluxo.



- Existem várias implementações de controladores SDN disponíveis no mercado hoje. Eles incluem controladores SDN de código aberto e controladores SDN comerciais. Os controladores SDN de código aberto vêm em muitas formas:

Controlador	Implementação	Software Livre	Desenvolvedor
POX	Python	Sim	Nicira
NOX	Python/C++	Sim	Nicira
MUL	C	Sim	Kulcloud
Maestro	Java	Sim	Universidade Rice
Trema	Ruby/C	Sim	NEC
Beacon	Java	Sim	Stanford
Jaxon	Java	Sim	Desenvolvedores Independentes
Helios	C	Não	NEC
Floodlight	Java	Sim	BigSwitch
SNAC	C++	Não	Nicira
Ryu	Python	Sim	NTT, grupo OSRG
NodeFlow	JavaScript	Sim	Desenvolvedores Independentes
ovs-controller	C	Sim	Desenvolvedores Independentes
Flowvisor	C	Sim	Stanford/Nicira
RouteFlow	C++	Sim	CPqD

Plano de aplicação:

Os aplicativos SDN são executados acima do controlador SDN, fazendo interface com a rede por meio da northbound API do controlador. Os aplicativos SDN são responsáveis por gerenciar as entradas de fluxo que são programadas nos dispositivos de rede, usando a API do controlador para gerenciar fluxos. Por meio dessa API, os aplicativos são capazes de:

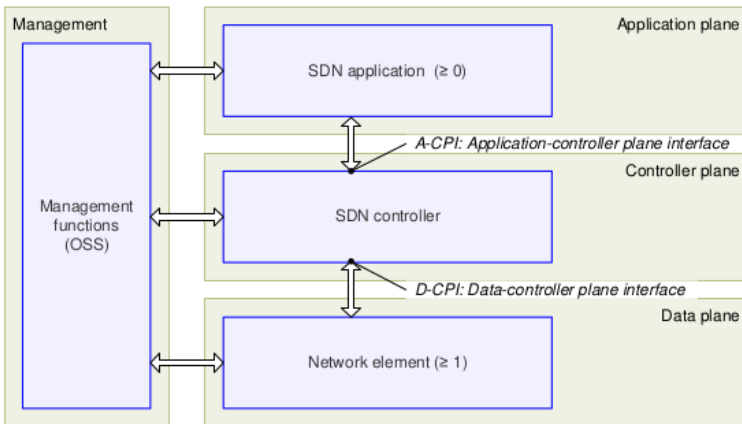
- (1) configurar os fluxos para rotear os pacotes pelo melhor caminho entre dois terminais;
- (2) equilibrar as cargas de tráfego em vários caminhos ou destinadas a um conjunto de terminais;
- (3) reagir a mudanças na topologia da rede, como falhas de link e adição de novos dispositivos e caminhos,
- (4) redirecionar o tráfego para fins de inspeção, autenticação, segregação e tarefas semelhantes relacionadas à segurança.



- O comportamento da camada de aplicação é orientado por eventos provenientes do controlador, bem como por entradas externas.
- As entradas externas podem incluir sistemas de monitoramento de rede, como Netflow, IDS ou BGP peer.
- O aplicativo SDN monitora certos eventos, e o controlador invocará o método de retorno de chamada do aplicativo sempre que tal evento ocorrer. Esta invocação será acompanhada pelos detalhes apropriados relacionados ao evento. Alguns exemplos de eventos tratados por um aplicativo SDN são: Descoberta de dispositivo do usuário final, Descoberta de dispositivo de rede e pacote de entrada.

- Nos dois primeiros casos, os eventos são enviados para o aplicativo SDN após a descoberta de um novo dispositivo de usuário final (ou seja, endereço MAC) ou um novo dispositivo de rede (por exemplo, switch, roteador ou ponto de acesso sem fio), respectivamente.
- Os eventos de pacote de entrada são enviados para o aplicativo SDN quando um pacote é recebido de um dispositivo SDN, devido a uma entrada de fluxo instruindo o dispositivo SDN a encaminhar o pacote para o controlador ou porque não há entrada de fluxo correspondente no dispositivo SDN.
- Quando não há nenhuma entrada de fluxo correspondente, a ação padrão geralmente é encaminhar o pacote ao controlador, embora possa ser descartá-lo, dependendo da natureza dos aplicativos.

## Plano de gerenciamento



- O conceito de gerenciamento engloba operações de suporte à infraestrutura, como instalação e manutenção de equipamentos e atualização de software.
- As funções de gerenciamento especiais para SDN incluem a alocação de recursos e política para clientes ou aplicativos SDN específicos.
- Um plano de dados pode incluir o subconjunto mínimo necessário de funções de controle e gerenciamento. No plano de dados, o gerenciamento é pelo menos necessário para configurar inicialmente os elementos de rede;
- Em todos os planos, o gerenciamento configura as associações de segurança que permitem que funções distribuídas se intercomunique com segurança.

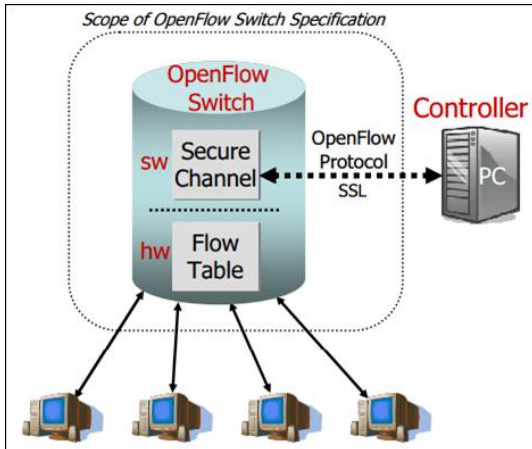
## OpenFlow

- Como se dá a comunicação entre o controlador e os dispositivos de rede genéricos?
- Qual deve ser o formato das mensagens enviadas pelo controlador?

## OpenFlow

- Stanford;
- Nicira;
- Martin Casado, Nick McKeown e Scott Shenker;
- Vendida para VmWare em 2011 por 1.2 bilhões de dólares.

- Composto por dois componentes principais:
  - Controlador OpenFlow
  - Dispositivos OpenFlow



## OpenFlow

- OpenFlow é um padrão aberto para um protocolo de comunicação que permite que o plano de controle se separe e interaja com o plano de encaminhamento de vários dispositivos a partir de algum ponto central, desacoplando funções para maior funcionalidade e programação;
- O protocolo OpenFlow define a interface entre um controlador OpenFlow e um switch OpenFlow e com isso instrua o switch OpenFlow sobre como lidar com os pacotes de dados recebidos.



- Um switch lógico OpenFlow consiste em uma ou mais tabelas de fluxo e uma tabela de grupo, que realiza pesquisas e encaminhamento de pacotes, e um ou mais canais OpenFlow para um controlador externo.
- O switch se comunica com o controlador e o controlador gerencia o switch por meio do protocolo de switch OpenFlow.

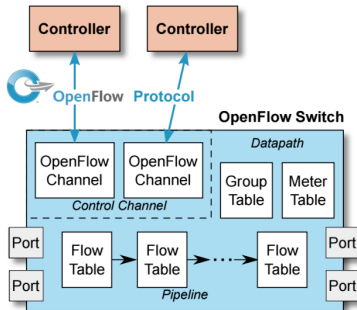
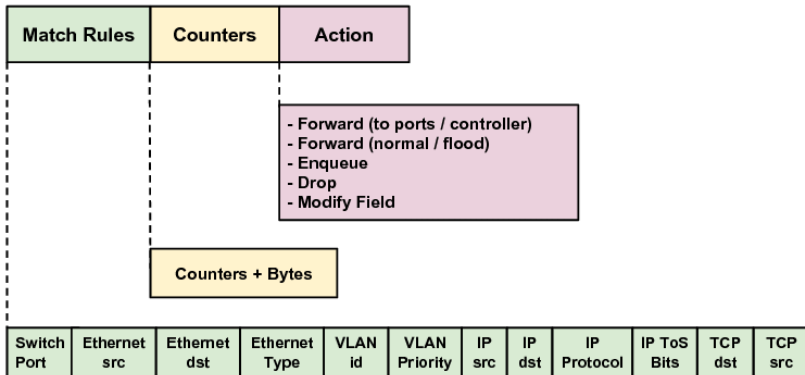


Figure 1: Main components of an OpenFlow switch.

## Tabelas de fluxo

- Uma tabela de fluxo reside no dispositivo de rede e consiste em uma série de entradas de fluxo e as ações a serem executadas quando um pacote correspondente a esse fluxo chega ao dispositivo;
- Usando o protocolo de comutação OpenFlow, o controlador pode adicionar, atualizar e excluir entradas de fluxo nas tabelas de fluxo, tanto de forma reativa (em resposta aos pacotes) quanto de forma proativa;
- As entradas de fluxo reativo são criadas quando o controlador aprende dinamicamente onde os dispositivos estão na topologia e deve atualizar as tabelas de fluxo nesses dispositivos para construir conectividade ponta a ponta.
- Por exemplo, uma vez que os switches em um ambiente OpenFlow puro são simplesmente encaminhadores de tráfego, toda lógica racional deve primeiro ser ditada e programada pelo controlador.

## Entrada da tabela de fluxo



## Comandos para modificações de fluxos:

```
enum ofp_flow_mod_command {  
    OFPFC_ADD,           /* New flow. */  
    OFPFC_MODIFY,        /* Modify all matching flows. */  
    OFPFC_MODIFY_STRICT, /* Modify entry strictly matching wildcards */  
    OFPFC_DELETE,        /* Delete all matching flows. */  
    OFPFC_DELETE_STRICT  /* Strictly match wildcards and priority. */  
};
```

### No Mininet:

- add-flow s1 in\_port=1,actions=output:2
- add-flow s1  
dl\_src=00:00:00:00:00:01,dl\_dst=00:00:00:00:00:02,actions=output:2

- Portanto, se um host no switch A precisar conversar com um switch host B, as mensagens serão enviadas ao controlador para descobrir como chegar a este host. O controlador aprenderá as tabelas de endereço MAC do host dos switches e como eles se conectam, programando a lógica nas tabelas de fluxo de cada switch. Esta é uma entrada de fluxo reativa.
- As entradas proativas de fluxo são programadas antes que o tráfego chegue. Se já for sabido que dois dispositivos devem ou não se comunicar, o controlador pode programar essas entradas de fluxo nos endpoints OpenFlow com antecedência.

- Em uma rede OpenFlow, cada switch OpenFlow contém pelo menos uma tabela de fluxo e um conjunto de entradas de fluxo nessa tabela.
- Essas entradas de fluxo contêm campos de correspondência, contadores e instruções de ações para aplicar aos pacotes correspondentes.
- Normalmente, você terá mais de uma única tabela de fluxo, por isso é importante observar que a correspondência começa na primeira tabela de fluxo e pode continuar para tabelas de fluxo adicionais do pipeline.
- Necessidade de separar as tabelas por características.

## Campos coringa (wildcards)

- Os campos de correspondência podem ter coringas para campos que não são relevantes para aquela correspondência específica.
- Por exemplo, ao combinar pacotes com base apenas no endereço IP ou sub-rede, todos os outros campos receberiam caracteres coringa. Da mesma forma, se corresponder apenas no endereço MAC ou porta UDP / TCP, os outros campos são irrelevantes e, conseqüentemente, esses campos são coringa.
- Dependendo das necessidades do aplicativo, todos os campos podem ser importantes e, nesse caso, não haveria coringas.
- A tabela de fluxo e as construções de entrada de fluxo permitem que o desenvolvedor de aplicativos SDN tenha uma ampla gama de possibilidades para combinar pacotes e tomar as ações apropriadas.

## Exemplos de tabela de fluxos:

Counter	Action	Dst L4 Port ICMP Code	Src L4 Port ICMP Type	IP ToS	IP Proto	Dst IP	Src IP	EtherType	Priority	VLAN ID	Dst MAC	Src MAC	Port
102	Port 1	*	*	*	*	*	*	*	*	*	0A:C8:*	*	*
202	Port 2	*	*	*	*	192.168.*.*	*	*	*	*	*	*	*
420	Drop	21	21	*	*	*	*	*	*	*	*	*	*
444	Local	*	*	*	0x806	*	*	*	*	*	*	*	*
1	Controller	*	*	*	0x1*	*	*	*	*	*	*	*	*



Existem dois tipos de Switches OpenFlow: OpenFlow-only e OpenFlow-hybrid.

- Switches somente OpenFlow são "switches burros" tendo apenas um plano de dados / encaminhamento e nenhuma maneira de tomar decisões locais. Todos os pacotes são processados pelo pipeline OpenFlow e não podem ser processados de outra forma.
- Os switches híbridos OpenFlow suportam a operação OpenFlow e a operação normal de switching Ethernet. Isso significa que você pode usar comutação Ethernet L2 tradicional, isolamento de VLAN, roteamento L3, ACLs e processamento de QoS por meio do plano de controle local do switch enquanto interage com o pipeline OpenFlow usando vários mecanismos de classificação.

- Podemos ter um switch com metade de suas portas usando roteamento e comutação tradicionais, enquanto a outra metade é configurada para OpenFlow.
- A metade OpenFlow seria gerenciada por um controlador OpenFlow e a outra metade pelo plano de controle do switch local.

Os switches OpenFlow devem possuir/suportar as seguintes características

- Protocolo OpenFlow;
- Canal Seguro;
- Tabela de Fluxos.

# Protocolo Openflow

O protocolo OpenFlow suporta três tipos diferentes de mensagens:

- Controlador-Switch - Geradas pelo controlador para gerenciar e inspecionar o estado de um switch;
- Assíncronas - Geradas pelo switch para atualizar o controlador sobre eventos da rede e mudanças no estado do switch;
- Simétricas - Podem ser geradas tanto pelo controlador quanto pelo switch. São enviadas sem solicitação;

## Controlador-Switch

- Características (Features) - O controlador requisita as características do switch. O switch deve responder com as características suportadas;
- Configuração (Configuration) - Usado para configurar ou solicitar configurações do switch;
- Modificação de estado (Modify-State) - Usado para adicionar, deletar e modificar a tabela de fluxos e para setar propriedades nas portas do switch;
- Leitura de estado (Read-State) - Coleta estatísticas;
- Envio de pacote (Send-Packet) - Utilizado para enviar pacotes por uma determinada porta do switch;
- Barreira (Barrier) - Usado para garantir que as dependências foram atendidas ou para receber notificações de operações finalizadas;

## Assíncrona

- Entrada de pacotes (Packet-In) - Utilizado quando fluxos não classificados entram no switch.
- Remoção de fluxo (Flow-Removed) - Mensagem enviada para o controlador, quando um fluxo é removido da tabela. Seja por Timeout ou por uma mensagem de modificação da tabela de fluxos que delete a entrada em questão;
- Estado da porta (Port-Status) - Mensagem enviada para o controlador sempre que há mudanças nas configurações das portas;
- Erro (Error) - Notificações de erros;

## Simétrica

- Hello - Mensagens trocadas entre o controlador e o switch quando uma conexão é estabelecida;
- Echo - Mensagens usadas para identificação de latência, largura de banda e existência de conectividade;
- Vendor - Provêem uma forma padrão para os switches OpenFlow oferecerem funcionalidades adicionais;



## Sequência de conexão OpenFlow

- O switch pode iniciar a conexão com o IP do controlador e a porta de transporte padrão (TCP 6633 pré-OpenFlow 1.3.2, post TCP 6653) ou uma porta especificada pelo usuário.
- O controlador também pode iniciar a solicitação de conexão, mas isso não é comum. Conexão TCP ou TLS estabelecida
- Ambos enviam um OFPT\_HELLO com um campo de versão preenchido
- Ambos calculam a versão negociada a ser usada
- Se isso não puder ser acordado, uma mensagem OFPT\_ERROR será enviada
- Se cada um suporta a versão, o controlador envia um OFPT\_FEATURES-REQUEST para reunir o Datapath ID do switch, junto com os recursos do switch.

Portas OpenFlow - Existem três tipos de portas que um switch OpenFlow deve suportar: portas físicas, portas lógicas e portas reservadas.

- As portas físicas são portas definidas pelo switch que correspondem a uma interface de hardware no switch. Isso pode significar mapeamento das portas físicas OpenFlow para interfaces Ethernet definidas por hardware no switch, mas não necessariamente precisa ser um a um.
- As portas lógicas são portas definidas pelo switch que não correspondem diretamente às interfaces de hardware no switch. Por exemplo, interfaces de loopback.

- As portas reservadas OpenFlow especificam ações de encaminhamento genéricas, como enviar para o controlador, broadcast ou encaminhamento usando métodos não OpenFlow, como processamento de switch "normal".
- Existem vários tipos de portas reservadas necessárias: ALL, CONTROLLER, TABLE, IN\_PORT, ANY, UNSET, LOCAL. A porta CONTROLLER representa o canal OpenFlow usado para comunicação entre o switch e o controlador.
- Em ambientes híbridos, você também verá as portas NORMAL e FLOOD para permitir a interação entre o pipeline OpenFlow e o pipeline de hardware do switch.

Canal seguro:

- O canal seguro irá conectar o switch OpenFlow ao controlador e permitirá que os comandos passem por ele.
- O controlador irá configurar e gerenciar o switch OpenFlow.
- Objetivo de garantir confidencialidade em toda comunicação.

O switch OpenFlow deve ser capaz de realizar três simples ações com os fluxos entrantes:

- Encaminhar o fluxo entrante para uma determinada interface de saída, caso haja uma entrada equivalente para o fluxo em questão em sua tabela de fluxos;
- Caso não haja entrada equivalente na tabela de fluxos, o switch deve ser capaz de encapsular o primeiro pacote (ou todos, de acordo com a necessidade) e encaminhá-lo para o controlador através do canal seguro. O controlador que irá decidir o que fazer com esse pacote. Inclusive, decidirá se uma entrada na tabela de fluxos do switch será adicionada ou não;
- Por último, o switch OpenFlow também deve ser capaz de descartar pacotes;

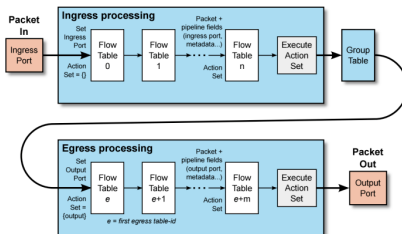
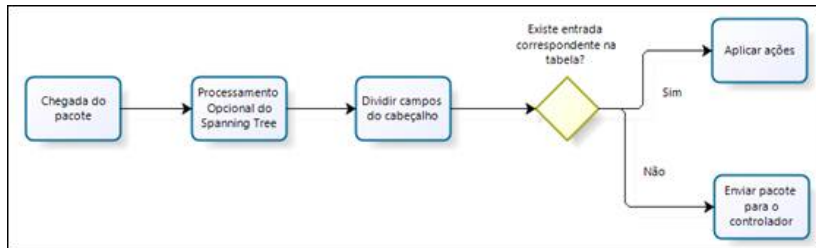
# Contadores

Counter	Bits
<b>Per Table</b>	
Active Entries	32
Packet Lookups	64
Packet Matches	64
<b>Per Flow</b>	
Received Packets	64
Received Bytes	64
Duration (seconds)	32
Duration (nanoseconds)	32
<b>Per Port</b>	
Received Packets	64
Transmitted Packets	64
Received Bytes	64
Transmitted Bytes	64
Receive Drops	64
Transmit Drops	64
Receive Errors	64
Transmit Errors	64
Receive Frame Alignment Errors	64
Receive Overrun Errors	64
Receive CRC Errors	64
Collisions	64
<b>Per Queue</b>	
Transmit Packets	64
Transmit Bytes	64
Transmit Overrun Errors	64

# Ações

- Encaminhamento
  - Obrigatório
    - ALL - Envia o pacote para todas as interfaces, exceto a interface de entrada;
    - CONTROLLER - Encapsula e envia o pacote para o controlador;
    - LOCAL - Envia o pacote para a pilha de rede local;
    - TABLE - Realiza ações na tabela de fluxos;
    - IN\_PORT - Envia o pacote para a porta de entrada;
  - Opcional
    - NORMAL - Processa o pacote utilizando um encaminhamento tradicional;
    - FLOOD - Inunda o pacote, sem incluir a interface de entrada, levando em consideração o Spanning Tree.
- Enfileirar (opcional) - Encaminha o pacote através de uma fila relacionada a uma porta;
- Descartar (obrigatória);
- Modificar campo (opcional):
  - Setar Vlan ID
  - Setar Vlan Priority
  - Separar o cabeçalho da Vlan
  - Modificar endereço MAC (Media Access Control) de origem
  - Modificar endereço MAC de destino
  - Modificar endereço IP de origem
  - Modificar endereço IP de destino
  - Modificar ToS
  - Modificar a porta de transporte de origem
  - Modificar a porta de transporte de destino

# Processamento de pipeline OpenFlow





- O processamento de pipeline OpenFlow define como os pacotes interagem com as tabelas de fluxo contidas por um switch.
- As tabelas de fluxo de um switch OpenFlow são numeradas sequencialmente, começando em 0. O pacote é primeiro correspondido com as entradas de fluxo da primeira tabela de fluxo, que é a tabela de fluxo 0. Uma entrada de fluxo só pode direcionar um pacote para um número de tabela de fluxo que é maior que seu próprio número de tabela de fluxo.
- Se uma entrada correspondente for encontrada, as instruções associadas à entrada de fluxo específica são executadas.
- A entrada do fluxo Table-miss é a última na tabela, tem uma prioridade de 0 e uma correspondência de qualquer coisa. É como um resumo geral e as ações a serem tomadas dependem de como você o configura. Você pode encaminhar o pacote para o controlador pelo canal OpenFlow, ou pode descartar o pacote ou continuar com a próxima tabela de fluxo.

## OpenFlow 1.1

- Na especificação do OpenFlow 1.1, os switches OpenFlow contêm diversas tabelas de fluxos e uma tabela de grupo, ao invés de uma única tabela de fluxos como na versão 1.0.0.
- Um fluxo de pacotes ao entrar no switch OpenFlow, poderá passar por várias tabelas de fluxos, para que diversas ações diferentes sejam realizadas. A tabela de grupo é um tipo especial de tabela projetada para executar ações que sejam comuns para diversos fluxos.

## OpenFlow 1.2

- O OpenFlow 1.2 foi lançado em dezembro de 2011 e uma das principais implementações foi o suporte ao protocolo IPv6, incluindo novos campos de cabeçalho nas tabelas de fluxos.
- Além disso, o mesmo passou a suportar a possibilidade dos switches se conectarem a mais de um controlador ao mesmo tempo. Ao se conectar a vários controladores, aumenta-se a segurança já que o switch pode continuar a operar normalmente mesmo se um controlador ou a conexão ficar indisponível.

## OpenFlow 1.3

- Sendo lançada em Junho de 2012, a versão 1.3.0 do OpenFlow passou a suportar o controle das taxas de pacotes através de medidores de fluxos, introduzindo a tabela de medições Meter Table.
- A tabela de medições por fluxo permite ao OpenFlow implementar simples operações de QoS, como limitação de taxas de transmissão e pode ser combinada com as filas por porta para criar políticas de QoS mais complexas.
- Além disso, a versão 1.3 permite ao switch criar conexões auxiliares para o controlador, ajudando a melhorar o desempenho de processamento do switch, explorando o paralelismo presente na maioria dos switches (Foundation, 2012a).

## OpenFlow 1.4

- Na versão 1.4 do OpenFlow portas ópticas passaram a ser suportadas.
- Além disso, os conceitos de Eviction e Vacancy events foram introduzidos para evitar que as tabelas de fluxos fiquem cheias, já que as mesmas tem capacidades finitas. Nas especificações anteriores quando uma tabela de fluxos enchia, novos fluxos não eram inseridos nas tabelas e uma mensagem de erro era enviada para o controlador. Entretanto, atingir tal situação era problemático e poderia causar interrupções no serviço.
- O Eviction adiciona um mecanismo que permite ao switch apagar automaticamente as entradas nas tabelas de fluxos que tenham menor importância.
- Já o Vacancy events permite ao controlador configurar um limiar, que ao ser atingido, faz o switch enviar mensagens de alerta para o controlador. Isso permite que o controlador possa reagir com antecedência, evitando que as tabelas de fluxos fiquem cheias (Foundation, 2012b).

## Referências:

- Caps. 4 e 5 do livro Goransson, Paul, Chuck Black, and Timothy Culver. Software defined networks: a comprehensive approach. Morgan Kaufmann, 2016.

## Vídeos Interessantes

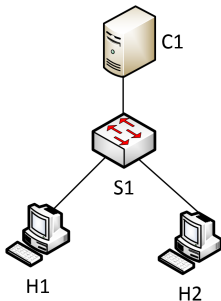
- Redes Definidas por Software parte 1/2  
<https://www.youtube.com/watch?v=M0cBvh1RPF0>
- Redes Definidas por Software parte 2/2  
<https://www.youtube.com/watch?v=I14r54581HY>
- OpenFlow: da teoria à prática, do conceito ao tráfego real  
<https://www.youtube.com/watch?v=YeVTE0DVquQ>

Ambiente de simulação:

- Mininet: <http://mininet.org/>;
- Virtualbox ou Vmware;
- Wireshark



# Simulação Mininet



```
Mininet-VM [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda

Ubuntu 14.04 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Mon Feb  8 08:42:29 PST 2016 from 192.168.56.1 on pts/6
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$
```

A screenshot of a terminal window titled 'Mininet-VM [Executando] - Oracle VM VirtualBox'. The terminal shows the login process for a user named 'mininet' on an Ubuntu 14.04 LTS system. The prompt is 'mininet@mininet-vm:~\$'. The window includes standard Ubuntu window controls and a menu bar with options like 'Arquivo', 'Máquina', 'Visualizar', 'Entrada', 'Dispositivos', and 'Ajuda'.

# Wireshark

Local Area Connection 2 [Wireshark 2.24 (v2.24.0-gc3dc1b)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: openflow\_v4 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
3867	71.473111	192.168.0.11	192.168.0.12	OpenFlow	82	Type: OFPT_HELLO
3869	71.477795	192.168.0.12	192.168.0.11	OpenFlow	82	Type: OFPT_HELLO
3871	71.478411	192.168.0.12	192.168.0.11	OpenFlow	74	Type: OFPT_FEATURES_REQUEST
3873	71.478805	192.168.0.11	192.168.0.12	OpenFlow	98	Type: OFPT_FEATURES_REPLY
3879	71.520722	192.168.0.12	192.168.0.11	OpenFlow	90	Type: OFPT_ROLE_REQUEST
3880	71.520960	192.168.0.11	192.168.0.12	OpenFlow	90	Type: OFPT_ROLE_REPLY
3882	71.527038	192.168.0.12	192.168.0.11	OpenFlow	74	Type: OFPT_BARRIER_REQUEST
3883	71.527380	192.168.0.11	192.168.0.12	OpenFlow	74	Type: OFPT_BARRIER_REPLY
3884	71.529609	192.168.0.12	192.168.0.11	OpenFlow	98	Type: OFPT_BARRIER_REQUEST
3885	71.529794	192.168.0.11	192.168.0.12	OpenFlow	90	Type: OFPT_ROLE_REPLY
3886	71.530025	192.168.0.12	192.168.0.12	OpenFlow	74	Type: OFPT_BARRIER_REPLY
3888	71.539437	192.168.0.12	192.168.0.11	OpenFlow	82	Type: OFPT_MULTIPART_REQUEST, OFPMP_TABLE_FEATURES
3889	71.539681	192.168.0.11	192.168.0.12	OpenFlow	94	Type: OFPT_ERROR
3890	71.566798	192.168.0.12	192.168.0.11	OpenFlow	282	Type: OFPT_FLOW_MOD
3891	71.567186	192.168.0.12	192.168.0.12	OpenFlow	274	Type: OFPT_MULTIPART_REPLY, OFPMP_PORT_DESC
3892	71.567188	192.168.0.11	192.168.0.12	OpenFlow	1138	Type: OFPT_MULTIPART_REPLY, OFPMP_DESC
3894	71.567323	192.168.0.11	192.168.0.12	OpenFlow	94	Type: OFPT_ERROR
3895	71.567325	192.168.0.11	192.168.0.12	OpenFlow	98	Type: OFPT_MULTIPART_REPLY, OFPMP_METER_FEATURES
3898	71.616364	192.168.0.12	192.168.0.11	OpenFlow	355	Type: OFPT_FLOW_MOD
3900	72.386944	192.168.0.11	192.168.0.12	OpenFlow	146	Type: OFPT_PORT_STATUS
3902	72.390077	192.168.0.12	192.168.0.11	OpenFlow	191	Type: OFPT_PACKET_OUT
3904	72.491674	192.168.0.12	192.168.0.11	OpenFlow	316	Type: OFPT_PACKET_OUT
3908	75.552798	192.168.0.12	192.168.0.11	OpenFlow	274	Type: OFPT_FLOW_MOD
3910	76.548767	192.168.0.12	192.168.0.11	OpenFlow	172	Type: OFPT_MULTIPART_REQUEST, OFPMP_FLOW

☒ Frame 3867: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0  
☒ Ethernet II, Src: PcsCompu\_06:dc:e4 (08:00:27:06:dc:e4), Dst: PcsCompu\_92:80:fb (08:00:27:92:80:fb)  
☒ Internet Protocol Version 4, Src: 192.168.0.11, Dst: 192.168.0.12  
☒ Transmission Control Protocol, Src Port: 60032, Dst Port: 6633, Seq: 1, Ack: 1, Len: 16  
☒ OpenFlow 1.3  
     version: 1.3 (0x04)  
     Type: OFPT\_HELLO (0)  
     Length: 16  
     Transaction ID: 4  
☒ Element  
     Type: OFPHET\_VERSIONBITMAP (1)  
     Length: 8  
     Bitmap: 00000010

## Instalação:

- Baixar a imagem Mininet ubuntu de acordo com o seu pc 32 ou 64 bits em  
<https://github.com/mininet/mininet/releases>;
- Baixar e instalar o virtualbox em  
<https://www.virtualbox.org/wiki/Downloads>
- Instalar o mininet na máquina virtual, sigam os passos deste vídeo: <https://youtu.be/TGmh6ppSS3M?list=FL8rCt095hxrCVZZRilHSuA>
- Instalar a interface gráfica usando Xming pelo link  
<https://sourceforge.net/projects/xming/>;
- Instalar o Putty pelo link <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Seguir as instruções do link  
<https://youtu.be/RTlituHdp7M> para acessar a máquina virtual pelo putty e xterm.

Tutorial Mininet

<http://mininet.org/walkthrough/>