



Introdução a SDN e OpenFlow

Renato Yoshio Soma RA 137478
Wilson Rodolfo de Souza RA 227016



Definições gerais para a apresentação

Layer 2 - camada de dados (data link layer). Responsável pela transferência de dados entre dispositivos em um único link.

Layer 3 - camada de redes (network layer). Responsável pela transferência de dados entre subredes.

Switch - dispositivo que recebe informação em uma porta e a encaminha para outra(s) portas.

Arquitetura de um Switch

Um switch é tradicionalmente composto de três partes/planos:

1. Plano de dados (data plane)
2. Plano de controle (control plane)
3. Plano de gerência (management plane)

Data plane

Responsável pela recepção e transmissão de pacotes.

Basicamente constituído de portas e uma tabela de encaminhamento (forwarding table), a qual é consultada para o encaminhamento de um pacote recebido.

O encaminhamento é feito baseado no header dele e um match na tabela. Dependendo do caso, o pacote é encaminhado diretamente pelo data plane, ou é encaminhado ao control plane.

Control plane

Responsável por manter a tabela de encaminhamento atualizada, de forma que o data plane encaminhe o maior tráfego possível.

Processa diferentes protocolos existentes de forma a manter a tabela atualizada. De tal forma, é capaz de ter noção da topologia da rede.

Exemplo de protocolos neste plano: Spanning Tree Protocol (STP), Open Shortest Path First (OSPF).

Management plane

O plano de gerência é a interface do switch com o operador da rede.

Operadores são capazes de configurar e monitorar estatísticas do switch por meio deste plano.

Este plano é então capaz de alterar o estado dos outros dois planos apresentados (control/data).

Exemplo de protocolo neste plano: SNMP

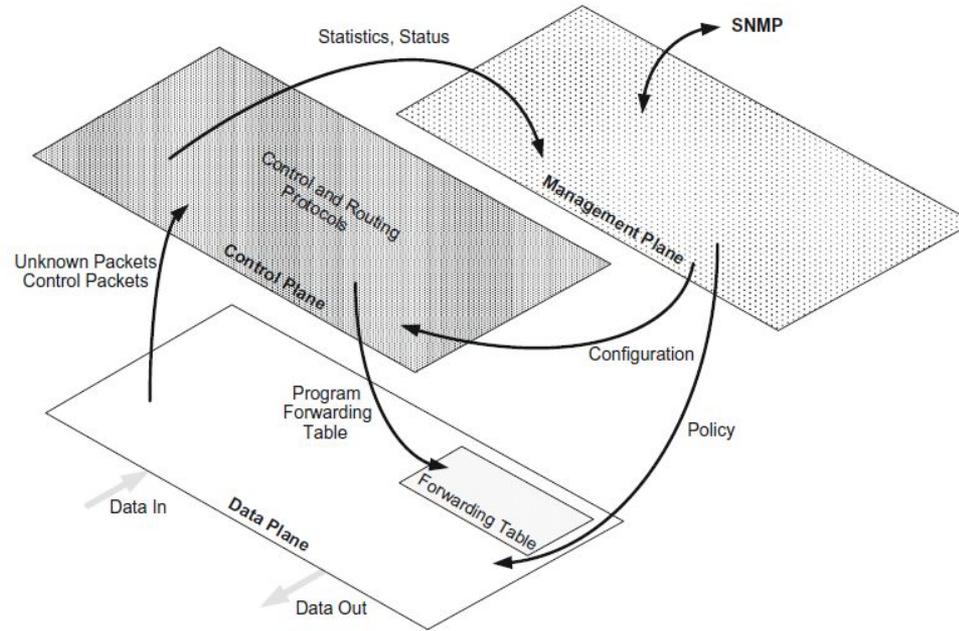


FIGURE 1.2

Roles of the control, management, and data planes.

Fonte: Software Defined Networks, a Comprehensive Approach. 1º Edição

Evolução dos Switches

1980 - 1990: Início da Internet, switches realizavam o roteamento de pacotes por meio de software. Poder computacional limitado com evolução de tecnologia de comunicação causou mudança para o próximo período.

1990 - presente: Desenvolvimento de IC's para realizar table lookups com aceleração em hardware.

2007 - presente: Necessidade em adicionar mais lógica nas tabelas de encaminhamento, refinar o controle: início do SDN.

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.
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.

Os 3 pontos acima são problemas no plano de controle e podem ser resolvidos por meio de uma abordagem centralizada, um dos pilares do SDN.

SDN

Software Defined Network é uma abordagem para gerenciar uma rede de forma mais programável.

Tem como princípios básicos:

1. Separação do plano de controle do plano de dados.
2. Centralização do plano de controle.
3. Simplificação da lógica interna dos dispositivos.

SDN

A idéia central do SDN é permitir que o plano de controle possa gerenciar os diversos elementos do plano de dados de forma transparente por meio de uma API bem definida (ex: OpenFlow).

SDN

SDN abrange as camadas L2 e L3, não devendo impactar outras camadas.

SDN promove a remoção do **plano de controle** nos switches e realocação para um servidor centralizado, capaz de se comunicar com todos os outros elementos da rede.

Ao centralizar o plano de controle e separá-lo do plano de dados, obtemos mais simplicidade e flexibilidade na rede, visto que ele tem a visão da topologia da rede e é capaz de tomar as melhores decisões globais de forma rápida.

SDN - Separação dos planos

No SDN, temos separação do plano de dados (encaminhamento de pacotes) do de controle.

O plano de dados continua com as tabelas de encaminhamento e informações como MAC, endereços IP, ID de VLAN.

Os protocolos de roteamento e algoritmos para controlar o plano de dados ficam no plano de controle. No entanto, esse plano de controle passa a ser centralizado e externo ao domínio do switch.

SDN - Controlador centralizado

Ao remover o plano de controle dos switches, simplificamos a lógica necessária a eles, de forma que todo o controle e gerência dos dispositivos passem a se encontrar no controlador centralizado, conhecedor da rede.

A centralização permite que o operador não precise se preocupar com detalhes do hardware dos dispositivos controlados, as API's expostas a ele permitem o uso transparente dos dispositivos. Com isso, possibilita controle automatizado da rede também.

1. Aplicações
2. API's northbound
3. Controlador
4. API southbound (OpenFlow)
5. Dispositivos SDN

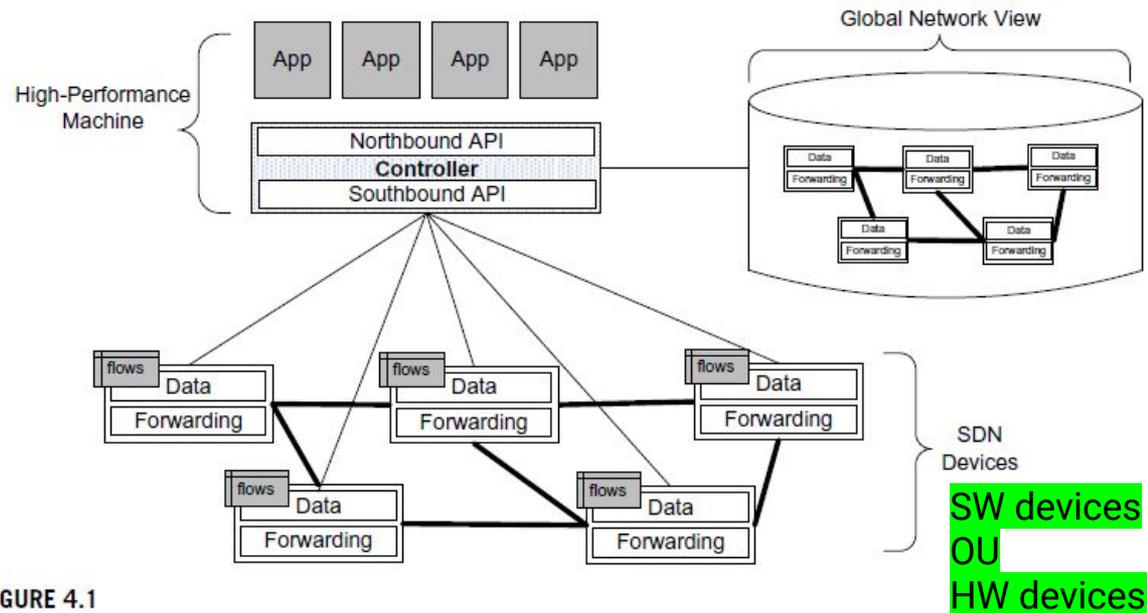


FIGURE 4.1

SDN operation overview.

Fonte: Software Defined Networks, a Comprehensive Approach. 1º Edição

SDN - Funcionamento geral

Os dispositivos SDN contém as tabelas de **flow** e dados que são usados para determinar a ação a ser tomada no recebimento de determinado pacote.

Flow (fluxo) é uma sequência de pacotes enviadas de um endpoint a outro.

A tabela de flows é composta por entradas de flow e as ações correspondentes a ele.

SDN - Funcionamento geral

Quando um pacote chega ao dispositivo, uma consulta à tabela é realizada, de forma a achar uma entrada que case com o pacote recebido.

Se nenhuma entrada na tabela casar com o pacote recebido, o pacote pode ser dropado ou ser repassado ao controlador.

SDN - Funcionamento geral

O controlador centralizado é responsável por ter a visão geral da rede por abstração da rede gerenciada por ele.

Também deve alimentar as tabelas de flow e ajudar a responder por pacotes que forem encaminhados a ele pelos dispositivos SDN.

Finalmente, é responsável por interfacear com as aplicações SDN que quiserem se comunicar com os dispositivos.

OpenFlow

OpenFlow

- Padronização
- Controle do tráfego da rede
- Reutilização da arquitetura de rede existente (tabelas switch's)

OpenFlow

- Open Source
- Stanford University e University of California
- Open Networking Foundation (ONF)

OpenFlow

Composto por dois componentes principais:

1. Controlador OpenFlow
2. Dispositivos OpenFlow

OpenFlow - Controlador

- Centralizado
- Comunicação segura com o Switch OpenFlow (SSL)
- Controlador das operações do plano de dados
 - Roteamento
 - Firewall
 - Priorização dos pacotes
- ***Gargalo***

OpenFlow - Switches

- Dividido em três partes:
 - Comunicação segura com o Controlador OpenFlow (SSL)
 - Interface do protocolo OpenFlow
 - Tabela de fluxos

OpenFlow - Switches

- Ações:
 - Encaminhar o pacote a determinada porta (física/virtual)
 - Dropar o pacote
 - Encaminhar o pacote ao controlador

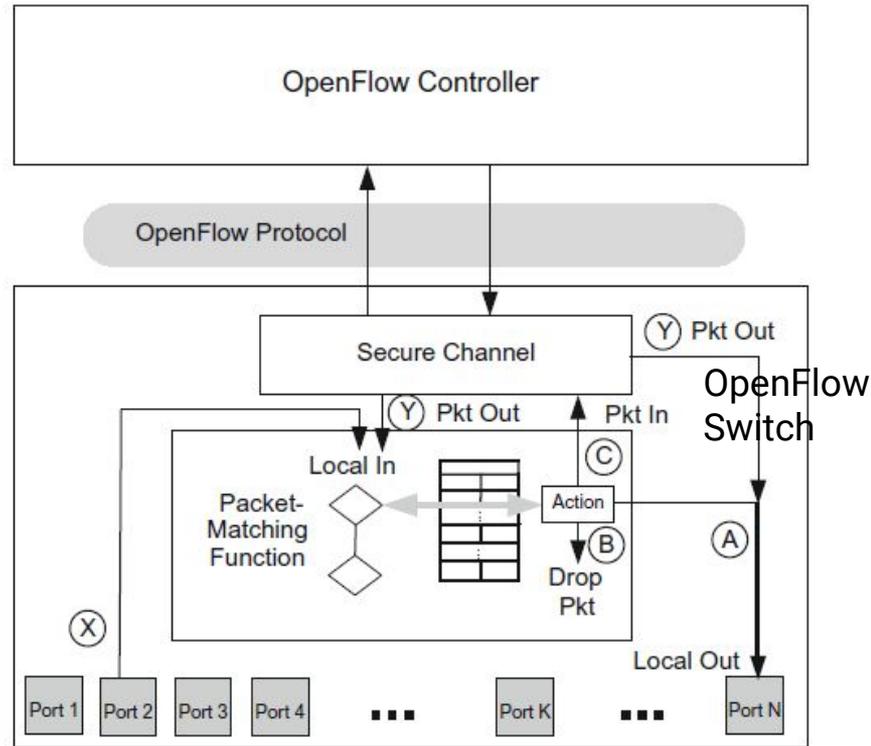


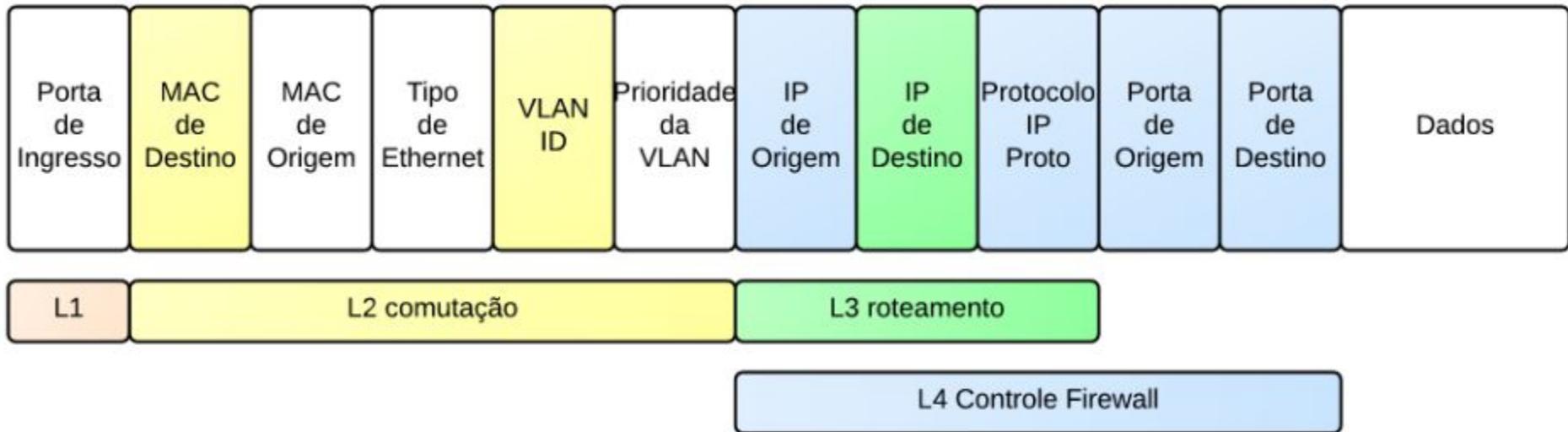
FIGURE 5.2

OpenFlow V.1.0 switch.

OpenFlow - Tabelas de fluxo

- Header
- Ações
- Contadores

Tabelas de fluxo - Header



Tabelas de fluxo - Ações

- Forwarding
- Set
- Drop
- Strip
- Push
- Pop
- Copy-in
- Copy-out
- Dec

Tabelas de fluxo - Contadores

- Controle
- Identificação
- Estatísticas

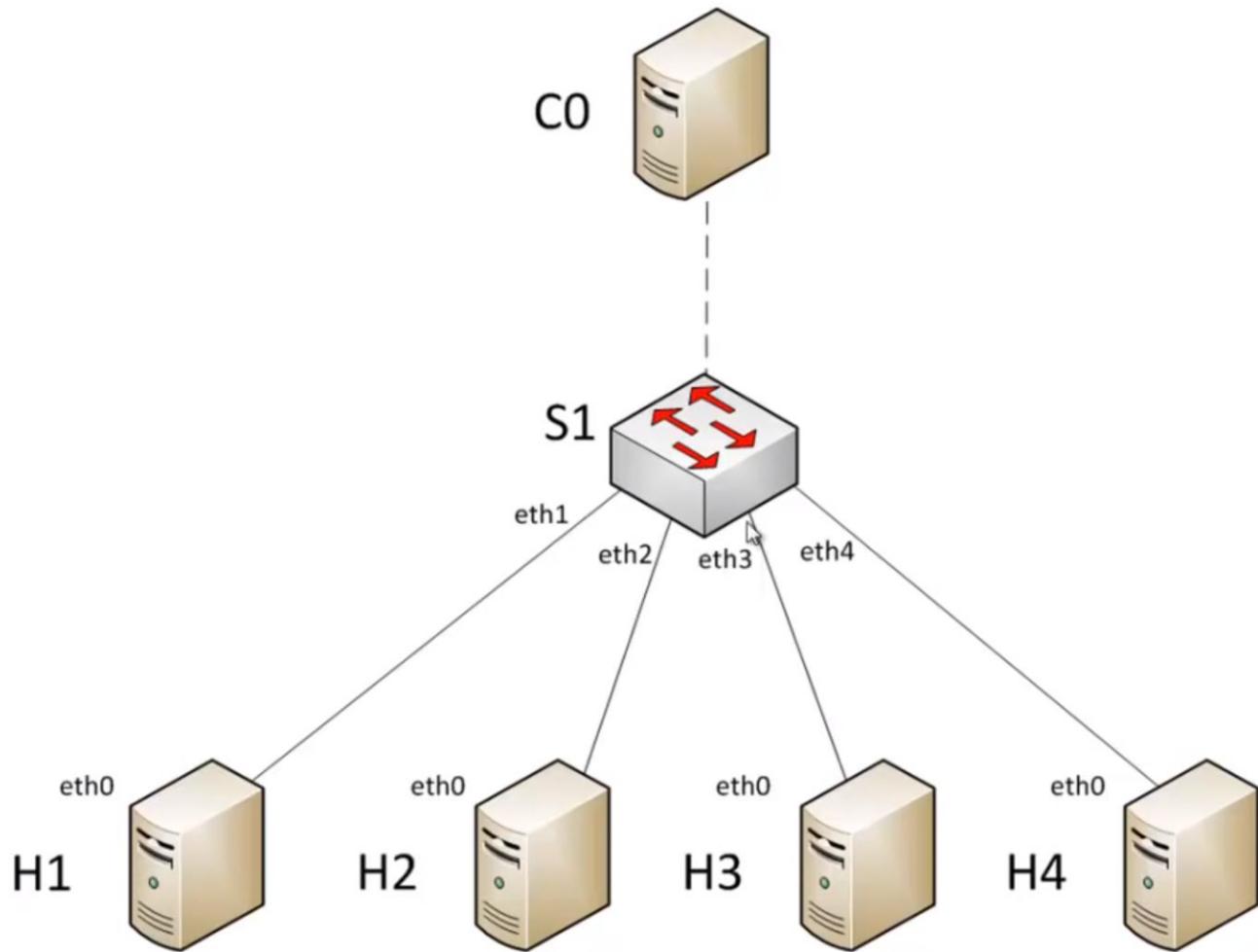
OpenFlow - Encaminhamento de pacotes

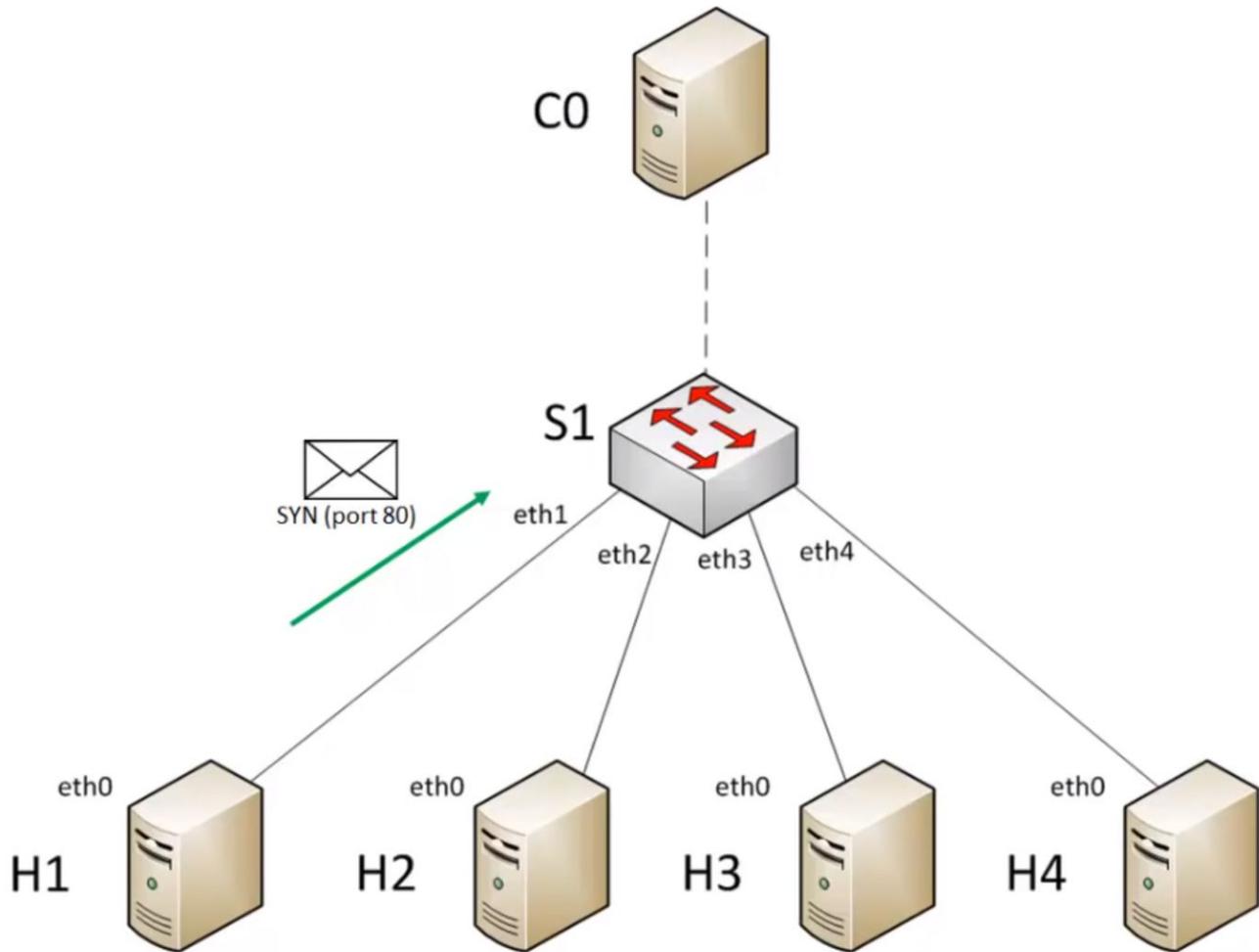
1. LOCAL
2. ALL
3. CONTROLLER
4. IN_PORT
5. TABLE

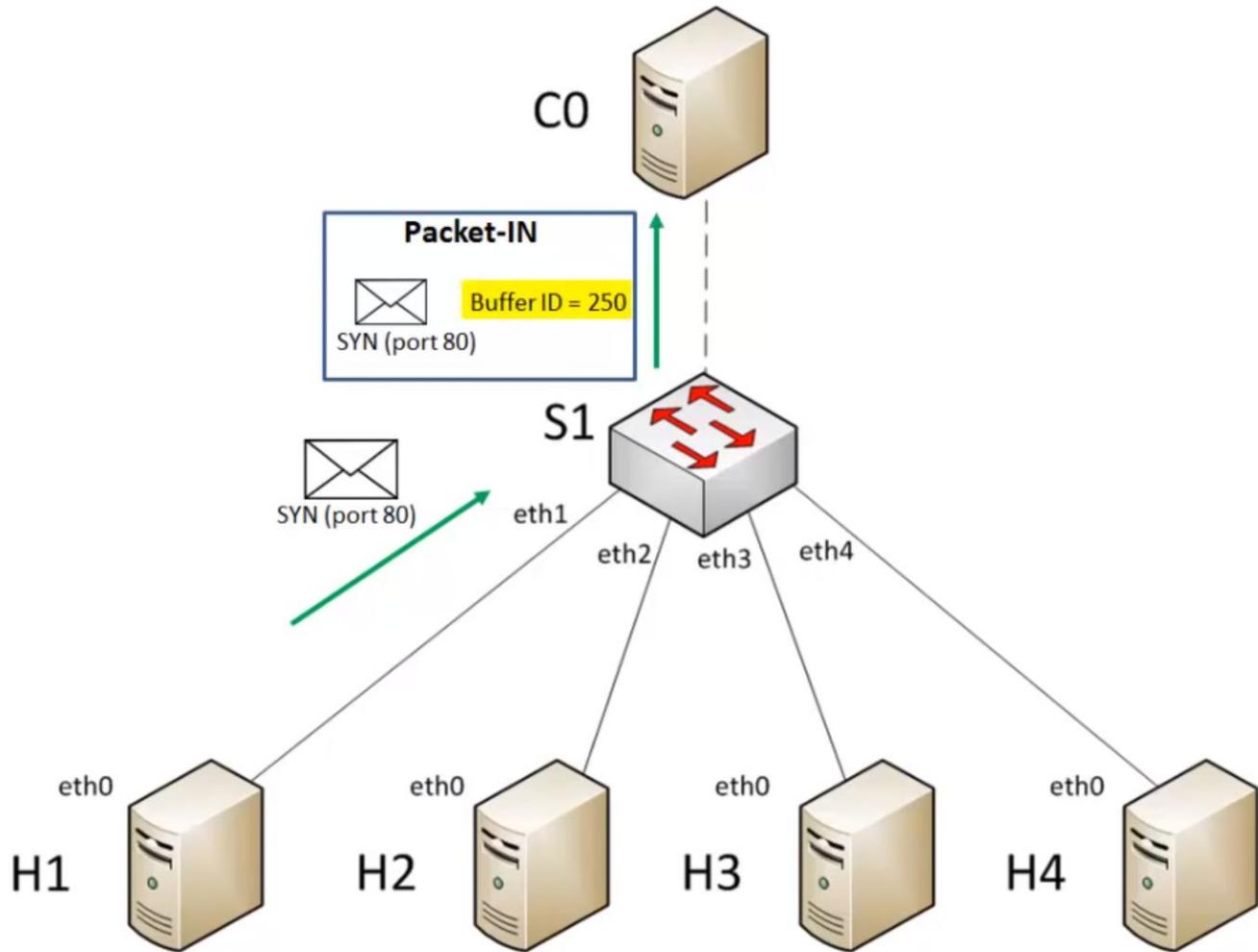
OpenFlow - Mensagens

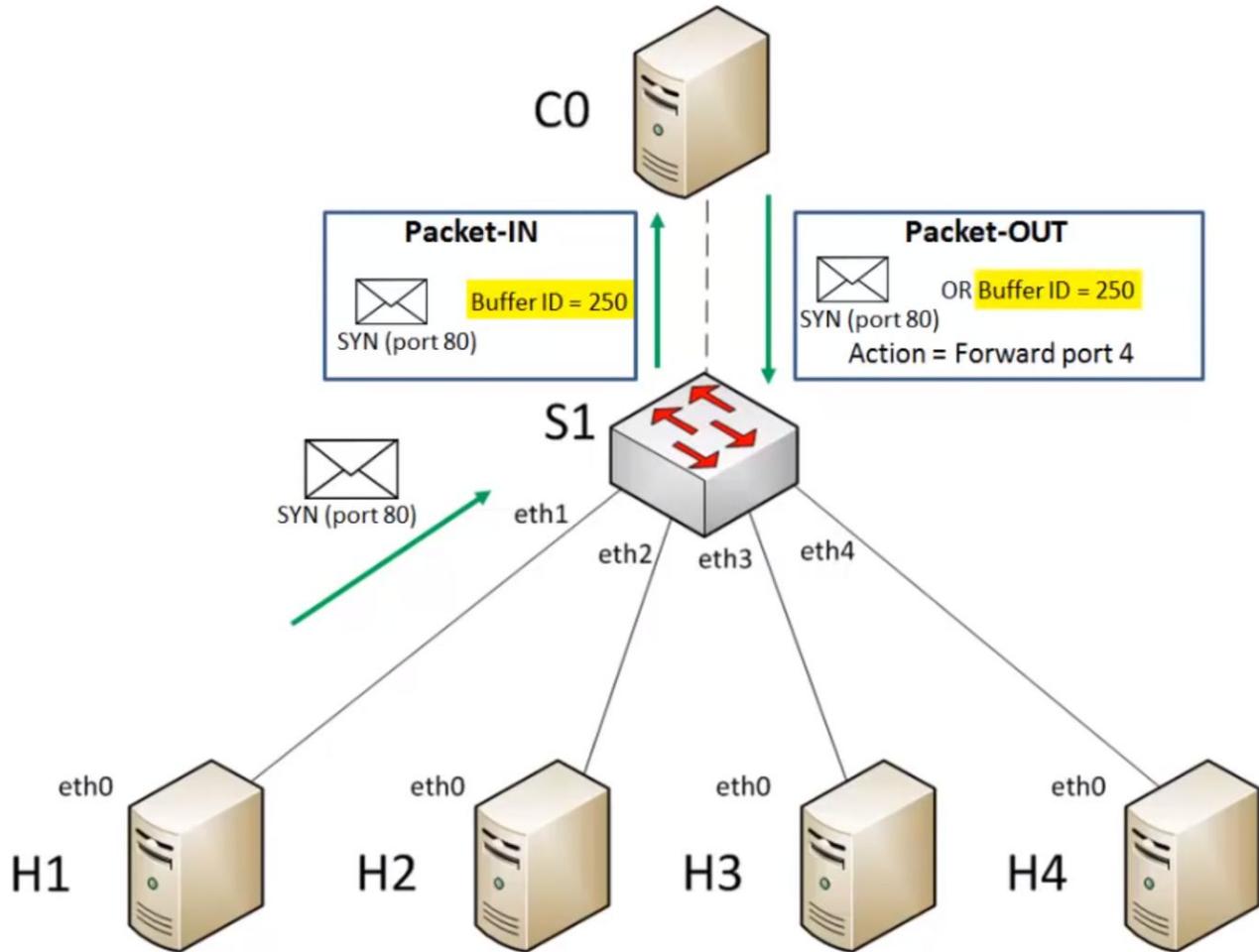
1. Symmetric (Both directions)
 - a. Hello (Handshake)
 - b. Echo (Keepalive)
 - c. Error (Notify)
 - d. Experimenter (Future proof)
2. Asynchronous (From switch)
 - a. Packet-in (Table-miss) -> Packet-out/Flow-mod
 - b. Flow-removed
 - c. Flow-monitor
 - d. Port-status, Role-status, Controller-Status, ...
3. Controller-Switch:
 - a. Features
 - b. Configuration
 - c. Modify-State, Read-State, Packet-out, Barrier, ...

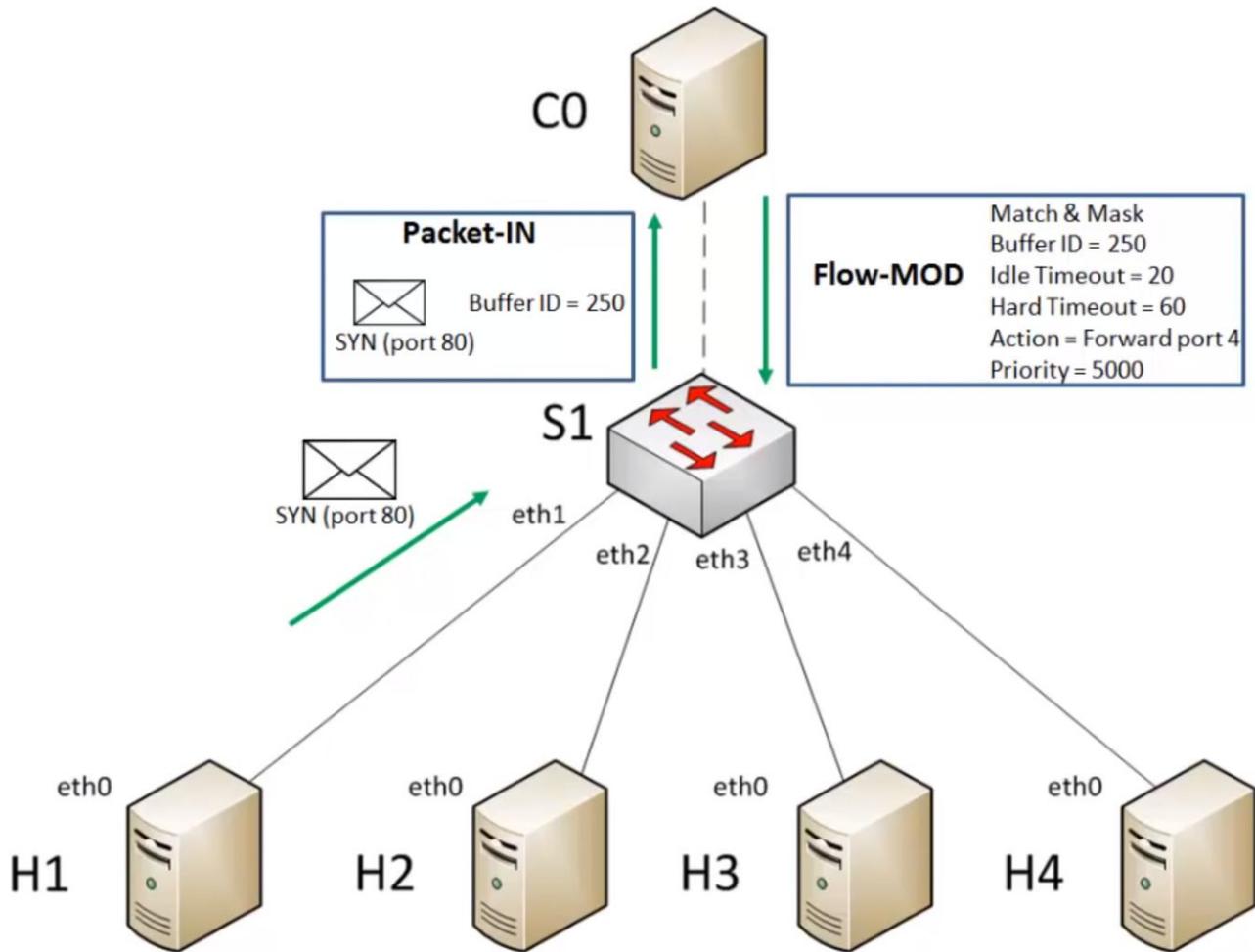
<https://www.youtube.com/watch?v=l25Ukkmk6Sk>

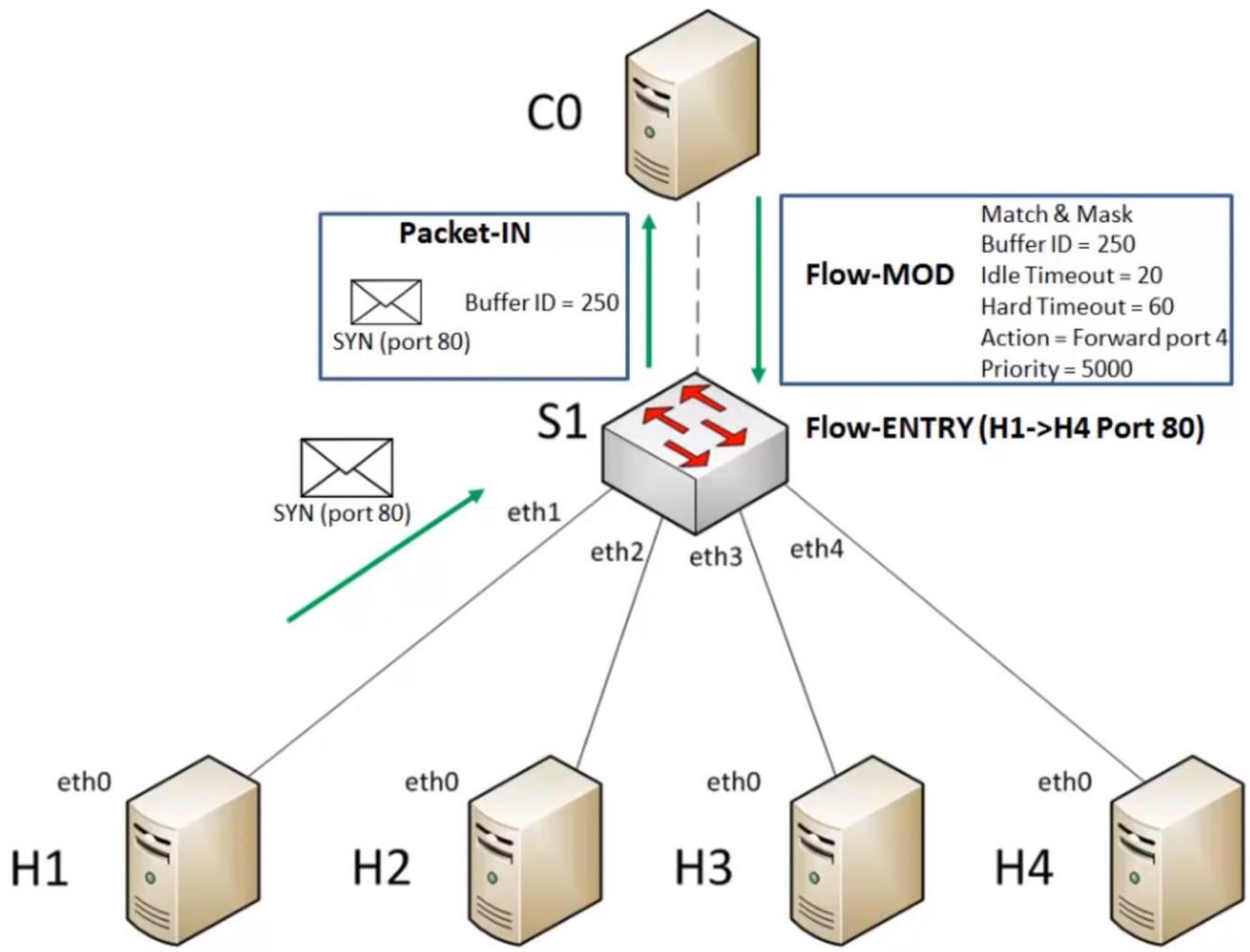


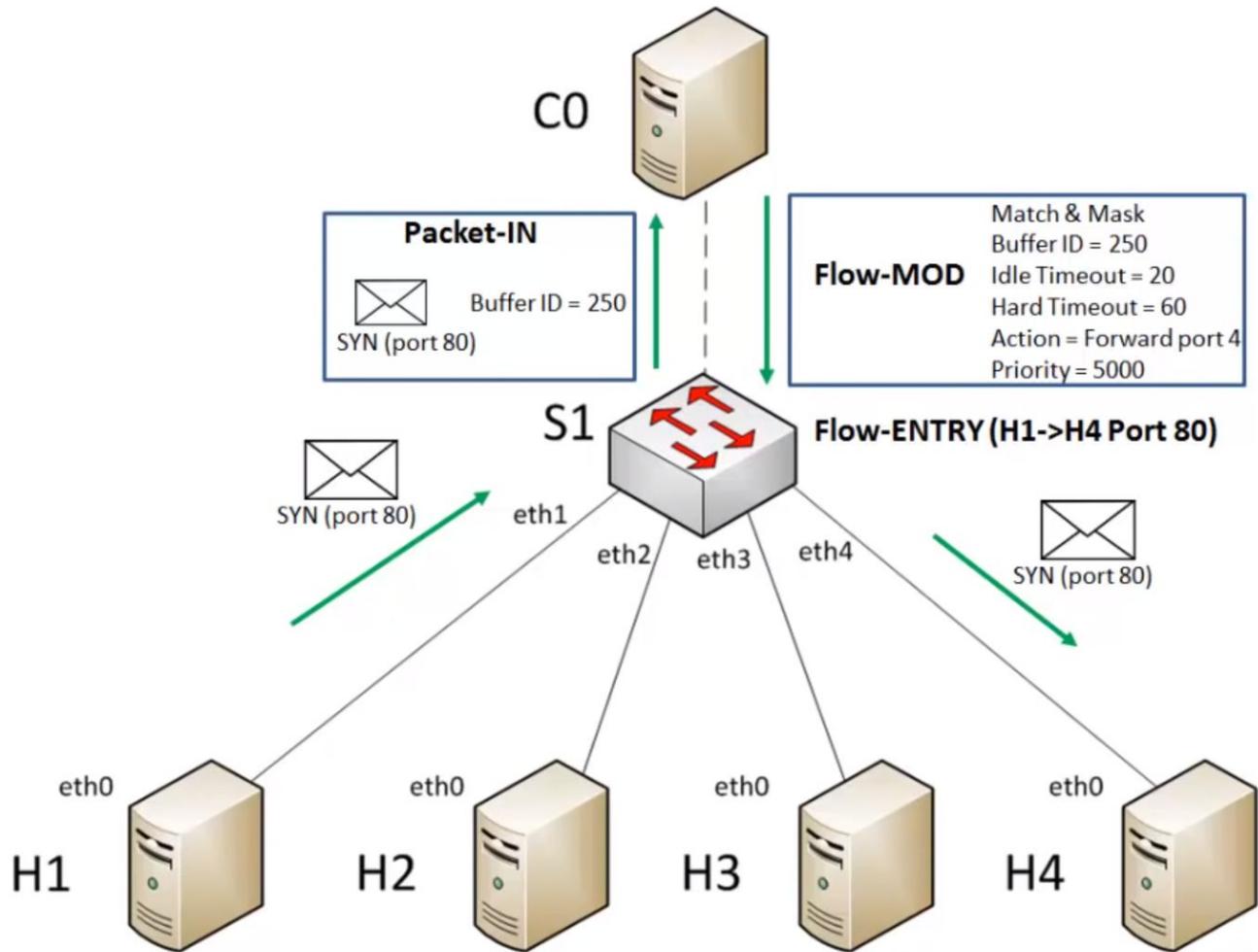


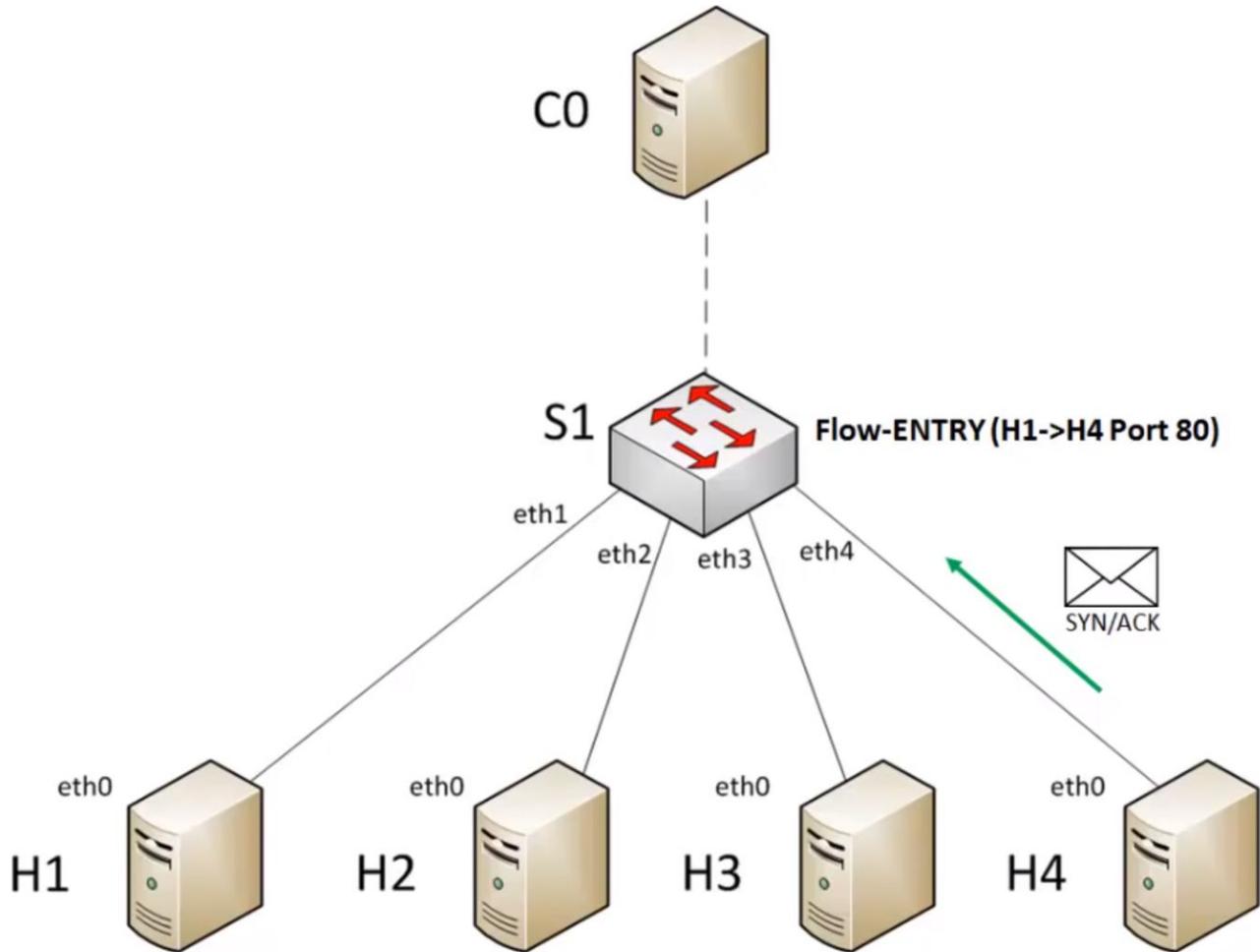


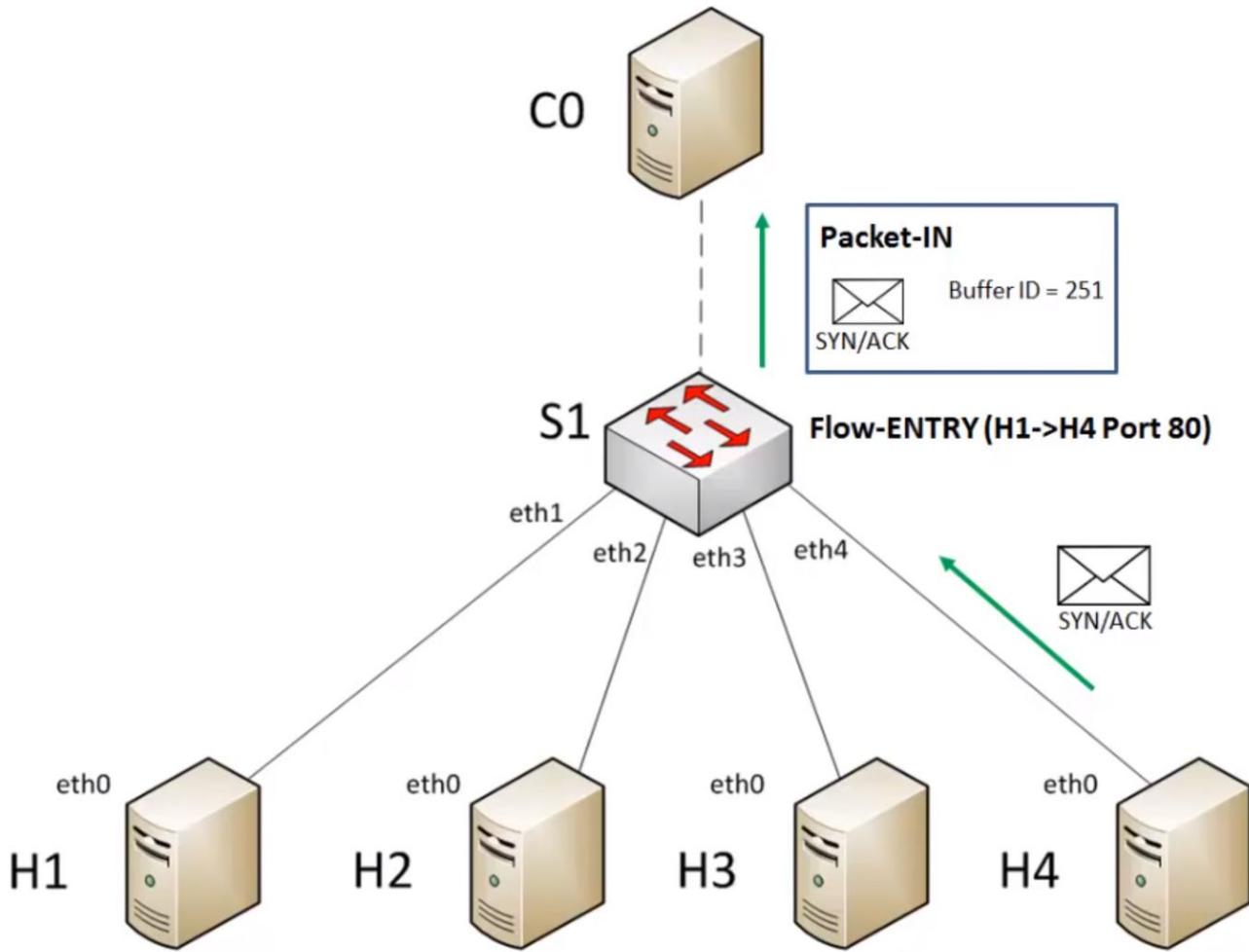


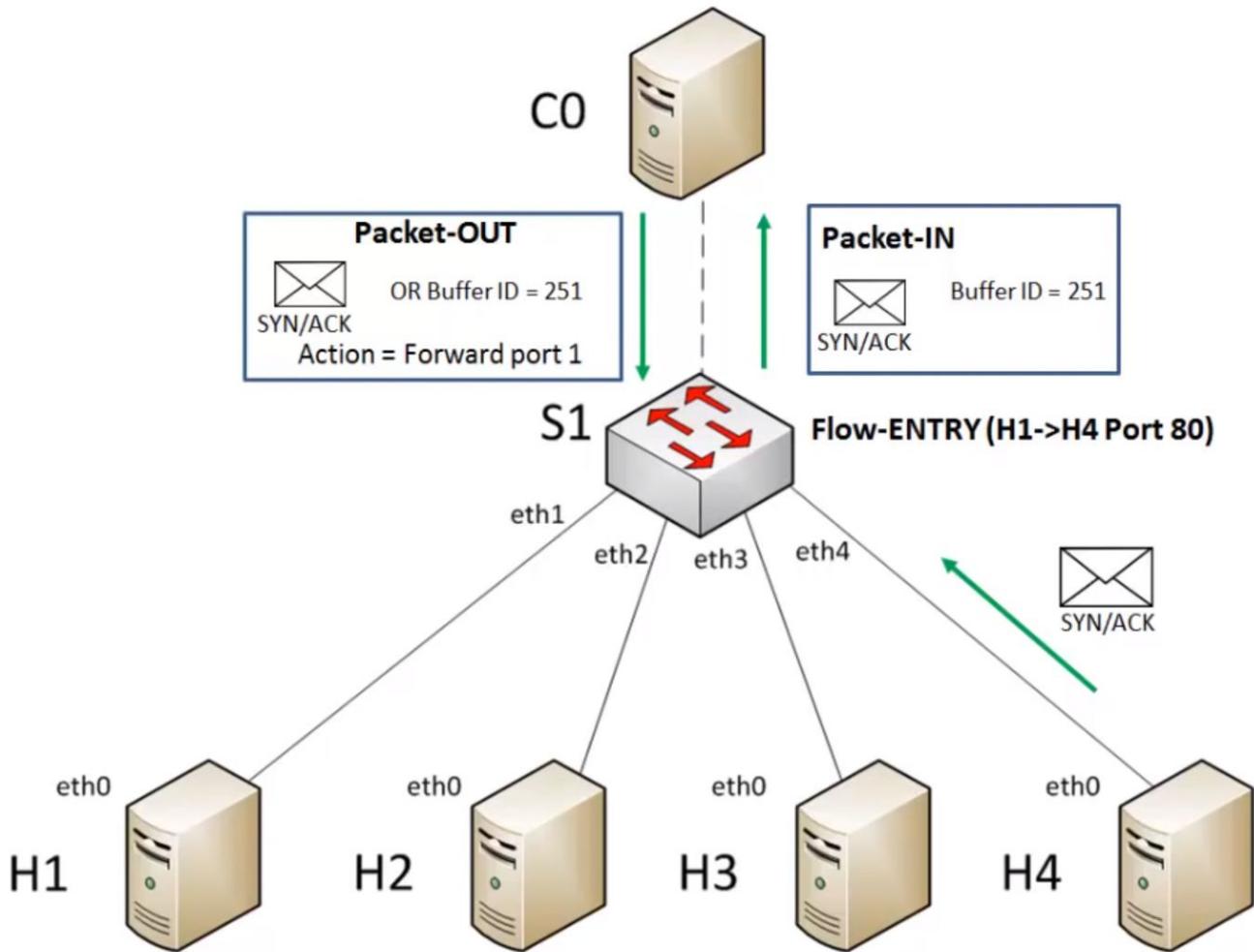


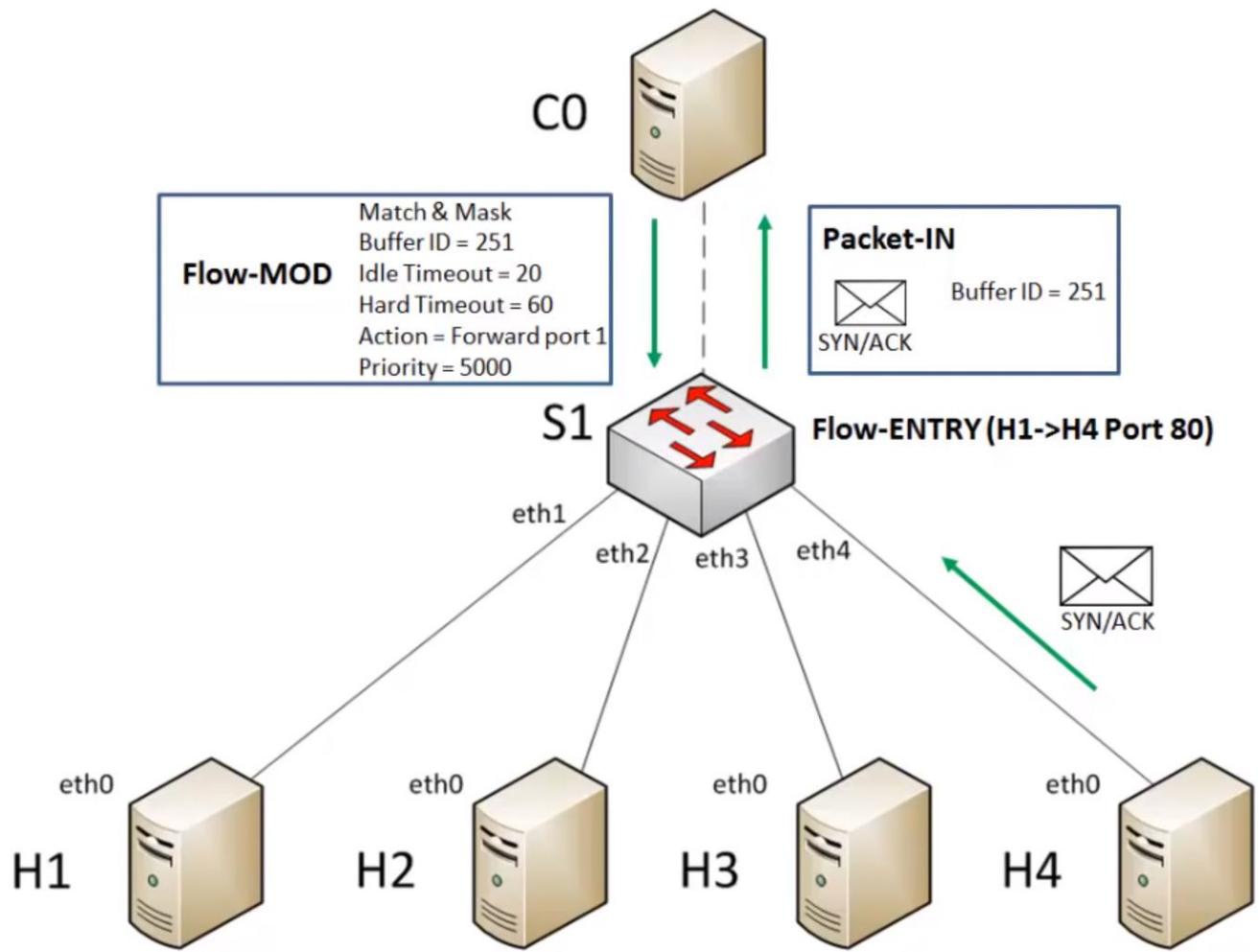


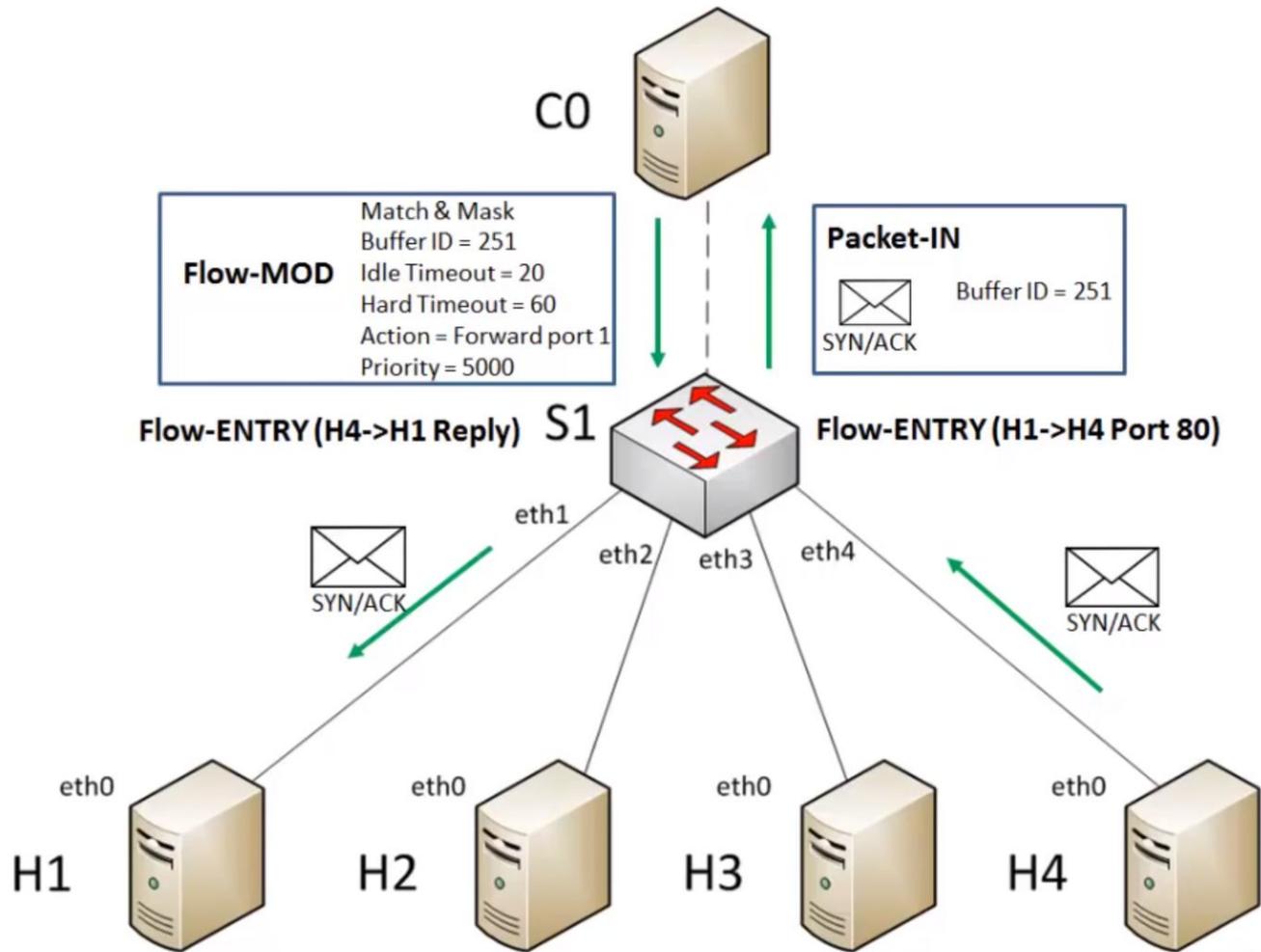


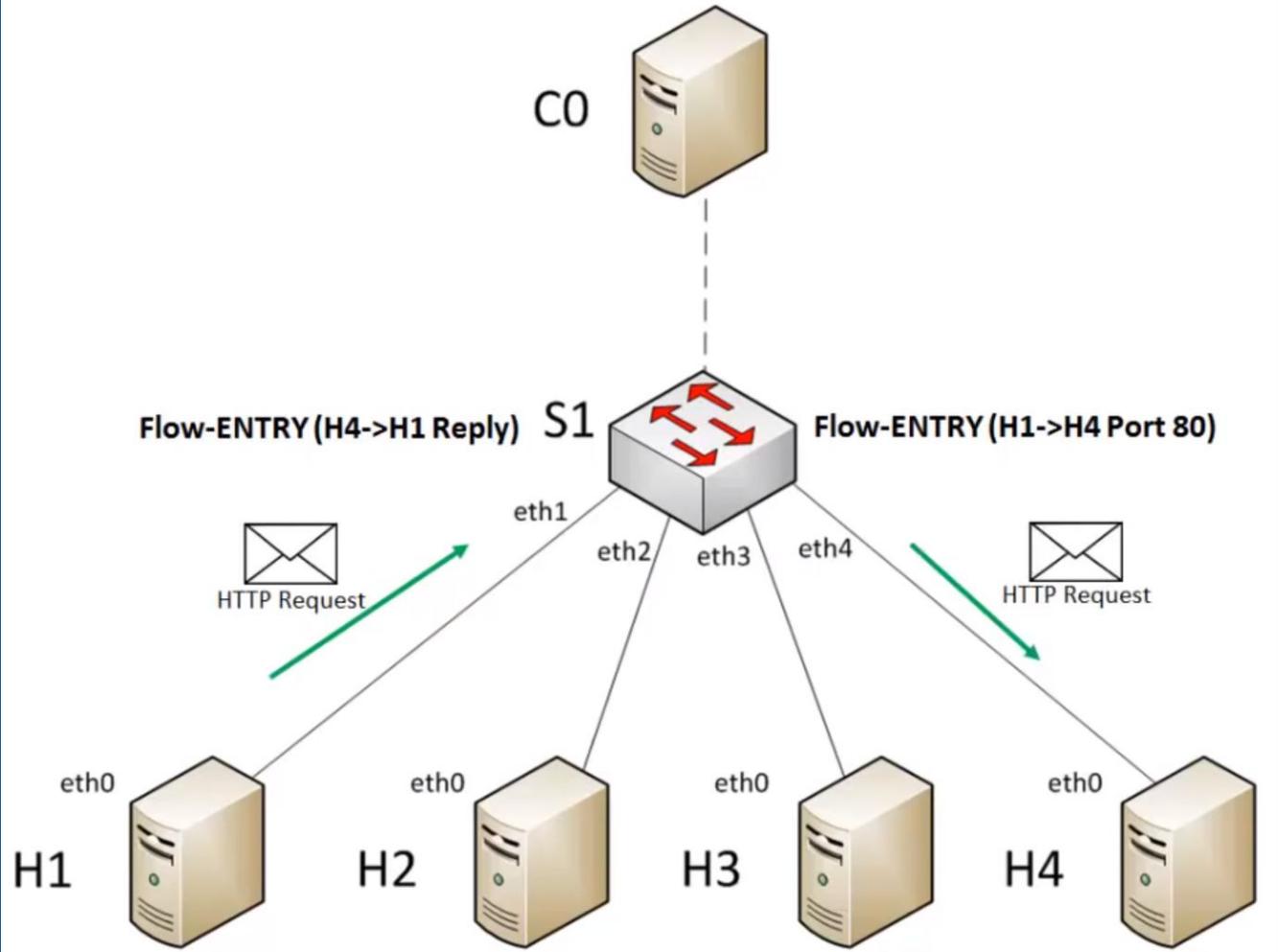


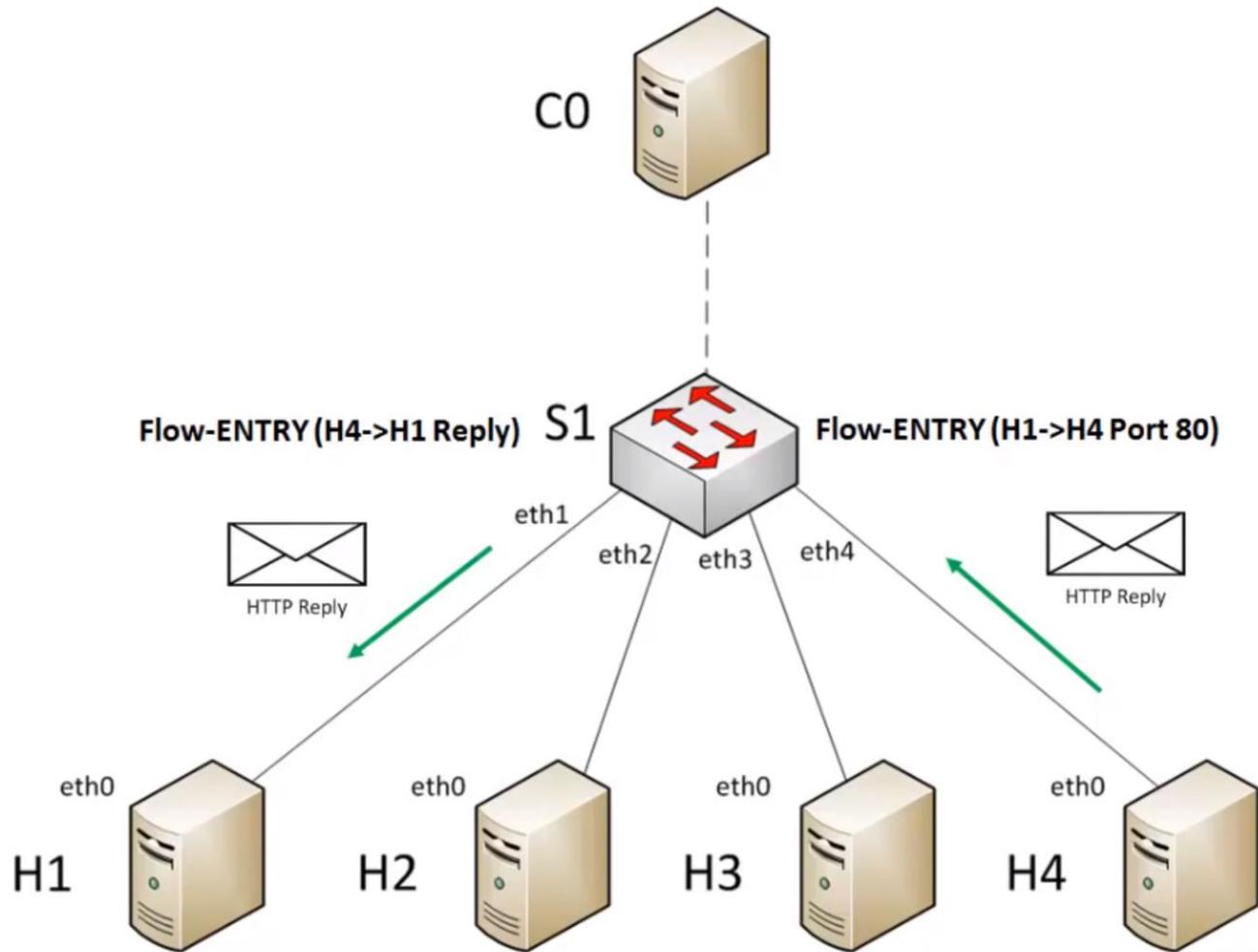












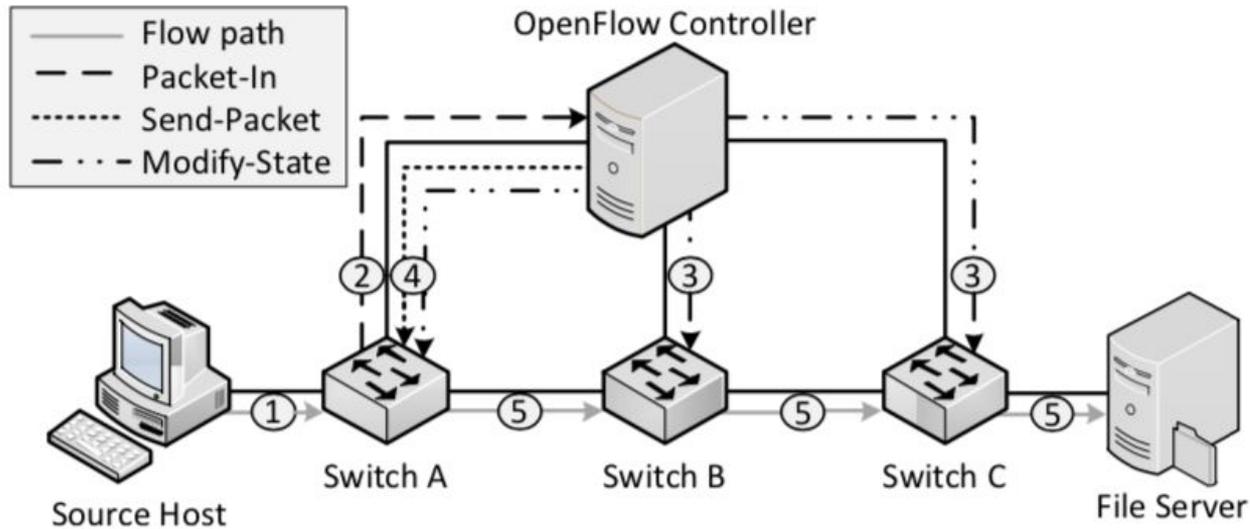


Fig. 1. OpenFlow controller *Forwarding* behavior example

Fonte da imagem: Ref. 6, Isolani et al

Numa configuração com vários switches, o Modify é repassado a todos os switches controlados (indicado por meio do 3 e 4 da figura).

Tendências

Network configuration - NetConf

- Protocolo para configuração de elementos de redes
- RPC (Remote Procedure Call)
- Yang models
- XML

RestConf

- Requisições HTTP
- Yang models
- XML ou JSON

Modelo Yang

- Modelagem de dados

```
module ietf-interfaces {
  import ietf-yang-types {
    prefix yang;
  }
  container interfaces {
    list interface {
      key "name";
      leaf name {
        type string;
      }
      leaf enabled {
        type boolean;
        default "true";
      }
    }
  }
}
```

Conclusões

Conclusões

SDN permitem experimentos em rede de forma rápida e independente da tecnologia proprietária dos dispositivos.

Possuem baixo custo e interesse de diversos fabricantes de switch.

Como desafios, existe a dependência em relação aos fabricantes para que eles exponham API's em seus produtos. E diferentemente das API's southbound que possuem um protocolo padronizado, ainda não existe um protocolo aceito como padrão com API northbound.

Perguntas?

Pergunta:

Baseado no artigo OpenFlow: Enabling innovation in campus networks, dê dois motivos pelos quais SDN e OpenFlow tem obtido sucesso.

https://www.researchgate.net/publication/220195143_OpenFlow_Enabling_innovation_in_campus_networks

Referências

1. Software Defined Networks, a Comprehensive Approach. 1º Edição - Paul Goransson, Chuck Black
2. The Road to SDN. ACM Queue - Nick Feamster, Jennifer Rexford, Ellen Zegura.
3. OpenFlow: Enabling innovation in campus networks - McKeown et al.
4. <https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html>
5. <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>
6. Interactive Monitoring, Visualization, and Configuration of OpenFlow-Based SDN - Isolani et al.