muHỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
****

TRẦN QUỐC TRUNG

GIẢI PHÁP CHỐNG TẤN CÔNG TRONG MẠNG ĐỊNH NGHĨA BẰNG PHẦN MỀM

LUẬN VĂN THẠC SỸ

Hà Nội-2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

****

TRẦN QUỐC TRUNG

GIẢI PHÁP CHỐNG TẤN CÔNG TRONG MẠNG ĐỊNH NGHĨA BẰNG PHẦN MỀM

LUẬN VĂN THẠC SỸ

Chuyên ngành : Kỹ thuật viễn thông
Mã số : 8.52.02.08

Người hướng dẫn khoa học: TS. Ngô Đức Thiện

Hà Nội- 2019

LỜI CAM ĐOAN

Tôi xin cam đoan nội dung luận văn của tôi là do sự tìm hiểu và nghiên cứu của bản thân. Các kết quả nghiên cứu cũng như ý tưởng của các tác giả khác đều được trích dẫn cụ thể.

Đề tài luận văn của tôi chưa được bảo vệ tại bất kỳ một hội đồng bảo vệ luận văn thạc sĩ nào trong nước và nước ngoài. Đồng thời cho đến nay chưa được công bố trên bất kỳ phương tiện thông tin truyền thông nào.

Tác giả luận văn

TRẦN QUỐC TRUNG

LỜI CẢM ƠN

Tôi xin cảm ơn TS. Ngô Đức Thiện và các thầy cô Khoa Đào Tạo Sau Đại Học đã tận tình hướng dẫn và giúp đỡ để tôi có thể hoàn thành tốt đề tài này. Do kinh nghiệm và kiến thức còn chưa được sâu sắc nên luận văn còn nhiều thiếu sót, mong quý thầy cô đánh giá và góp ý để tôi có thể hoàn thiện tốt hơn luận văn này cũng như các đề tài nghiên cứu sau này!

Xin chân thành cảm ơn!

Hà nội, ngày 16 tháng 01 năm 2020

Tác giả luận văn

TRẦN QUỐC TRUNG

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

Chữ viết tắt	Tiếng anh	Tiếng Việt
DoS	Distributed Denial of Service	Từ chối dịch vụ
DNS	Domain Name System	Hệ thống phân giải tên miền
TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
UDP	User Datagram Protocol	Giao thức gói tin người dùng
SDN	Software Defined Networking	Mạng định nghĩa bằng phần mềm
ID	Identification	Địa chỉ mạng
IP	Internet Protocol	Giao thức Internet
API	Application Programming Interface	Giao diện lập trình ứng dụng
CPU	Center Processing Unit	Bộ xử lý trung tâm
TTL	Time To Live	Thời gian cập nhật
TLS	Transport Layer Security	Giao thức bảo mật
SSL	Secure Sockets Layer	Lớp cổng bảo mật
FPGA	Field Progammable Gate Array	Cấu trúc mảng phần tử logic lập trình được
MAC	Media Access Control	Kiểm soát truy cập
CapEx	Capital Expenditures	Chi phí triển khai
OpEx	Operating Expenditures	Chi phí hoạt động

DANH MỤC HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

Với sự phát triển không ngừng của Internet ngày nay, nhu cầu mở rộng mạng ngày càng tăng, đòi hỏi về số lượng các thiết bị mạng ngày càng lớn, từ đó kiến trúc mạng truyền thống đã bộc lộ ra nhiều khuyết điểm. Sự phức tạp trong hệ thống, khả năng mở rộng mạng kém, chính sách không nhất quán, chi phí triển khai tốn kém, nhiều điểm yếu về bảo mật.

Nhu cầu đặt ra cần có một kiến trúc mạng đảm bảo được sự tích hợp linh hoạt, kết hợp với các giải pháp bảo mật an toàn cho hệ thống. Chính vì vậy, công nghệ mạng định nghĩa bằng phần mềm (SDN) ra đời như một giải pháp cho hệ thống mạng trong tương lai. Bên cạnh đó, SDN còn là lựa chọn cho việc triển khai các giải pháp đảm bảo an ninh mạng, trước sự phức tạp của những cuộc tấn công không ngừng thay đổi về cách thức cũng như độ nguy hại, cản trở nhiều hoạt động giao dịch, các dịch vụ mạng.

Những hạn chế về bảo mật của kiến trúc mạng truyền thống đã để lộ ra những lỗ hổng cho kẻ tấn công, những tổ chức tội phạm có thể thực hiện hành vi phá hoại tới những hệ thống, gây hậu quả cho các doanh nghiệp, cơ quan, tổ chức, các nhà cung cấp dịch vụ. Vì lý do đó em xin chọn đề tài "*Giải pháp chống tấn công trong mạng định nghĩa bằng phần mềm*" làm đề tài luận văn tốt nghiệp.

2. Tổng quan về vấn đề nghiên cứu

SDN ra đời vào năm 2008 tại Đại học Stanford và tạo ra một cuộc cách mạng trong giới công nghệ. Ưu điểm của SDN là việc tách phần logic điều khiển mạng khỏi các bộ chuyển mạch, thúc đẩy điều khiển tập trung và cung cấp khả năng lập trình cho mạng. Hiện tại, Google và Facebook đều đầu tư rất mạnh cho SDN và dự đoán trong 5 năm tới sẽ thay thế toàn bộ mạng truyền thống.

Vấn đề nghiên cứu về mạng SDN được nghiên cứu rộng rãi trên thế giới trong những năm gần đây. Từ nhiều năm trở lại đây, đã có các bài báo trên thế giới đã đưa ra rất nhiều giải pháp nhằm nâng cao hiệu quả cho quá trình sử dụng mạng Internet.

Trong đó SDN là một trong những giải pháp được kỳ vọng cao. Tại Việt Nam, đã có các công trình nghiên cứu và áp dụng SDN vào việc thiết kế và áp dụng cho việc điều khiển mạng.

Luận văn sẽ nghiên cứu về công nghệ SDN/OpenFlow, các ưu điểm mà SDN cung cấp so với cấu trúc mạng truyền thống. Bên cạnh đó tìm hiểu các hình thức tấn công SDN và cách phòng chống các hình thức tấn công này. Từ đó đưa ra phương thức phòng chống tấn công dựa trên công nghệ SDN/OpenFlow.

3. Mục đích nghiên cứu

Nghiên cứu các phương pháp phòng chống tấn công SDN và áp dụng công nghệ SDN/OpenFlow vào việc phòng chống tấn công, nâng cao bảo mật.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng: Công nghệ mạng định nghĩa bằng nội dung (SDN).

Phạm vi: Phòng chống tấn công trong SDN dựa trên SDN/OpenFlow.

5. Phương pháp nghiên cứu

Nghiên cứu tìm hiểu lý thuyết từ các tài liệu, bài báo, công trình nghiên cứu về SDN và tấn công trong SDN. Xây dựng một kiến trúc mạng sử dụng trong phòng chống tấn công SDN, tiến hành mô phỏng tấn công trên server testbed của Mobifone, sử dụng các card phần cứng và phầm mềm, công cụ hỗ trợ. Dựa trên kết quả mô phỏng đưa ra giải pháp giảm thiểu các tấn công vào SDN.

6. Nội dung đề tài

Nội dung của luận văn bao gồm 3 chương với cấu trúc như sau:

Chương 1. Tổng quan về SDN:

Chương 2. Xây dựng kiến trúc mạng SDN/OpenFlow sử dụng trong phòng chống tấn công

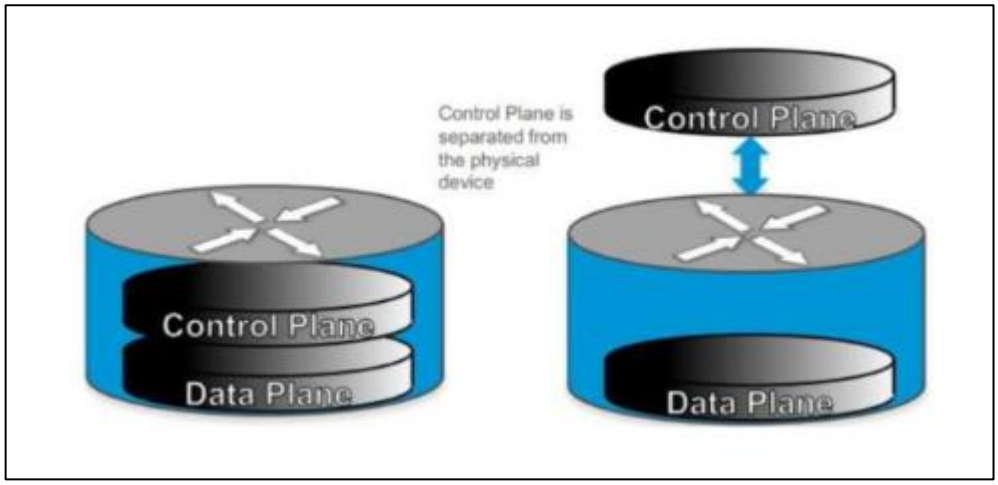
Chương 3. Kết quả mô phỏng chống tấn công trong SDN

CHƯƠNG 1. TỔNG QUAN VỀ SDN

1.1. Tổng quan về SDN

1.1.1. Định nghĩa

Software-Definded Networking (SDN) là một cách tiếp cận mới trong việc thiết kế, xây dựng và quản lý hệ thống mạng. Về cơ bản, SDN chia tách độc lập hai cơ chế hiện đang tồn tại trong cùng một thiết bị mạng: Cơ chế điều khiển (Control Plane *controller – thành phần điều khiển*), cơ chế chuyển tiếp dữ liệu (Data Plane - data forwarding plane – thành phần chuyển tiếp dữ liệu) nhằm tối ưu nhiệm vụ và chức năng của hai thành phần này (Hình 1.1). Mục đích của sự phân tách này là tạo ra mạng có thể được lập trình và quản lý một cách tập trung.



Hình 1.1. Sự phân tách trong kiến trúc mạng SDN

Cơ chế điều khiển (*Control Plane*): Là thành phần điều khiển với các giải thuật và giao thức dùng để tính toán và lưu trữ các thông tin định tuyến lên bảng FIB (Forwarding Information Base) nhằm xác định đường đi cho mỗi gói tin trong Data Plane. Đối với Switch thì Control Plane đơn giản là cơ chế tự học MAC thông qua việc Broadcast gói tin còn đối với Router thì Control Plan là các giao thức định tuyến như OSPF, EIGRP, BGP, …

Cơ chế chuyển tiếp dữ liệu (*Data Plane*): Là thành phần thực hiện chức năng Forwarding Data dựa vào bảng FIB mà Control Plane đã xây dựng. Dữ liệu sẽ đổ về

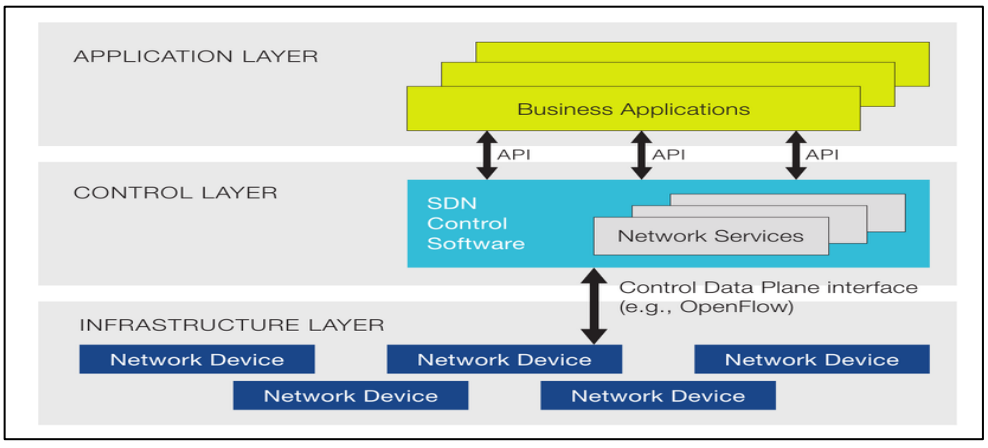
Switch hoặc Router tại các Port tương ứng như 10Gb, 100Gb Ethernet và cũng đi ra khỏi Switch, Router với Port tương ứng.

SDN là một kiến trúc mạng linh hoạt, dễ quản lý, hiệu suất cao, khả năng chịu lỗi và thích nghi tốt,… Điều đó làm cho công nghệ này thật sự lý tưởng cho các ứng dụng đòi hỏi băng thông cao và cần sự linh hoạt hiện nay. Mục đích cơ bản của truyền thông mạng là truyền tải thông tin từ điểm này tới các điểm khác nhưng với SDN thì dữ liệu trong mạng sẽ được truyền tải giữa các node với sự hỗ trợ từ các ứng dụng và dịch vụ nên việc truyền thông trở nên hiệu quả và tối ưu hơn rất nhiều.

Giống như các máy chủ giám sát sử dụng trong ảo hóa, SDN định nghĩa ra một lớp phần mềm đứng chặn giữa các phần tử mạng và người quản trị mạng (là người cấu hình và cài đặt chúng). Lớp phần mềm này cung cấp cho người quản trị mạng khả năng điều khiển các thiết bị mạng của họ thông qua một giao diện phần mềm thay vì phải tự cấu hình phần cứng và các tác động vật lý của thiết bị mạng..

1.1.2. Kiến trúc của SDN [2]

Kiến trúc của SDN gồm 3 lớp riêng biệt: lớp ứng dụng, lớp điều khiển, và lớp cơ sở hạ tầng (lớp chuyển tiếp). (Hình 1.2)



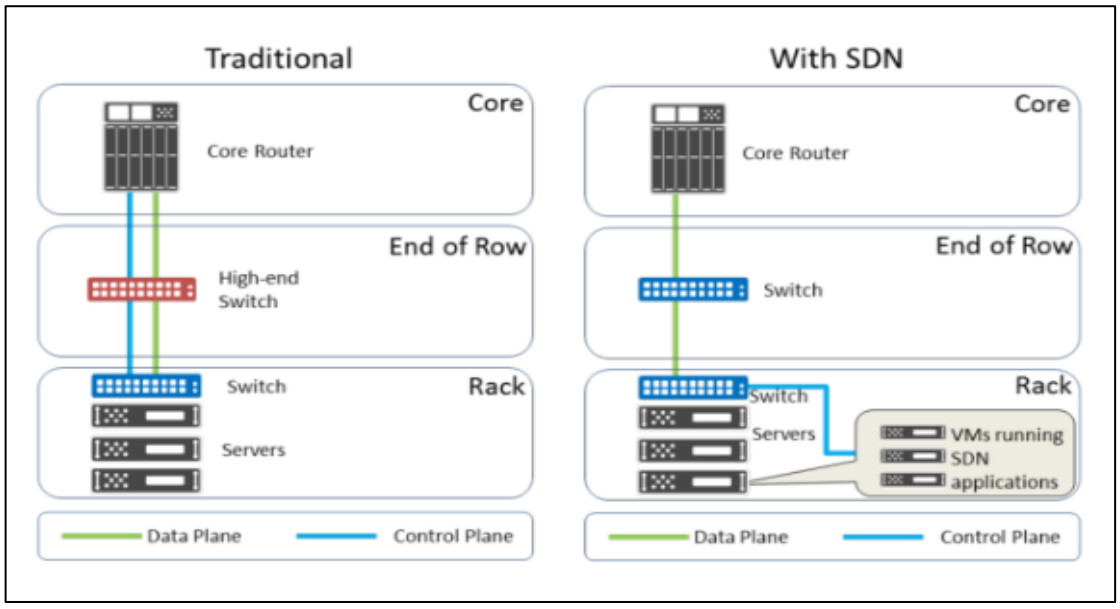
Hình 1.2. Kiến trúc mạng SDN

Lớp ứng dụng: Là các ứng dụng kinh doanh được triển khai trên mạng, được kết nối tới lớp điều khiển thông qua các API, cung cấp khả năng cho phép lớp ứng dụng lập trình lại (cấu hình lại) mạng (điều chỉnh các tham số trễ, băng thông, định tuyến, …) thông qua lớp điều khiển.

Lớp điều khiển: Là nơi tập trung các bộ điều khiển thực hiện việc điều khiển cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng. Các bộ điều khiển này có thể là các phần mềm được lập trình.

Lớp cơ sở hạ tầng: Là các thiết bị mạng thực tế (vật lý hay ảo hóa) thực hiện việc chuyển tiếp gói tin theo sự điều khiển của lớp điều khiển. Một thiết bị mạng có thể hoạt động theo sự điều khiển của nhiều bộ điều khiển khác nhau, điều này giúp tăng cường khả năng ảo hóa của mạng.

1.1.3. So sánh kiến trúc mạng truyền thống và kiến trúc SDN



Hình 1.3. So sánh mạng SDN với mạng truyền thống

Mô hình so sánh giữa kiến trúc mạng truyền thống và kiến trúc SDN trên (hình 1.3) cho thấy trong kiến trúc mạng truyền thống Control Plane và Data Plane đều được ghép chung vào trong Network Node. Trong đó Control Plane có nhiệm vụ cấu hình các Node mạng và lập trình đường đi (định tuyến) để vận chuyển Data Flow. Data Flow (luồng dữ liệu) sẽ được đẩy xuống Data Plane thông qua các API và chuyển tiếp tới các thiết bị phần cứng dựa trên các thông tin điều khiển trên Control Plane.

Trong kiến trúc mạng truyền thống khi chính sách Forwarding đã được thông qua thì cách duy nhất để điều chỉnh lại các chính sách này theo ý muốn là phải đi

cấu hình lại trên tất cả thiết bị vật lý (Switch, Router, Firewall, …). Điều này không chỉ mất thời gian mà còn khá là phiền toái bởi trong kiến trúc hệ thống mạng truyền thống đặc biệt đối với hệ thống mạng quy mô trong doanh nghiệp thì việc xác định vị trí thiết bị và tiến hành cấu hình điều chỉnh rất là phức tạp và có nhiều rủi ro sai sót có thể ảnh hưởng tới nhiều hoạt động quan trọng khác trong hệ thống mạng.

Trong kiến trúc mạng SDN thì Control Plane được tách riêng ra khỏi Node mạng và là một thành phần độc lập trong SDN Stack. Các bộ chuyển mạch SDN đều được kiểm soát bởi Network Operating System (NOS) để thu thập thông tin thông qua các API và chuyển tiếp thông tin ấy vào Control Plane, đồng thời cung cấp một mô hình mạng trừu tượng tới SDN Controller vốn được cài đặt trong các ứng dụng. Nhờ thế mà Controller có thể khai thác đầy đủ thông tin mạng để tối ưu trình điều khiển luồng (Flow Management) và hỗ trợ đáp ứng các yêu cầu của người dùng về khả năng mở rộng và tính linh hoạt. Việc quản lý, cấu hình và điều phối các hoạt động trong hệ thống mạng sẽ thông qua một giao diện phần mềm quản lý tập trung với mô hình hệ thống mạng tổng quan nên người quản trị sẽ làm việc hiệu quả hơn và tránh được các sự cố lỗi phát sinh.

Kiến trúc mạng SDN cho phép điều khiển hoạt động của mạng dựa trên việc điều khiển hoạt động của các Switch, sử dụng các ngôn ngữ lập trình và có thể thường xuyên thay đổi bằng cách nạp các đoạn mã nguồn khác vào thay thế. SDN có lợi thế to lớn mà các kiến trúc mạng hiện tại không có được. Đó chính là tính mềm dẻo, khả năng mở rộng và tính linh hoạt của mạng. Các kiến trúc mạng hiện nay không cho phép can thiệp vào hoạt động của các thiết bị mạng, hoạt động của các thiết bị này bị phụ thuộc hoàn toàn vào nhà sản xuất quy định. Chính vì thế, rất khó cho việc thay thế một giao thức cũ bằng các giao thức mới và thử nghiệm chúng trên các thiết bị thật. SDN thì ngược lại, nó cho phép người lập trình có thể can thiệp vào hoạt động của thiết bị và qua đó điều khiển thiết bị hoạt động theo ý muốn. Việc này dễ dàng cho việc nghiên cứu cũng như triển khai các công nghệ mới vào trong mạng mà không cần thay đổi về phần cứng.

1.1.4. Lợi ích của SDN

SDN là một sản phẩm mã nguồn mở. Bởi vì SDN tuân thủ các chuẩn mở, về mặt lý thuyết có thể hoạt động với bất kỳ phần cứng mạng nào của nhà cung cấp. Từ quan điểm CNTT, điều này cho phép các tổ chức có khả năng tránh việc nhà cung cấp cứng nhắc trong một loạt sản phẩm mạng. Điều này cho phép CNTT trở lên nhanh nhẹn rất lớn bởi vì một giải pháp chuẩn mở như SDN đơn giản hóa nhiệm vụ kết nối đến các đám mây, các ứng dụng, và các thiết bị khác nhau. Và nó cho phép người quản trị mạng sử dụng phần mềm cho nhiều công việc họ thường làm bằng tay.

Cụ thể, với các tính năng của mình, SDN đem lại các lợi ích sau:

- Giảm CapEx: SDN giúp giảm thiểu các yêu cầu mua phần cứng theo mục đích xây dựng các dịch vụ, phần cứng mạng trên cơ sở ASIC, và hỗ trợ mô hình pay-as-you-grow (trả những gì bạn dùng) để loại bỏ lãng phí cho việc dự phòng.
- Giảm OpEx: thông qua các phần tử mạng đã được gia tăng khả năng lập trình, SDN giúp dễ dàng thiết kế, triển khai, quản lý và mở rộng mạng. Khả năng phối hợp và dự phòng tự động không những giảm thời gian quản lý tổng thể, mà còn giảm xác suất lỗi do con người tới việc tối ưu khả năng và độ tin cậy của dịch vụ.
- Truyền tải nhanh chóng và linh hoạt: giúp các tổ chức triển khai nhanh hơn các ứng dụng, các dịch vụ và cơ sở hạ tầng để nhanh chóng đạt được các mục tiêu kinh doanh.
- Cho phép thay đổi: cho phép các tổ chức tạo mới các kiểu ứng dụng, dịch vụ và mô hình kinh doanh, để có thể tạo ra các luồng doanh thu mới và nhiều giá trị hơn từ mạng.

SDN hứa hẹn khả năng ảo hóa hạ tầng CNTT. Cho đến nay, phần lớn nhất của cơ sở hạ tầng vẫn còn chưa được ảo hóa là mạng. Từ một quan điểm nhanh gọn, các doanh nghiệp sẽ muốn liên hiệp lại và di chuyển vào và ra khỏi đám mây riêng nội

bộ (private cloud) và công cộng (public cloud). Đây là nơi mà một công nghệ linh hoạt như SDN đặt dấu chấm hết. Chắc chắn, tất cả các hoạt động này nối với mạng và các sự cố có thể được thực hiện "bằng tay" như bây giờ. Nhưng trong tương lai, khung thời gian hạn hẹp của dự án sẽ thúc đẩy CNTT tìm cách hiệu quả hơn để cấu hình và quản lý mạng.

- Bên cạnh đó, SDN cũng tồn tại một số nhược điểm so với mạng truyền thống như:
- Vấn đề đầu tiên là bảo mật, các hacker có thể lợi dụng lỗ hổng phần mềm để tấn công. Nếu hacker có thể tấn công vào hệ thống, chúng có thể truy cập các thiết lập và thay đổi chúng bất cứ ở nơi đâu, tại thời điểm nào, và chúng có thể truy cập bất kỳ tập tin được mã hóa nào miễn là nó ở trong mạng. Đối với mạng truyền thống thì điều này không thẻ xảy ra bởi để có thể truy cập vào mạng ta phải có quyền truy cập vào phần cứng của nó. Hầu hết các doanh nghiệp chỉ cho phép một số cá nhân được quyền đó bởi vậy hệ thống sẽ an toàn và ít có khả năng bị truy cập bởi các hacker.
- Thứ hai đó là quá trình triển khai SDN không thể hoàn thiện trong chốc lát mà nó phải theo từng bước một. Chúng ta không thể một lúc thay thế toàn bộ các thiết bị hiện có thành OpenFlow switch được bởi vì điều đó rất tốn kém.
- Thứ ba, SDN là một kiến trúc mạng kiểu mới, các giao thức tương tác giữa các controller với nhau còn chưa được phát triển toàn diện nên việc phát triển SDN trên phạm vi toàn cầu vẫn còn nhiều hạn chế.

1.1.5. Ứng dụng của SDN

Với những lợi ích mà mình đem lại, SDN có thể triển khai trong phạm vi các doanh nghiệp (Enterprises) hoặc trong cả các nhà cung cấp hạ tầng và dịch vụ viễn thông để giải quyết các yêu cầu của các nhà cung cấp tại mỗi phân khúc thị trường.

- *Áp dụng trong mạng doanh nghiệp*: Mô hình tập trung, điều khiển và dự phòng tự động của SDN hỗ trợ việc hội tụ dữ liệu, voice, video, cũng như là

việc truy cập tại bất kỳ thời điểm nào, bất kỳ đâu. Điều này được thực hiện thông qua việc cho phép nhân viên IT thực thi chính sách nhất quán trên cả cơ sở hạ tầng không dây và có dây. Hơn nữa, SDN hỗ trợ việc quản lý và giám sát tự động tài nguyên mạng, xác định bằng các hồ sơ cá nhân và các yêu cầu của ứng dụng, để đảm bảo tối ưu trải nghiệm người dùng với khả năng của mạng.

- *Áp dụng trong Data Center (DC)*: Việc ảo hóa các thực thể mạng của kiến trúc SDN cho phép việc mở rộng trong DC, di cư tự động các máy ảo, tích hợp chặt chẽ hơn với kho lưu trữ, sử dụng server tốt hơn, sử dụng năng lượng thấp hơn, và tối ưu băng thông.
- *Áp dụng đối với dịch vụ Cloud*: Khi được sử dụng để hỗ trợ một môi trường đám mây riêng hoặc tích hợp, SDN cho phép các tài nguyên mạng được cấp phát theo phương thức linh hoạt cao, cho phép dự phòng nhanh các dịch vụ đám mây và hand off linh hoạt hơn với các nhà cung cấp đám mây bên ngoài. Với các công cụ để quản lý an toàn các mạng ảo của mình, các doanh nghiệp và các đơn vị kinh doanh sẽ tin vào các dịch vụ đám mây hơn.

1.2. Giao thức OpenFlow

1.2.1. Định nghĩa

Khái niệm SDN đặt ra 2 vấn đề khi triển khai thực tế:

- Cần phần có một kiến trúc logic chung cho tất cả các switch, router và các thiết bị mạng khác được quản lý bởi SDN Controller (bộ điều khiển mạng SDN). Kiến trúc này có thể được triển khai bằng nhiều cách khác nhau trên các thiết bị của các nhà cung cấp khác nhau và phụ thuộc vào nhiều loại thiết bị mạng, miễn là SDN controller thấy được chức năng chuyển mạch thống nhất.
- Một giao thức chuẩn, bảo mật để giao tiếp giữa SDN controller và các thiết bị mạng.

OpenFlow được đưa ra để giao quyết cả hai vấn để đó.

OpenFlow là giao thức chuẩn mở cho phép các nhà nghiên cứu có thể thử nghiệm, kiểm chứng các giao thức mạng mới trong môi trường thực tế với quy mô lưu lượng thật, giúp cho việc học tập, nghiên cứu được dễ dàng và có thể kiểm nghiệm được mà không cần các thiết bị thật phức tạp. OpenFlow là giao thức giúp bộ điều khiển có thể giao tiếp, cấu hình, điều khiển các bộ chuyển mạch ở phía dưới, cung cấp một giao diện đồng nhất cho các thiết bị của nhiều hãng khác nhau có thể hoạt động được trên cùng một bộ điều khiển.

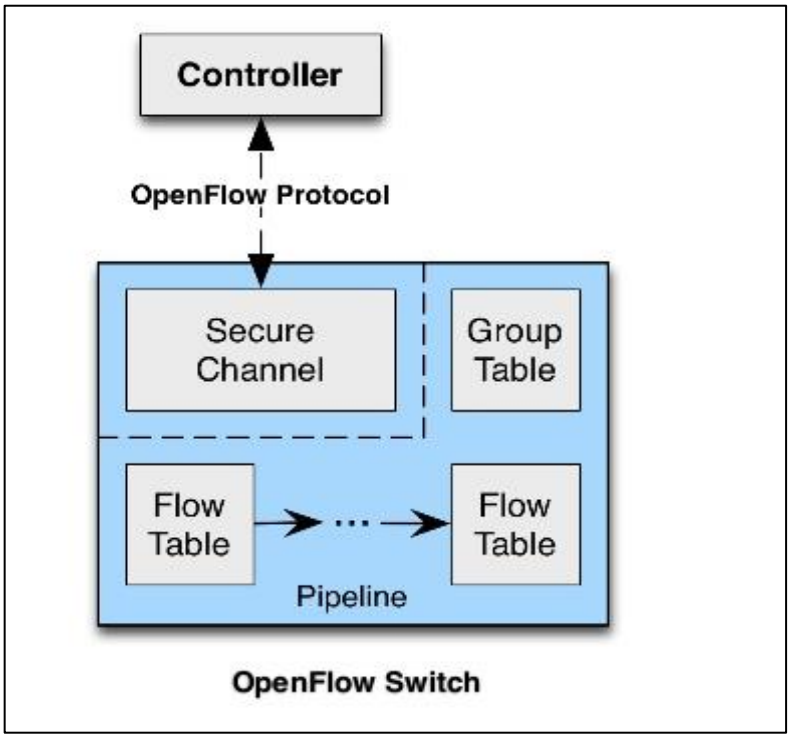
1.2.2. Kiến trúc của OpenFlow Switch

Một thiết bị chuyển mạch OpenFlow bao gồm ít nhất 3 thành phần: Bảng luồng (*Flow table*), Kênh an toàn (*Secure Chanel*) và Giao thức Openflow (*OpenFlow Protocol*).

Flow Table: một liên kết hành động với mỗi luồng, giúp thiết bị xử lý các luồng. Nó có trách nhiệm "nói chuyện" với switch để chỉ ra rằng phải xử lý flow ra sao, mỗi hành động tương ứng với 1 flow-entry.

Secure Channel: kênh kết nối thiết bị tới bộ điều khiển (controller), cho phép các lệnh và các gói tin được gửi giữa controller và thiết bị. Nó kết nối switch với controller sử dụng giao thức OpenFlow chạy qua Secure Sockets Layer (SSL), để gửi các commands và các packets.

OpenFlow Protocol: giao thức cung cấp phương thức tiêu chuẩn mở cho một controller truyền thông với thiết bị.



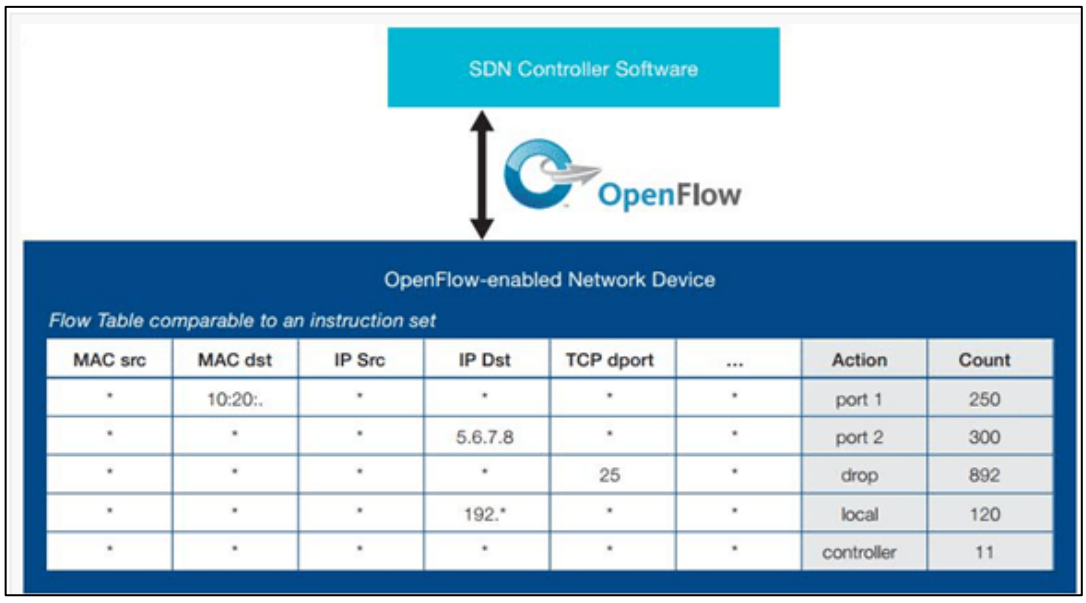
Hình 1.4. Kiến trúc OpenFlow Switch

Có ba loại tập hợp các flow tables:

- Một flow table sẽ ghép các gói tin tới với một flow nhất định và chỉ định các chức năng được thực hiện trên các gói tin đó. Có thể có nhiều flow tables vận hành trong một pipeline.
- Một flow table có để chuyển một luồng vào một Group Table, tại đó có thể kích hoạt cùng một lúc nhiều hành động ảnh hưởng tới một hoặc nhiều flow.
- Một Meter Table có thể kích hoạt nhiều hành động liên quan tới hiệu năng trên một flow.
- Mỗi flow-entry trong flow table có một hành động tương ứng với nó và gồm 3 trường:
    - Packet header định nghĩa nên flow
    - Hành động (Action) định nghĩa cách mà gói tin sẽ được xử lý
    - Thống kê (Statistics) giữ thông tin theo dõi về số lượng gói tin và kích thước theo bytes của mỗi flow, thời gian kể từ lúc gói tin cuối đưa vào flow (nhằm mục đích loại bỏ các flow đã ngừng hoạt động).

Mỗi flow-entry có một hành động tương ứng với nó, và có ba loại hành động cơ bản:

- Chuyển các gói tin của một flow tới port (hoặc các port) đã chỉ định. Điều này cho phép các gói tin được định tuyến qua mạng.
- Đóng gói và chuyển tiếp các gói tin của flow tới controller. Gói tin sẽ được đưa tới Secure Channel, tại đó nó được đóng gói và gửi tới controller. Điển hình như gói tin đầu tiên của mỗi flow mới sẽ được gửi tới controller để được quyết định xem liệu flow có được đưa vào trong flow table hay không.
- Hủy các gói tin của flow. Hành động này được sử dụng nhằm mục đích bảo mật, như tấn công từ chối dịch vụ (DoS).



Hình 1.5. Ví dụ về Flow table trong OpenFlow Switch

Các switch hỗ trợ OpenFlow có 2 loại: **OpenFLow-only** và **OpenFlow-hybrid.** Switch OpenFlow-only chỉ hoạt động theo OpenFlow, ở những switch loại này, tất cả gói tin được xử lý bởi OpenFlow pipeline, và không thể xử lý theo các khác.

OpenFlow-hybrid hỗ trợ các hoạt động theo OpenFLow và chuyển mạch Ethernet thông thường, chuyển mạch lớp 2 truyền thống, cô lập VLAN, định tuyến Layer 3. ACL và xử lý QoS. Những switch này cung cấp một cơ chế phân loại bên

ngoài OpenFlow mà định tuyến lưu lượng hoặc tới OpenFlow pipeline hoặc chuyển mạch thông thường. Ví dụ, một switch có thể sử dụng VLAN tag hoặc cổng ra của gói tin để quyết định cách thức xử lý sử dụng pipeline hoặc cách khác, hoặc nó có thể chuyển tiếp tất cả gói tin tới OpenFlow pipeline. Một switch OpenFlow-hybrid có thể cho phép gói tin đi từ OpenFlow pipeline tới chuyển mạch thông thường thông qua các cổng reserved port như NORMAL hoặc FLOOD.

1.2.3. Hoạt động của OpenFlow Switch

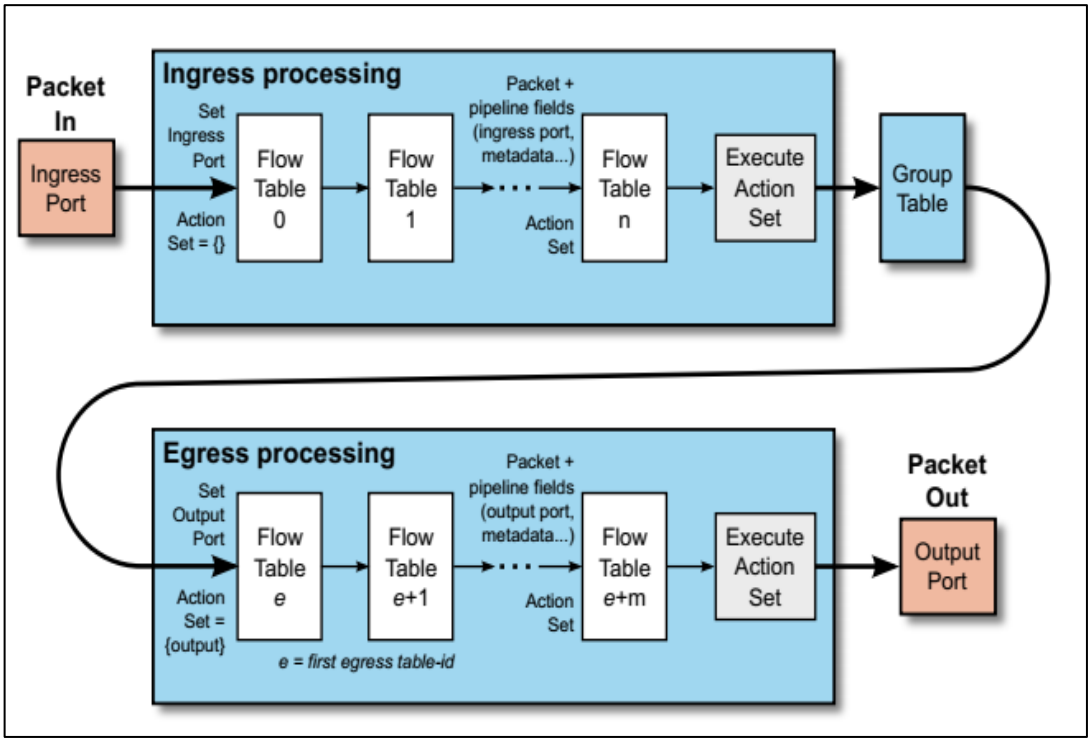
Hình 1.6. Ví dụ về hoạt động của OpenFlow Switch

Giao thức OpenFlow mô tả bản tin trao đổi giữa OpenFlow Controller và một OpenFlow switch. Giao thức này được triển khai trên Secure Socket Layer (SSL) hoặc Transport Layer Security (TLS), cung cấp kênh OpenFlow bảo mật. Giao thức OpenFlow cho phép controller thực hiện các thao tác bổ sung, cập nhật và xóa các hành động vào các flow entry trong các flow tables.

Các bản tin trao đổi trong mạng SDN

- Quá trình xử lý pipeline trong Flow Table

Quá trình xử lý pipeline trong switch được hỗ trợ từ phiên bản openflow 1.1 [3] trở lên với đa bảng trong switch. Quá trình pipeline của mỗi Openflow switch bao gồm nhiều flow tables, mỗi flow table bao gồm nhiều flow entries.



Hình 1.7. Quá trình xử lý Pipeline trong Flow Table

Quá trình xử lý trong pipeline luôn bắt đầu với xử lý ingress ở bảng flow table đầu tiên, gói tin đầu tiên phải match với entry flow trong bảng flow table. Các ingress flow table khác có thể được sử dụng phụ thuộc vào đầu ra khi match của bảng đầu tiên. Nếu đầu ra của quá trình xử lý ingress là chuyển tiếp tới output port, switch OpenFlow có thể thực hiện tiến trình xử lý egress theo output port đó. Xử lý egress là tùy chọn, switch có thể không hỗ trợ bất kì bảng egress nào hoặc có thể không cấu hình sử dụng chúng. Nếu không có egress table nào, gói tin phải được xử lý bởi output port, và hầu hết là chuyển tiếp gói tin ra bên ngoài switch. Nếu có một egress table được cấu hình là bảng egress table đầu tiên, thì gói tin phải match với các flow entry trong bảng đó, và các bảng egress còn lại có thể được sử dụng phụ thuộc vào đầu ra ở flow table.

Khi được xử lý bởi flow table, gói tin được match với các flow entry để chọn ra flow entry phù hợp. Nếu một flow entry được tìm thấy, instruction thiết lập trong flow entry đó được xử lý. Những instruction này có thể chuyển hướng gói tin tới flow table khác, nơi mà quá trình xử lý lại được lặp lại. Một flow entry có thể chỉ chuyển hướng một gói tin tới một bảng flow table có số lớn hơn table của chính nó, hay nói cách khác là xử lý trong pipeline chỉ có chuyển tiếp chứ không có quay lại. Dĩ nhiên, các flow entry của bảng cuối cùng của pipeline có thể không có instruction Go-Table (tới bảng khác). Nếu flow entry match mà không chuyển hướng gói tin tới bảng khác, trạng thái hiện tại của pipeline sẽ dừng lại ở bảng này, gói tin được xử lý với action liên kết trong với nó và thường được chuyển tiếp đi.

Nếu một gói tin không match với flow entry trong bảng, bảng sẽ bị bỏ qua. Hành động trên bảng này phụ thuộc vào cấu hình của bảng. Các instruction trong flow entry của bảng này có thể linh hoạt xác định cách xử lý các gói tin không match, tùy chọn hữu ích là drop gói tin, đưa tới bảng khác hoặc gửi tới controller thông qua kênh control channel thông qua gói tin packet-in.

Trong các trường hợp gói tin không được xử lý đầy đủ bởi flow entry và xử lý pipeline ngừng lại mà không xử lý action cho gói tin hoặc gửi tới một bảng khác. Hoặc nếu không có action với gói tin không match với flow entry thì gói tin sẽ bị drop. Nếu TTL vẫn còn hiệu lực, gói tin có thể được gửi tới controller.

OpenFLow pipeline trên mỗi switch OpenFlow chứa một hoặc nhiều bảng flow tables, mỗi flowtable chứa nhiều flow entry. Quá trình xử lý trong pipeline định nghĩa cách mà các gói tin tương tác với các flow table này. Một OpenFlow switch yêu cầu phải có ít nhất một ingress flow table, và có thể có thêm nhiều flow table tùy ý. OpenFlow switch với chỉ một flow table là có thể, trong trường hợp này xử lý trong pipeline là đơn giản nhất.

Các flow table được đánh số để các gói tin đi qua, bắt đầu từ 0. Quá trình xử lý trong pipeline xảy ra 2 trạng thái: ingress processing và egress processing. Tất cả các table với số nhỏ hơn bảng flow đầu tiên ở egress phải được sử dụng trong các

bagnr ingress, và không bảng nào với số cao hơn hoặc bằng với egress table đầu tiên có thể được sử dụng trong bảng ingress table.

Mỗi flow table gồm các cột sau:

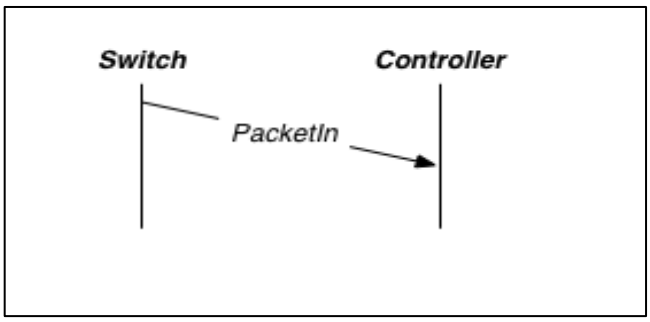
Match field: để so sánh với các gói tin. Trường này bao gồm thông tin về ingress port và các packet header, và có thể các trường pipeline field thêm vào như metadata chỉ định trong các bảng trước đó.

- Priority: Mức độ ưu tiên của flow entry.
- Counters: được cập nhật khi các gói tin được match.
- Instructions: điều chỉnh thiết lập action hoặc xử lý pipeline.
- Timeouts: số lượng thời gian giới hạn hoặc thời gian chờ trước khi flow bị vô hiệu bởi switch.
- Cookie: Dữ liệu không rõ ràng được chọn bởi switch. Có thể được sử dụng bởi controoler để lọc các flow entry ảnh hưởng bởi các flow tĩnh, yêu cầu chỉnh sửa hoặc xóa flow. Không sử dụng khi xử lý các gói tin.
- Flags: Các flag thay đổi cách mà các flow entry được quản lý, ví dụ: cờ OFPFF_SEND_FLOW_REM tự động loại bỏ các bản tin bị xóa cho flow entry đó.

Một entry flow trong bảng được xác định bởi trường match fields và priority của nó: 2 trường này kết hợp tạo nên sự duy nhất cho flow entry trong một bảng. Flow không khớp (tất cả các trường đều không khớp) và có độ ưu tiên bằng 0 được gọi là flow entry table-miss. Instruction có thể chứa các action được thực hiện trên gói tin ở một vài điểm của của pipeline. Action set-field có thể chỉ định viết một số trường header.

1.3. Các bản tin trao đổi OpenFlow [6]

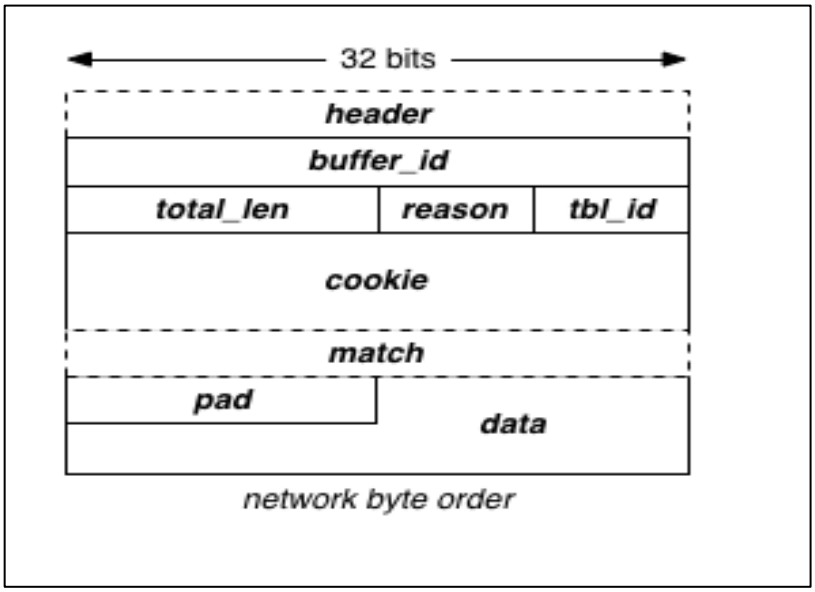
1.3.1. Bản tin PacketIn



Hình 1.8. Bản tin PacketIn

PacketIn là bản tin được gửi từ switch lên Controller. Có hai lý do để thực hiện điều này: Một gói tin match với một flow entry và flow entry đó có action gửi lên Controller hoặc một gói tin match với table_miss và trong table_miss có action là gửi lên Controller.

Cấu trúc bản tin bao gồm các trường được mô tả như hình bên dưới.



Hình 1.9. Cấu trúc bản tin PacketIn

Trong đó:

Header: Là header tiêu chuẩn của giao thức openflow

Buffer_id: Là giá trị bộ đệm –lưu trữ gói tin được match mà được sử dụng bởi các datapath để các định vị trí đệm của gói tin. Một gói tin được match có thể được đệm vào bộ đệm hoặc không. Khi một gói tin được đệm vào bộ đệm thì một phần dữ liệu của gói tin đó sẽ được nằm trong phần data của bản tin PacketIn. Nếu gói tin không được đệm vào bộ đệm do bộ đệm không có sẵn hoặc do yêu cầu thì toàn bộ gói tin sẽ nằm trong phần data của gói tin PacketIn và khi đó trường buffer_id sẽ có giá trị là OFP_NO_BUFFER.

Total_len: Tổng kích thước gói tin PacketIn

Reason: Có thể là một trong các giá trị sau:

- OFPR_NO_MATCH=0: Không match bất cứ flow entry nào
- OFPR_ACTION=1: Action gửi lên Controller
- OFPR_INVALID_TTL = 2: Gói tin có TTL không hợp lệ

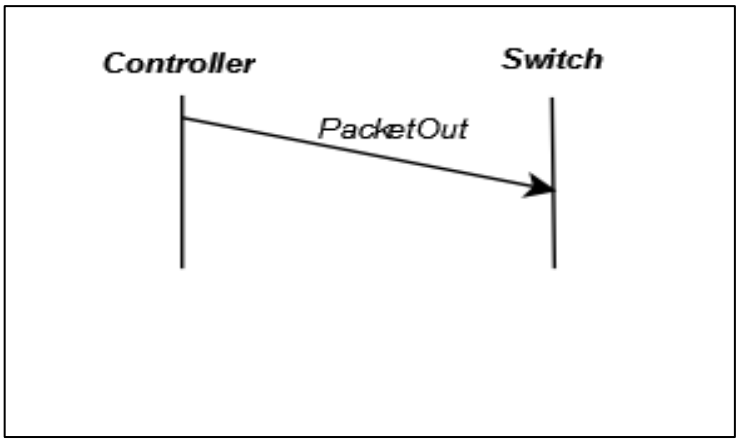
Table_id: Table gửi gói tin này

Match: Phản ánh tiêu đề của gói tin

Data: Nội dung của gói tin

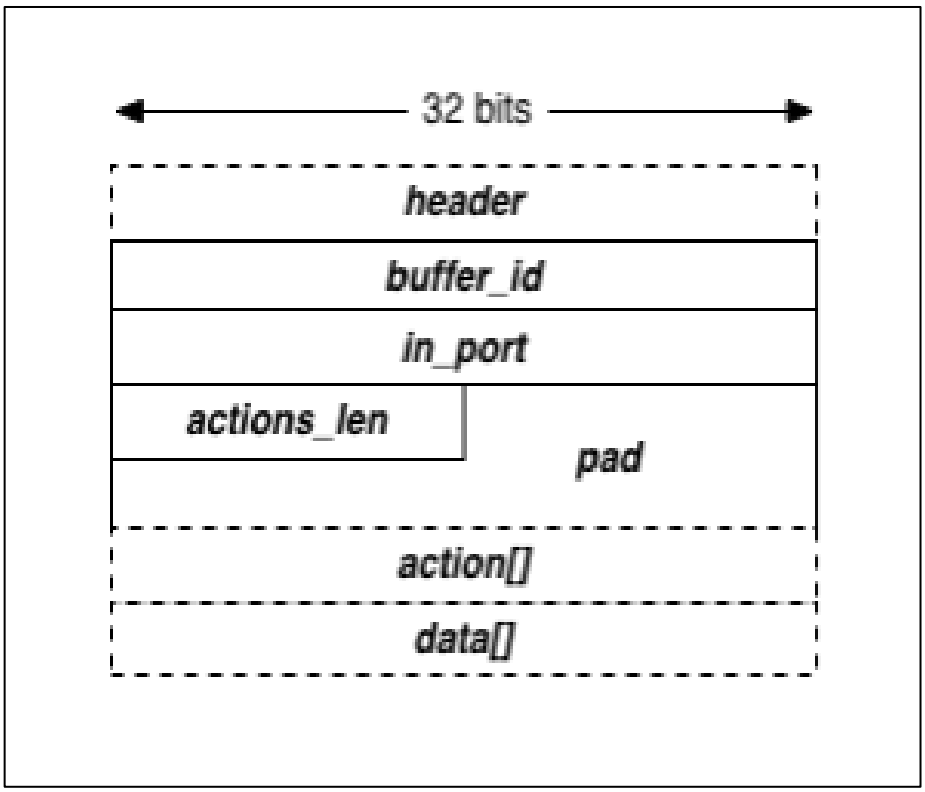
Trong đề tài này tác giả sử dụng bản tin PacketIn để gửi toàn bộ gói tin SYN và gói tin SYN_ACK lên Controller. Vì vậy các flowentry trên Switch có trường actions = CONTROLLER: 65535.

1.3.2. Bản tin PacketOut



Hình 1.10. Bản tin PacketOut

Bản tin PacketOut được gửi từ Controller tới Switch dùng để hướng dẫn gói tin đi như thế nào trong mạng. Bản tin này không dùng để thiết lập entry trên Switch mà chỉ gửi gói tin ra ngoài Switch. Cấu trúc trường của bản tin được thể hiện như hình bên dưới:



Hình 1.11. Cấu trúc bản tin PacketOut

Trong đó:

Header: Là header tiêu chuẩn của giao thức openflow.

Buffer_id: Là giá trị bộ đệm –lưu trữ gói tin được match mà được sử dụng bởi các datapath để các định vị trí đệm của gói tin. Một gói tin được match có thể được đệm vào bộ đệm hoặc không. Khi một gói tin được đệm vào bộ đệm thì một phần dữ liệu của gói tin đó sẽ được nằm trong phần data của bản tin PacketIn. Nếu gói tin không được đệm vào bộ đệm do bộ đệm không có sẵn hoặc do yêu cầu thì toàn bộ gói tin sẽ nằm trong phần data của gói tin PacketIn và khi đó trường buffer_id sẽ có giá trị là OFP_NO_BUFFER.

In_port: Là port đầu vào mà được liên kết với gói tin. Nó có thể là port tiêu chuẩn hoặc OFPP_CONTROLLER

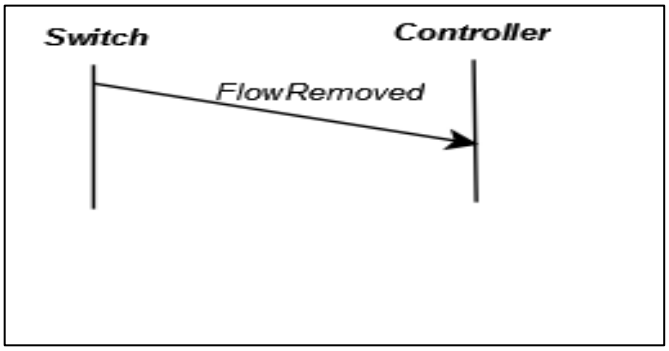
Action []: Là một danh sách các Action định nghĩa các gói tin nên được xử lý như thế nào bởi Switch. Nó có thể là sửa đổi gói tin, port đầu ra.

Data []: Trường dữ liệu của gói tin

Trong đề tài này, tác giả sử dụng bản tin này để gửi các gói tin TCP có cờ SYN,cờ SYN_ACK, cờ ACK, cờ RST ra ngoài cổng Switch mà không thiết lập entry trên Switch.

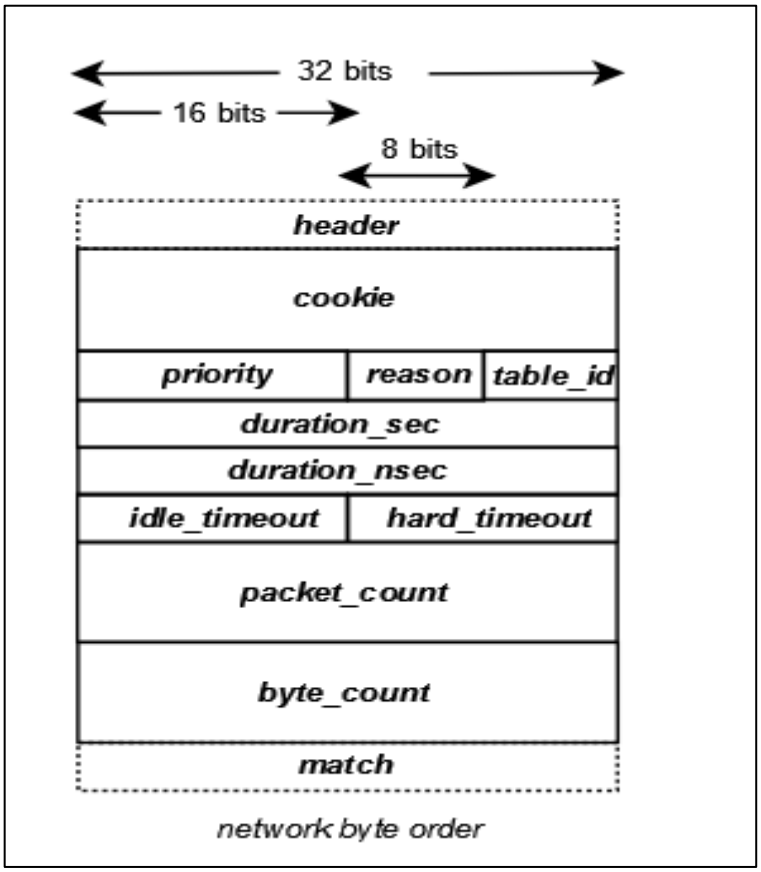
1.3.3. Bản tin FlowRemoved

FlowRemoved là bản tin được gửi tới Controller từ Switch khi có flow entry trong flow table bị xóa.



Hình 1.12. Hoạt động của bản tin FlowRemoved

Cấu trúc bản tin FlowRemoved được mô tả như bên dưới:



Hình 1.13. Cấu trúc bản tin FlowRemoved

Trong đó:

Header: Là header tiêu chuẩn của giao thức openflow

Priority: Mức ưu tiên của flow entry bị xóa

Reason: Trường reason sẽ có một trong các giá trị sau

- OFPRR_IDLE_TIMEOUT = 0:flow có thời gian vượt quá idle_timeout
- OFPRR_HARD_TIMEOUT = 1 : flow có thời gian vượt quá hard_timeout
- OFPRR_DELETE = 2: bị xóa bởi flow mod
- OFPRR_GROUP_DELETE = 3

Duration_sec và **duration_nsec** là thời gian sống của flow entry đó trước khi bị xóa khỏi table tính theo giây và nano giây

Idle_timeout: thời gian timeout của flow entry đó. Idle_timeout được định nghĩa là khoảng thời gian lớn nhất giữa hai gói tin liên tiếp của một flow

Hard_timeout: thời gian tồn tại của flow đó. Hard_timeout được định nghĩa là khoảng thời gian tồn tại của flow trong flow table

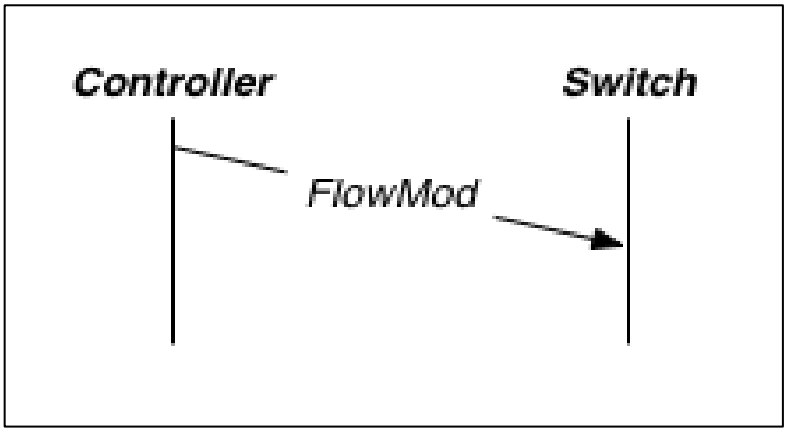
Packet_count: Số lượng gói tin mà match flow entry đó trước khi bị xóa khỏi table

Byte_count: Số lượng byte dữ liệu mà match flow entry đó trước khi bị xóa khỏi table

Trong luận văn này, tác giả sử dụng bản tin FlowRemoved để báo cho Controller biết các flow entry nào trong table 2 bị xóa và từ đó có các hành động xử lý tiếp theo

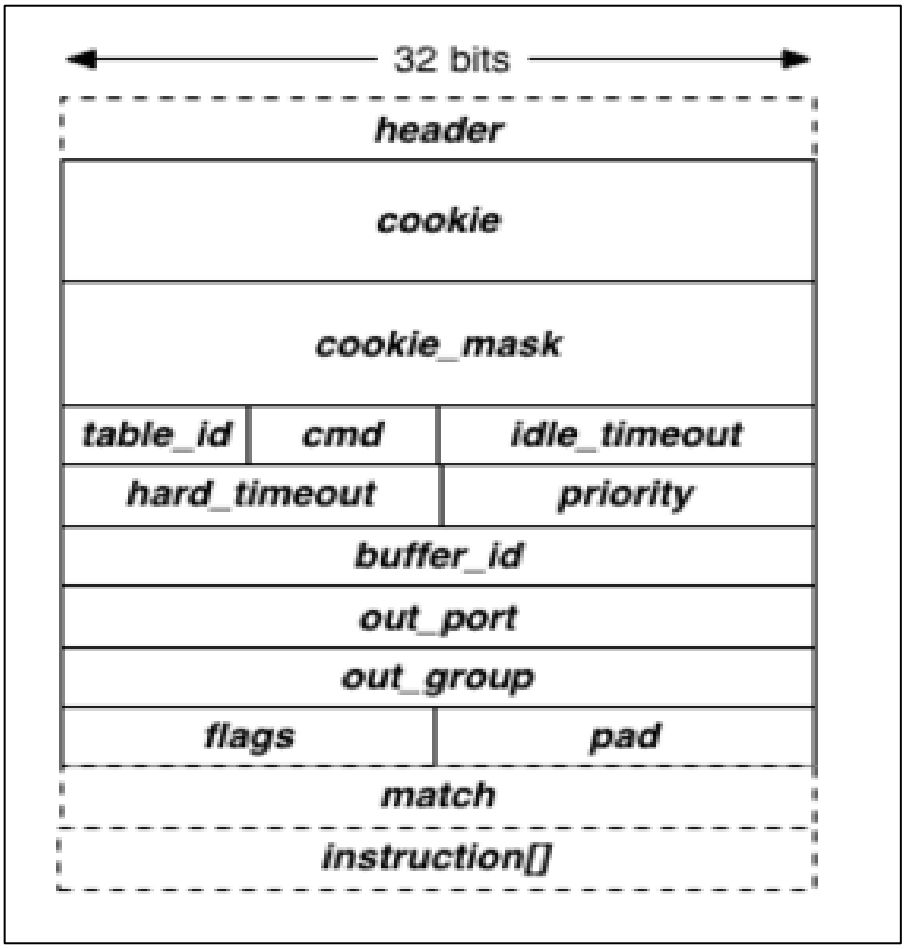
1.3.4. Bản tin FlowMod

Đây là một trong những bản tin chính hay được sử dụng giữa Controller và Switch. Nó được gửi từ Controller xuống Switch cho phép Controller sửa đổi trạng thái của Openflow Switch.



Hình 1.14. Hoạt động của FlowMod

Cấu trúc bản tin FlowMod được mô tả như bên dưới:



Hình 1.15. Cấu trúc bản tin FlowMod

Table_id: Chỉ ra table mà flow entry được chèn vào, bị sửa đổi hoặc bị xóa.

Cmd: Là một trong các giá trị sau:

- OFPFC_ADD = 0 : Tạo mới một flow
- OFPFC_MODIFY = 1: Sửa đổi tất cả các flow được match
- OFPFC_MODIFY_STRICT = 2
- OFPFC_DELETE = 3: Xóa tất cả các flow được match
- OFPFC_DELETE_STRICT = 4

Idle_timeout: thời gian timeout của flow entry đó. Idle_timeout được định nghĩa là khoảng thời gian lớn nhất giữa hai gói tin liên tiếp của một flow

Hard_timeout: thời gian tồn tại của flow đó. Hard_timeout được định nghĩa là khoảng thời gian tồn tại của flow trong flow table

Priority: Mức ưu tiên của flow trong flow table, số càng cao thì mức ưu tiên càng cao. Trường này chỉ sử dụng trong bản tin OFPFC_ADD khi match và khi thêm flow entry và cho bản tin OFPFC_MODIFY_STRICT hoặc OFPFC_DELETE_STRICT khi match flow entries.

Buffer_id: Chỉ ra gói tin được đệm ở Switch và gửi lên Controller bằng bản tin Packet_in. Một bản tin mà có buffer_id là tương đương với việc gửi hai bản tin tuần tự là bản tin flow mod và bản tin PacketOut với yêu cầu là Flow mod được xử lý trước.

Out_port và **Out_group**: Lọc ra phạm vi của OFPFC_DELETE và OFPFC_DELETE_STRICT

Flags: Là một trong các giá trị sau:

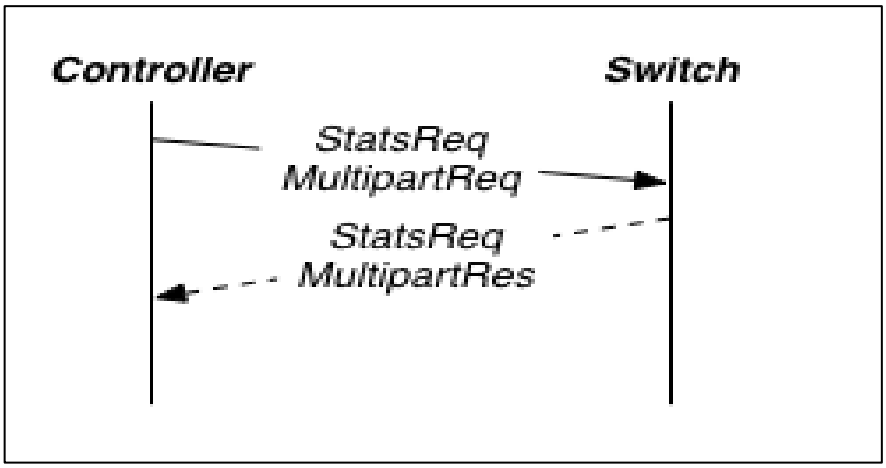
- OFPFF_SEND_FLOW_REM = 1 << 0: Gửi flow removed khi flow hết hạn hoặc flow bị xóa.
- OFPFF_CHECK_OVERLAP = 1 << 1
- OFPFF_RESET_COUNTS = 1 << 2
- OFPFF_NO_PKT_COUNTS = 1 << 3
- OFPFF_NO_BYT_COUNTS = 1 << 4

Instruction: Chứa các hướng dẫn cho các flow entry khi được thêm hoặc sửa đổi các entries.

Trong đề tài này tác giả đã sử dụng thiết lập cờ OFPFF_SEND_FLOW_REM đối với các flow ở table 2. Vì vậy khi flow entry bị timeout ở table này thì nó phải đóng gói bản tin FlowRemoved để gửi lên Controller. Sử dụng bản tin FlowMod với trường Cmd = OFPFC_ADD để add một flow entry xuống table 2 (dùng để bẫy cờ ACK) và để add hai flowentry xuống table 3 hướng dẫn cho các gói tin thuộc luồng đó làm thế nào để đi trong mạng. Trường Cmd = OFPFC_MODIFY dùng để sửa đổi flowentry trên table 1. Trường Cmd = OFPFC_DELETE dùng để xóa các flow entry trên table 2.

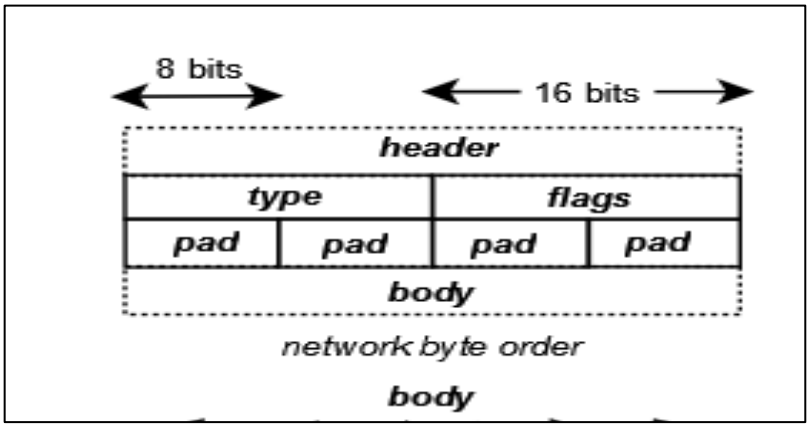
1.3.5. Bản tin StatsRequest

Bản tin StatsRequest được sử dụng để yêu cầu về thông tin của từng flow, từng table trên Switch.



Hình 1.16. Hoạt động của bản tin StatsRequest

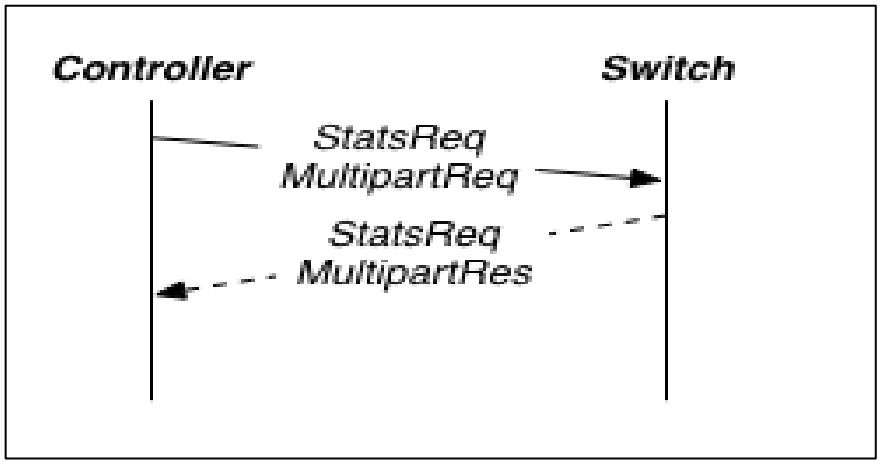
Cấu trúc bản tin StatsRequest được mô tả như hình bên dưới:



Hình 1.17. Cấu trúc bản tin StatsRequest

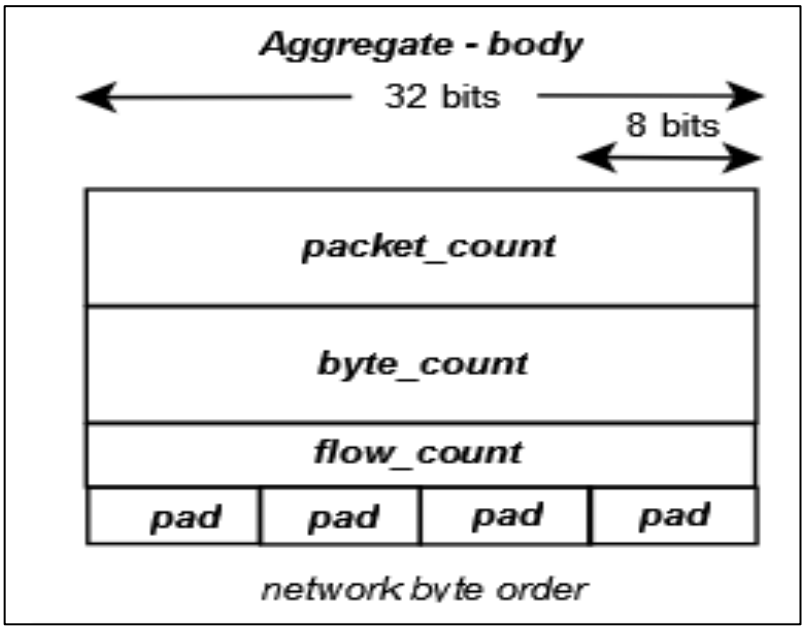
1.3.6. Bản tin StatsResponse

StatsResponse dùng để trả lời bản tin StatsRequest.



Hình 1.18. Hoạt động của bản tin StatsResponse

Trong luận văn này tác giả lấy tổng số flowentries trong phần body.



Hình 1.19. Phần body của bản tin StatsResponse

Trong đó **flow_count**: Đếm tổng số flow entries trên Switch.

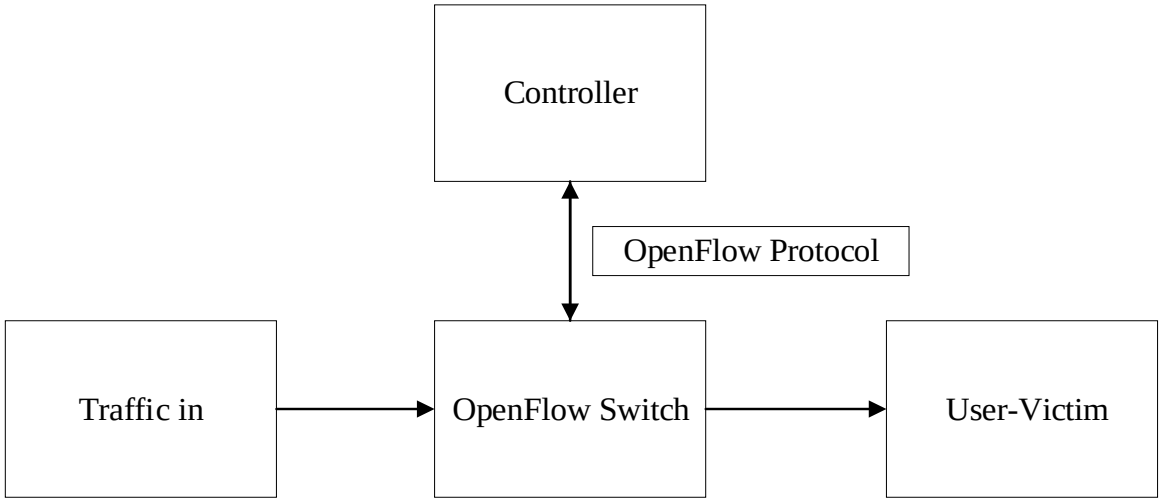
CHƯƠNG 2. XÂY DỰNG KIẾN TRÚC MẠNG SDN/OPENFLOW SỬ DỤNG TRONG PHÒNG CHỐNG TẤN CÔNG

Qua việc tìm hiểu và nghiên cứu lý thuyết cũng như các giải pháp có thể triển khai phù hợp với điệu kiện cơ sở vật chất thực tế. Tác giả đã thực hiện xây dựng một hệ thống kiến trúc mạng SND/OpenFlow, đồng thời giả lập hình thức tấn công và đưa ra giải pháp giảm thiểu tấn công trên hệ thống giả lập này. Ý tưởng của việc giảm thiểu tấn công đó là việc kiểm soát lưu lượng tới,đặt một ngưỡng giới hạn để phát hiện tấn công. Lưu lượng tới sẽ bị hủy bỏ hoàn toàn nếu vượt qua ngưỡng giới hạn này, việc hủy bỏ sẽ được điều khiển bởi SDN Controller.

2.1. Giả lập kiến trúc mạng SDN/OpenFlow

Dựa vào kiến trúc mạng SDN đã nghiên cứu trong những chương trước, tác giả đưa ra giả lập mô hình mạng của một trung tâm cung cấp dịch vụ có sự tiếp nhận và sử dụng lưu lượng gửi tới các DNS Server thông qua thiết bị chuyển mạch, chịu sự giám sát và điều khiển trực tiếp bởi bộ điều khiển (Controller). SDN Controller như một khối đầu não quan trọng của hệ thống, nó kiểm soát và theo dõi mọi hoạt động của hệ thống tại mọi thời điểm, đồng thời có khả năng ngăn chặn các luồng tấn công được phát hiện trên bộ chuyển mạch OpenFlow, bộ chuyển mạch đóng vai trò như một “bức tường” bảo vệ cho các thiết bị phía trong mô hình, thực hiện các chức năng đảm bảo cho việc cung cấp dịch vụ không bị gián đoạn. Sự linh hoạt của hệ thống đó chính là chức năng lập trình được trên bộ phận điều khiển SDN Controller, người kiểm soát có thể cài đặt các chức năng khác nhau trên bộ điều khiển này, đây chính là giải pháp cho nhiều loại tấn công từ chối dịch vụ khác trong mạng SDN.

Kiến trúc tổng quan của hệ thống được thể hiện như bên dưới (Hình 2.1)



Hình 2.1. Kiến trúc mạng SDN/OpenFlow giả lập

Theo đó lưu lượng đi vào hệ thống sẽ phải đi qua OpenFlow Switch rồi trao đổi với SDN Controller sau đó mới đến các máy User (đóng vai trò là nơi bị tấn công). Tuy nhiên, tồn tại một điểm yếu của hệ thống đó là trong quá trình xảy ra tấn công, lưu lượng lưu thông trao đổi giữa OpenFlow Switch với bộ điều khiển đạt ngưỡng khá cao khiến kênh kết nối này phải chịu tải lớn. Điều này làm cho thời gian đáp ứng của hệ thống là khá lâu, việc kiểm soát và thu thập dữ liệu tốn nhiều thời gian sẽ ảnh hưởng tới tốc độ phản ứng của bộ điều khiển. Để giải quyết vấn đề này, tác giả đã nghiên cứu và tham khảo trên nhiều phương diện từ nhiều nguồn khác nhau và đưa ra giải pháp tích hợp sử dụng NetFPGA làm khối OpenFlow Switch (Hình 2.2) và xây dựng một khối giám sát lưu lượng Network Monitor được tích hợp ngay trên OpenFlow Switch, dựa trên các flow bộ giám sát có thể theo dõi các gói tin đi trong luồng lưu lượng tới bộ chuyển mạch.

Hình 2.2. Board mạch NetFPGA

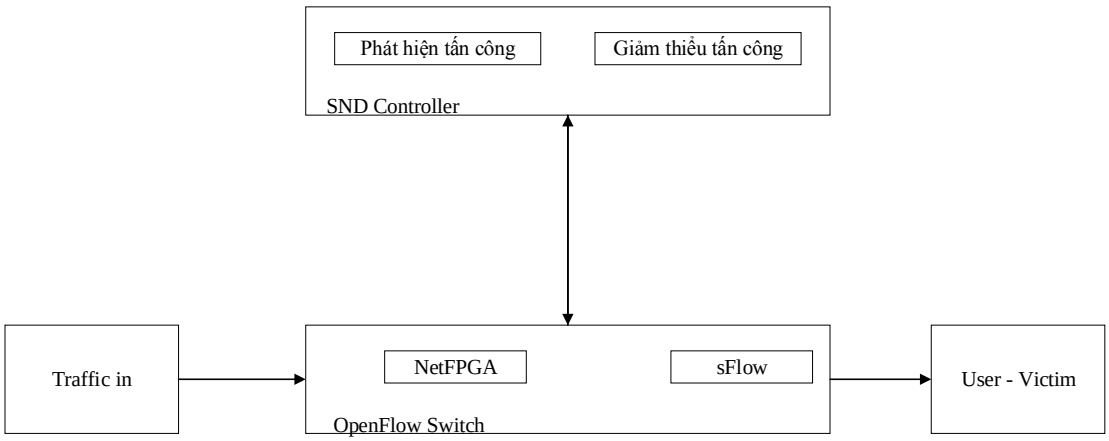
NetFPGA là một nền tảng phần cứng, với mục đích thiết kế nhắm phục vụ cho việc giảng dậy và nghiên cứu chế tạo các khối phần cứng trong mạng như các card mạng, bộ định tuyến, khối chuyển mạch….Lợi ích trong việc sử dụng nền tảng NetFPGA là khả năng cung cấp cho người sử dụng một thiết bị chuyển mạch có hiệu năng lớn, băng thông tối đa trên mỗi cổng có thể lên tới 1Gbps (tùy phiên bản có thể lên tới giá trị 10Gbps). Trong điều kiện bình thường với tần số làm việc là 125Mhz bộ chuyển mạch dựa trên nền tảng NetFPGA được sử dụng trong hệ thống giả lập có thể đáp ứng luồng lưu lượng tối đa là 8Gbps (4 card x 1Gbps x 2 chiều). Bên cạnh đó trong cấu trúc bộ Kit còn có một chip FPGA, người sử dụng có thể tùy ý thay đổi và lập trình để phù hợp với chức năng của OpenFlow Switch.Trên khối NetFPGA này, các gói tin đi vào từ một cổng mạng bất kỳ của nó sẽ được xử lý định tuyến, sau đó được đẩy ra bởi một cổng mạng khác. Quá trình này diễn ra liên tục khi có luồng lưu lượng đến và các gói tin đi vào bộ chuyển mạch. Song song với quá trình tiếp nhận lưu lượng vào thì NetFPGA vẫn theo dõi và nhận các tín hiệu điều khiển được chuyển xuống từ SDN controller. Khi một gói tin đến Kit NetFPGA trước tiên nó sẽ được tra cứu và so sánh với các flow entry nằm trong flow table của bộ chuyển mạch OpenFlow Switch, nếu trùng khớp sẽ được chuyển

tiếp trong phần cứng với một tốc độc cao, còn không trùng khớp thì nó sẽ được đóng gói vào một bản tin và gửi lên phía khối điều khiển là SDN Controller.

Cấu trúc của NetFPGA được sử dụng trong hệ thống giả lập :

- Phần cứng.
- Xilinx Virtex^{TM} II pro 50 FPGA
- 4.5 Mb SRAM, 64 Mb DDR2
- 4 Gbps Ethernet Port
- 64MB DDR2 DRAM
- FPGA Spartan II Chip
- Phần mềm.
- Hệ điều hành Linux
- Drive kit NetFPGA
- Phần mềm giao diện người dùng.

Khối giám sát lưu lương được triển khai và tích hợp ở đây là phần mềm sFlow, một công nghệ cho phép chúng ta có thể kiểm soát lưu lượng với tốc độ cao, khả năng giám sát các liên kết có tốc độ lên tới 10Gbps và hơn thế nữa mà không ảnh hưởng đến hiệu năng của OpenFlow Switch. Có thể triển khai trên nhiều thiết bị mạng. Nó cung cấp một cái nhìn toàn diện về mạng với các chức năng đo lưu lượng, thu thập, lưu trữ, phân tích dữ liệu truy cập…. Là một giải pháp chi phí thấp, đơn giản không yêu cầu thêm các bộ nhớ và CPU.

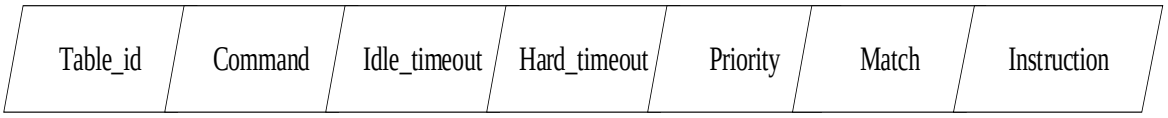


Hình 2.3. Kiến trúc tổng thể hệ thống giả lập

2.2. Nguyên lý hoạt động của hệ thống [5]

2.2.1. Cách thức hoạt động của Controller

Bộ điều khiển SDN Controller có nhiệm vụ ra quyết định cho khối chuyển mạch OpenFlow Switch thực thi thông qua giao thức OpenFlow. Nhận thông báo về những phân tích các flow entry đi vào mạng, khi xảy ra tấn công, SDN Controller sẽ nhận được một thông báo và kèm theo đó là một dãy các địa chỉ IP của nguồn tin truy vấn có tần suất cao tới hệ thống, đa số đều là các nguồn tin đang thực hiện việc tấn công hệ thống, đều cần phải được ngăn chặn. Khi đã nhận được thông báo đang có tấn công cùng với danh sách địa chỉ IP, SDN Controller sẽ gửi một bản tin điều khiển xuống khối chuyển mạch OpenFlow Switch để làm rớt tất cả các gói tin đi từ những địa chỉ IP đang bị nghi ngờ, từ đó lưu lượng đi qua bộ chuyển mạch được khống chế, đảm bảo cho các thiết bị ở phía người dùng không bị ảnh hưởng. SDN Controller sẽ gửi bản tin FlowMod xuống cho bộ chuyển mạch để thực hiện việc hủy bỏ các gói tin. Cấu trúc các khối của một bản tin FlowMod được mô tả như Hình 2.4



Hình 2.4. Cấu trúc bản tin FlowMod

Thông thường một FlowMod gửi xuống để thêm một Flow-entry và Flow table tương ứng với trường Table_id, thời gian hard_timeout không giới hạn, đặt một giá trị giới hạn cho idle_timeout, trường priority là 65535 để có mức độ ưu tiên cao trong Flow table, trường Match để lọc ra các gói tin tấn công ứng với địa dịa IP trong danh sách gửi lên Controller, trường instruction để chứa hành động là Drop gói tin.

2.2.2. Cách hoạt động của chuyển mạch OpenFlow Switch

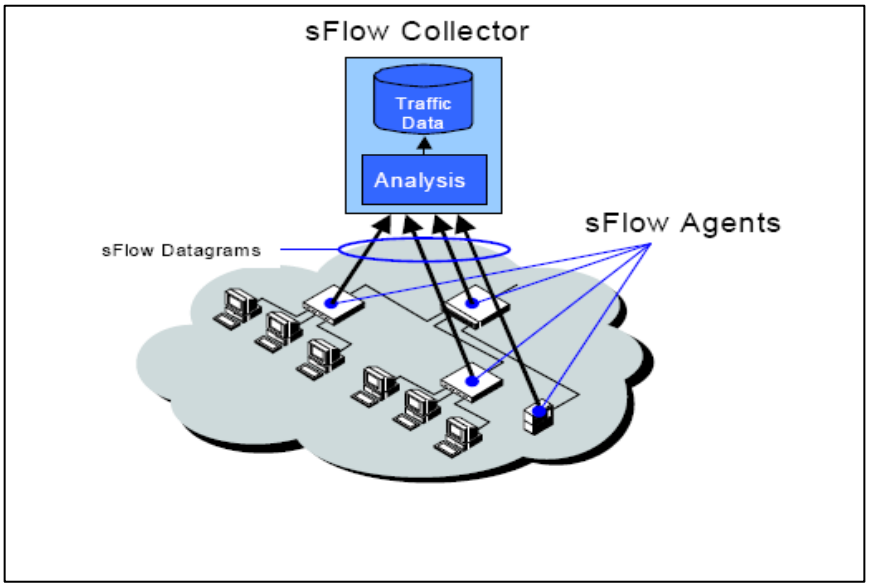
Bộ chuyển mạch sẽ thực hiện việc chuyển tiếp lưu lượng lưu thông trong mạng, lưu giữ các bảng trạng thái. Khi các Flow entry đi đến bộ chuyển mạch

OpenFlow Switch sẽ được đối chiếu với bảng trạng thái Flow table để thực hiện việc định tuyến và chuyển tiếp các gói tin, tất cả mọi hoạt động đều chịu sự giám sát của SDN Controller, các giao tiếp giữa Controller với chuyển mạch OpenFlow Switch đều được thực hiện thông qua giao thức OpenFlow. Bên cạnh đó, trên bộ chuyển mạch còn tích hợp thêm khối giám sát lưu lượng sFlow, các gói tin đến sẽ được lấy mẫu để theo dõi, kiểm soát và xử lý, phân tích dữ liệu và gửi lên controller theo định kỳ đế đảm bảo kịp thời phát hiện khi có tấn công xảy ra.

2.2.3. Cách thức hoạt động của bộ kiểm soát lưu lượng sFlow – Network Monitoring

Bộ kiểm soát lưu lượng có chức năng thực hiện giám sát và kiểm soát và phân tích lưu lượng đi vào hệ thống mạng, và chuyển tiếp toàn bộ thông tin phân tích được lên bộ điều khiển. Là phần mềm sẽ được nhúng trong khối OpenFlow Switch trên nền tảng NetFPGA. Hoạt động dựa trên công nghệ sFlow với cấu trúc Agent – Collector. Các sFlow Agent sẽ thực hiện nhiệm vụ thu thập lưu lượng theo một chu kỳ thời gian lấy mẫu, rồi đưa lên khối sFlow Collector, giúp Collector có thể bao quát toàn bộ lưu lượng trong mạng theo thời gian thực.

sFlow Collector sẽ liên tục nhận được các gói tin được gửi lên từ sFlow Agent, nó sẽ theo dõi số lượng gói tin đến, các bản tin Response cũng như là tính toán kích thước trung bình của gói tin với mỗi cặp địa chỉ IP nguồn và IP đích. Nếu những thông số trên vượt ngưỡng đã đặt ra thì tất cả những thông tin phân tích sẽ được thông báo tới SDN Controller, từ đó SDN Controller có được thông tin về nguồn IP đang tấn công và IP đang bị tấn công, để thực hiện các chính sách giảm thiểu cần thiết.



Hình 2.5. Cấu trúc Agent- Collector của sFlow

Việc tích hợp khối sFlow ngay trên khối chuyển mạch OpenFlow Switch giúp cho hệ thống tinh gọn, giảm lưu lượng tải cho đường truyền giữa bộ chuyển mạch và SDN controller mà không ảnh hưởng tới hiệu năng của việc chuyển mạch lưu lượng. Bên cạnh đó kết hợp giữa sFlow với OpenFlow trong kiến trúc mạng SDN mang lại một sự linh hoạt trong việc triển khai các giải pháp chống tấn công DDoS.

2.3. Kịch bản tấn công và giải pháp giảm thiểu tấn công khuyếch đại DNS

2.3.1. Xây dựng hệ thống

Hệ thống bao gồm các thiết bị : 1 SDN Controller, 1 OpenFlow Switch trên nền tảng NetFPGA(tích hợp khối Network Monitoring), 1 máy phát lưu lượng giả lập như lưu lượng thời gian thực (bao gồm cả lưu lượng tấn công và lưu lượng bình thường), 1 Server đóng vai trò là nạn nhân của cuộc tấn công (Victim).

a) Khối SDN Controller – Floodlight Controller.

Foodlight controller là một bộ điều khiển mở được lập trình và phát triển bởi ngôn ngữ Java, cấu trúc tinh giản dễ dàng tạo và thêm mới các module. Được hỗ trợ các platform cho phép người lập trình triển khai các ý tưởng mới trong lĩnh vực mạng, sử dụng các thiết bị phần cứng thật, các phiên bản giao thức OpenFlow từ 1.0

tới 1.4. Trong hệ thống giả lập này Floodlight Controller được cài đặt trên hệ điều hành Ubuntu 16.04 LTS trên máy tính cấu hình:

- Vi xử lý: Intel® Core TM i7-6700M CPU @ 3.4GHz
- HDD: 1TB
- Ram: 16GB

Trình ứng dụng chạy trên Floodlight Controller bao gồm các khối chức năng nhỏ đó là :

- Khối phân tích lưu lượng : nhận thông tin đầu vào là các bản tin StatisticResponse từ OpenFlow Switch gửi lên phía Floodlight Controller.
- Khối thực thi hành động : thực hiện chức năng đóng gói và gửi bản tin FlowMod xuống bộ chuyển mạch để thực thi chính sách giảm thiểu tấn công.

Trong mô hình mà tác giả xây dựng, Floodlight Controller được chứa trong Server có địa chỉ IP: 192.168.101.243. Đây là địa chỉ IP tĩnh, sẽ không thay đổi theo thời gian nên rất thuật tiện cho những lần truy cập khác.

b) Khối OpenFlow Switch

Được cấu hình với các card NetFPGA để tạo ra nhiều cổng Ethernet đem lại những tối ưu về hiệu suất sử dụng của hệ thống như đã nói ở các phần trước. Các card của NetFPGA sẽ được cấu hình như những cổng của OpenFlow Switch đảm bảo thực hiện chức năng như là một bộ chuyển mạch OpenFlow Switch. Khối OpenFlow Switch sử dụng phiên bản 2.3.0 được cài đặt hệ điều hành Ubuntu 16.04 trên máy tính với cấu hình:

- Vi xử lý: Intel® Core TM i7-6700M CPU @ 3.4GHz
- HDD: 1TB
- Ram: 8GB

Trình ứng dụng được cài trên OpenFlow Switch là khối sFlow-rt : thực hiện việc lấy mẫu và kiểm soát lưu lượng lưu thông tại cả hai phía vào và ra của Controller.

c) Khối Victim – FPT Server

Đóng vai trò là nơi tiếp nhận lưu lượng của cuộc tấn công. Server được cài đặt hệ điều hành Ubuntu 14.04 LTS trên máy tính với cấu hình:

- Vi xử lý: Intel® Core TM i7-6700M CPU @ 3.4GHz
- HDD: 1TB
- Ram: 64GB

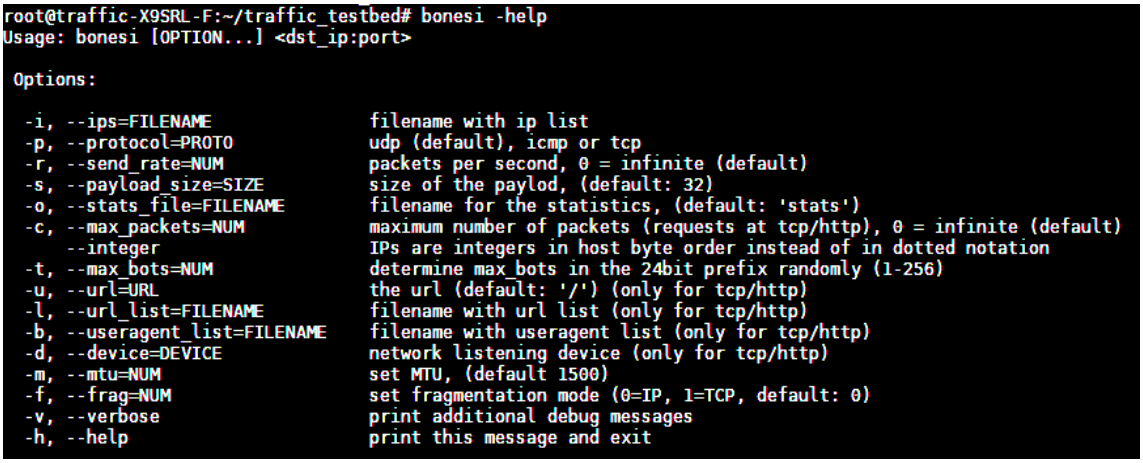
2.3.2. Công cụ hỗ trợ

a) Bonesi

Là một công cụ giả lập tấn công phát triển trên hệ điều hành mã nguồn mở Ubuntu, tạo ra được môi trường Botnet giả lập chứ nhiều địa chỉ IP được coi là nguồn tấn công. Ngoài ra còn hỗ trợ nhiều kiểu tấn công DDoS với các loại giao thức sẵn có như UDP, TCP,…Tất cả các thông số như tốc độ phát, kích thước gói, số lượng gói phát trong một đơn vị thời gian đều có thể linh hoạt thay đổi được để tạo ra nhiều kịch bản tấn công khác nhau. Đây là một phần mềm miễn phí được cấp phép theo giấy phép Apache, Phiên bản 2.0. Khi người dùng sử dụng Bonesi sẽ được cung cấp sẵn một tệp chứa sẵn 50000 botnet (50k-bots.txt). Đây chính là những botnet mà công cụ này giả lập trong quá trình mô phỏng tấn công. Thông thường để sử dụng trong khi kiểm thử hay mô phỏng, người dùng thường tách tệp này tạo ra những tệp chứa số botnet nhỏ hơn hoặc lớn hơn (sử dụng python để sinh ra nhiều botnet hơn) để phục vụ những nhu cầu khác nhau. Để tạo ra những tệp nhỏ hơn, chúng ta chỉ đơn giản là sao chép số dòng chúng ta muốn sang một tệp mới. Ví dụ, chúng ta sao chép 3000 botnet từ 50000 botnet trên tạo ra một tệp mới có tên là 3k-bots.txt. Cũng giống như các công cụ nguồn mở khác, Bonesi cho phép người dùng tra cứu trợ giúp thông qua câu lệnh truy vấn sự trợ giúp như sau:

Sudo bonesi -help

Kết quả của lệnh trên được thể hiện trong Hình 2.6 sau:



Hình 2.6. Các tùy chọn sử dụng để phát tấn công

Trong đó :

- -i, --ips : tệp chứa các botnet trong thư mực bonesi (file.txt)
- -p, --protocol : giao thức truyền gói tin UDP,TCP,ICMP..
- -r, --send_rate : tốc độ phát gói tin đi (packets/s).
- -d, --device: Cổng đầu ra khi phát lưu lượng
- <dst_ip:port> : địa chỉ IP của máy chủ nhận gói tin tới,kèm theo cổng.

Câu lệnh được sử dụng trong mô phỏng :

$:sudo bonesi –i 50k-bots –d eth3 –r 1500 –p tcp 192.168.20.30:80

Ta có thể hiểu câu lệnh trên như sau, phát lưu lượng từ máy phát qua cổng eth3 của máy phát. Trong trường hợp này, chúng ta đang giả lập kẻ tấn công tạo ra một mạng botnet với 50000 botnet với tốc độ phát gói là 1500 gói trên giây, giao thức được sử dụng là giao thức TCP, địa chỉ của nạn nhân là 192.168.20.30, cổng trên nạn nhân là cổng 80.

b) Wireshark.

Wireshark là một công cụ rất phổ biến trong lĩnh vực mạng thông tin. Wireshark có giao diện dễ nhìn, cung cấp cho người dùng đầy đủ các thông tin về các gói tin đi qua một cổng vật lý (thậm chí cả cổng ảo). Hơn nữa, Wireshark cũng tích hợp thêm các phần mềm phụ trợ theo kèm khác, bao gồm editcap – là công cụ hỗ trợ được dùng phổ biến nhất. Với rất nhiều thông tin mà công cụ này có thể cung

cấp, sau đây là một số thông số cơ bản và quan trọng nhất khi chúng ta bắt những gói tin bằng công cụ này.

Time: Thời gian tương đối (kể từ thời điểm bắt đầu bắt gói tin, có thể chuyển sang thời gian tuyệt đối bao gồm, ngày giờ,…)

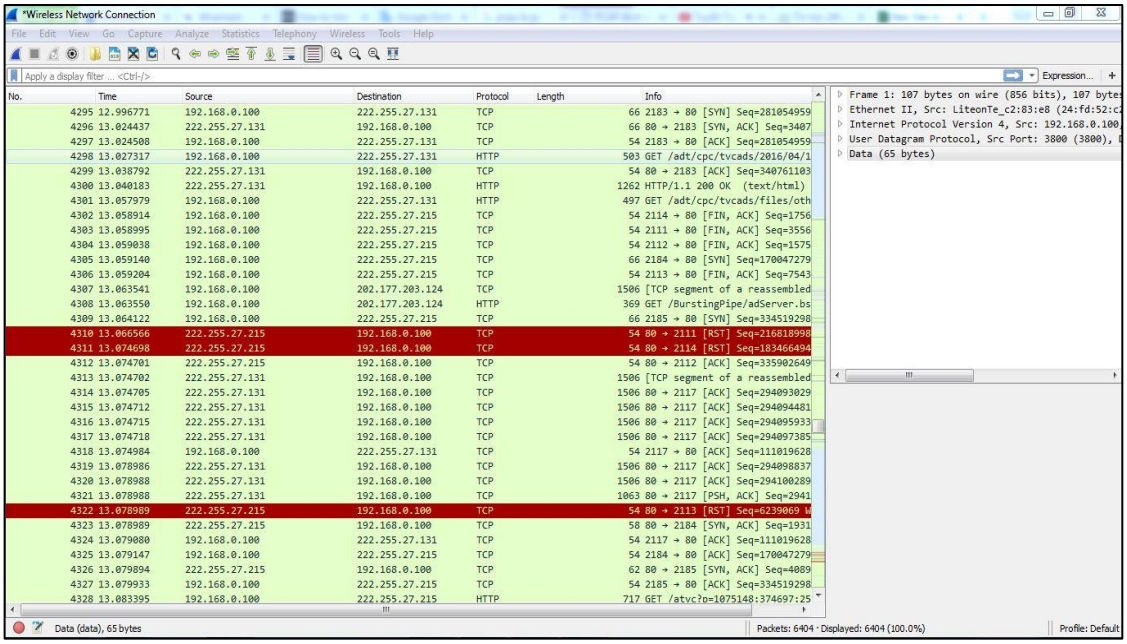
Source: Địa chỉ IP nguồn

Destination: Địa chỉ IP đích

Protocol: Giao thức của gói tin

Length: Chiều dài gói tin

Info: Thông tin chi tiết hơn về gói tin



Hình 2.7. Giao diện phần mềm Wireshark

Phần mềm bao gồm các bộ lọc gói tin, bảng mã màu quy chuẩn thực hiện bắt và hiển thị gói tin theo một định dạng mà người dùng có thể dễ dàng đọc được các thông số. Bộ lọc này của Wireshark vô cùng linh hoạt, cho phép người dùng có thể lọc theo bất cứ thông số nào, miễn là nó bao gồm trong bộ công cụ này. Trong hệ thống giả lập này tác giả sử dụng wireshark để bắt các gói tin trong môi trường botnet được phát bằng Bonesi sau đó lưu lại thành tập pcap để tạo ra một bộ lưu lượng tấn công, phục vụ cho việc chạy kịch bản phát và phòng chống tấn công DNS sau này. Sau khi đã có tệp dữ liệu tấn công, tệp này tiếp tục được điều chỉnh sao cho

phù hợp với nhu cầu tấn công. Ở đây hai công cụ được sử dụng là editcap của Wireshark và TCPReplay.

c) TCPReplay

Là một công cụ để phát lại gói tin dưới dạng file pcap. Sau khi thu thập được các file lưu lượng tấn công nhờ Bonesi, TCPReplay sẽ là công cụ hỗ trợ để phát lại lưu lượng đã thu được. Ngoài ra, nó còn cho phép người dùng chỉnh sửa, phân loại lưu lượng như máy chủ hoặc khách hàng. TCPReplay có rất nhiều sự lựa chọn khác nhau cho người dùng có thể tuỳ chỉnh cách phát lưu lượng tấn công. Sau đây là một số thông số quan trọng mà ta quan tâm trong mục đích phát lưu lượng tấn công khuếch đại DNS.

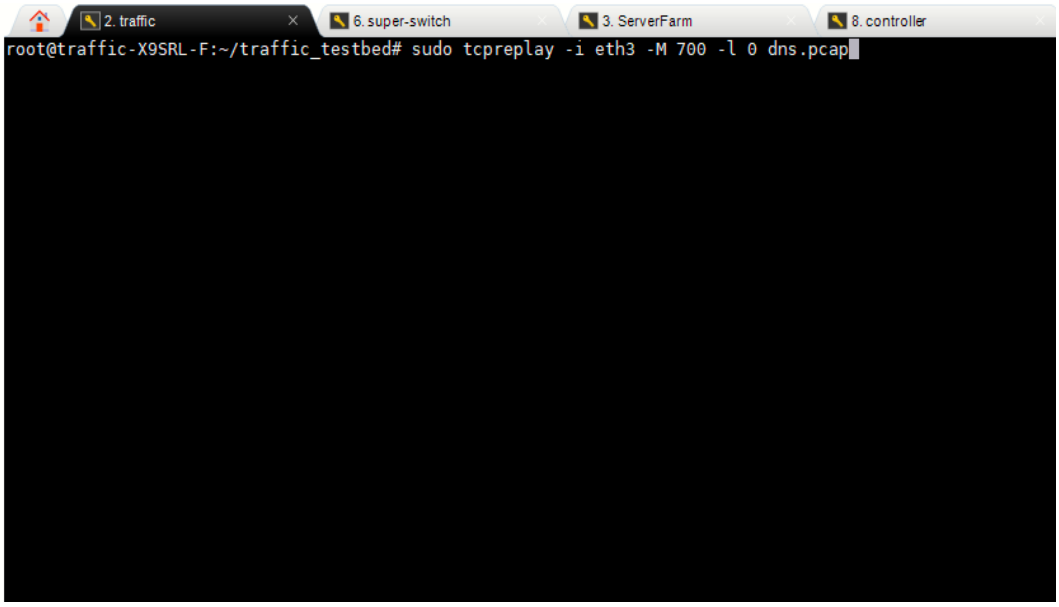
-i: Cổng ra của lưu lượng khi phát giả lập tấn công

-M: phát lại gói tin với tốc độ cho trước (Mbps)

-l: phát lại file pcap sau bao nhiêu ms

Ví dụ, trong khi giả lập tấn công ta sử dụng câu lệnh sau:

Sudo tcpreplay -i eth3 -M 500 -l 0 dns.pcap



Hình 2.8. Cửa sổ sử dụng TCPReplay để phát lại gói tin

Ta có thể hiểu câu lệnh trên như sau, chúng ta sẽ phát lại file dns.pcap qua cổng eth3 của Traffic Generator với tốc độ phát gói lên tới 500 Mbps và lặp lại sau

0ms, có nghĩa là file pcap sẽ được phát lặp lại ngay lập tức sau khi nó vừa kết thúc. Ngoài việc có nhiều sự lựa chọn cho việc phát lại dữ liệu tấn công, TCPReplay còn có hỗ trợ thêm một số tính năng vô cùng tiện ích như sau:

- TCPReplay : phát lại các gói tin vào hệ thống mạng.

- TCPbridge : tạo cầu kết nối hai đoạn mạng với nhau.

- TCPcapinfo: phân tích các bug của tcprewrite hay như các gói tin pcap bị lỗi.

- TCPrewrite: chỉnh sửa thông tin header của gói tin.

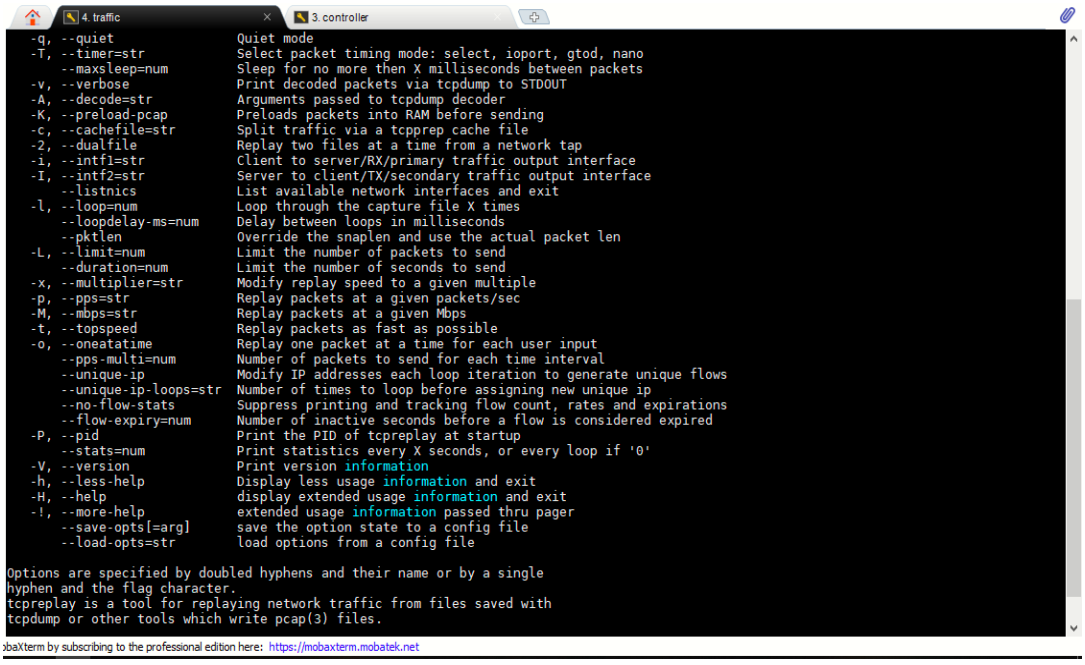
- TCPprep: xử lý các tập tin định dạng pcap, cho phép bóc tách gói tin.

- TCPplieplay: cho phép phát các gói tin truyền theo giao thức truyền thông TCP vào mạng internet, thiết lập quá trình bắt tay ba bước.

Để biết thông tin chi tiết về các tham số cũng như cách sử dụng, người dùng có thể dễ dàng tra cứu trợ giúp của người phát triển, bằng cách dùng câu lệnh:

Tcpreplay -help

Sau khi sử dụng câu lệnh trên, màn hình trợ giúp của tcpreplay sẽ hiện ra như trên hình 2.9:

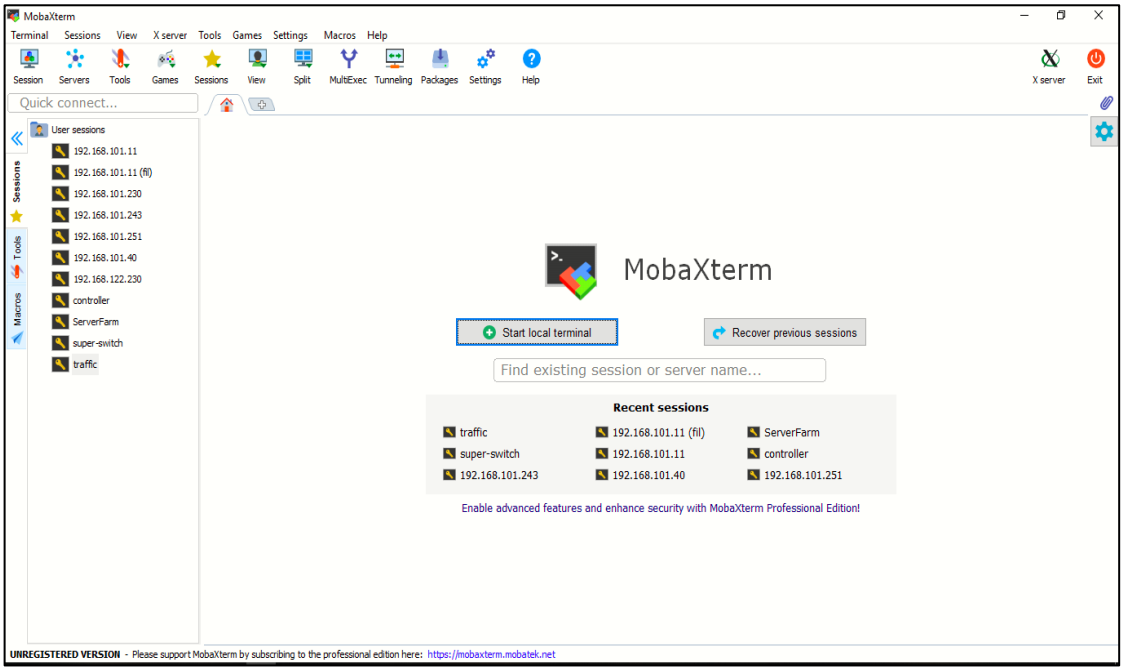


Hình 2.9. Cửa sổ sử dụng TCPReplay để phát lại gói tin

TCPRplay sẽ được tích hợp cài trên bộ phát lưu lượng Traffic-generator đóng vai trò là kẻ tấn công. Với nhiều lựa chọn trong một câu lệnh phát lưu lượng, ta có thể tùy chỉnh các thông số để tạo ra các kịch bản phát lưu lượng với tốc độ, dung lượng khác nhau.

d) Moba Xterm

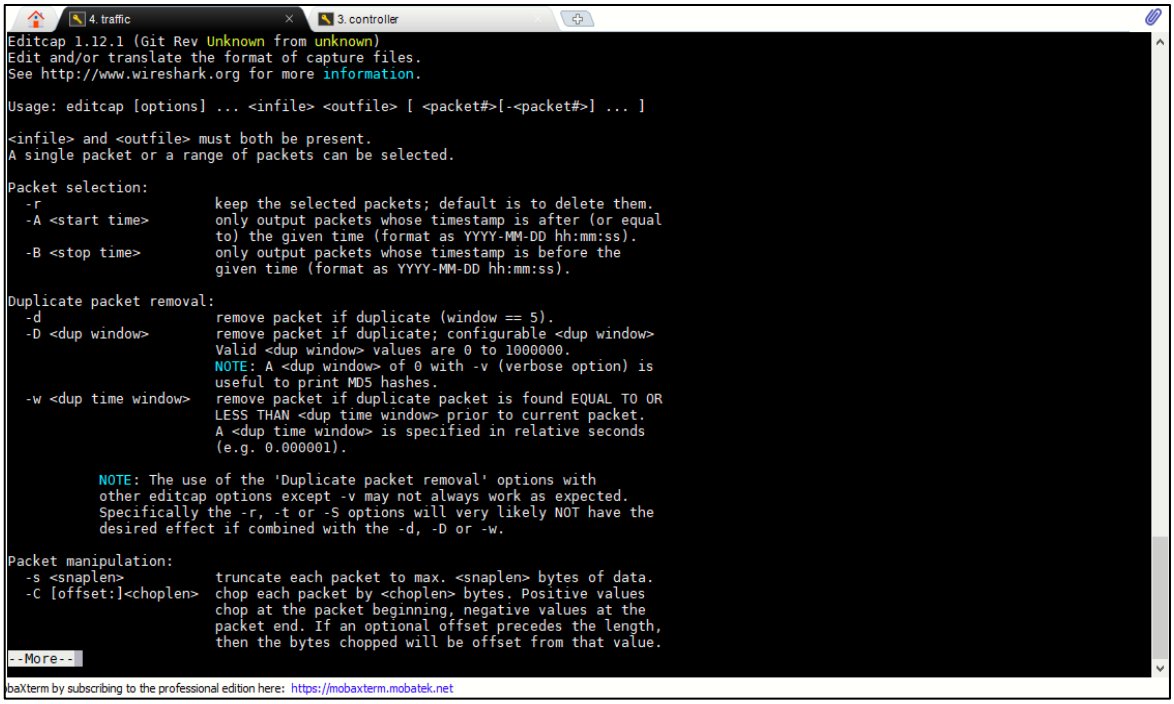
Hệ thống được xây dựng dựa trên tài nguyên của từng Server. Để thuận tiện cho việc điều khiển các Server từ xa thì tất cả các Server đều được kết nối với mạng LAN qua hệ thống các Router, Switch và Modem. Tất cả các Server trên hệ thống trên đều nằm chung một giải mạng và được điều khiển từ xa qua máy tính cùng dải mạng thông qua phần mềm MobaXterm. Việc truy cập thống nhất các máy tính giúp cho việc điều khiển hệ thống rất dễ dàng hơn nữa còn cho ta cái nhìn tổng quát về mô hình thử nghiệm. Trên Hình 2.10 là màn hình khởi động của phần mềm MobaXterm, nơi ta có thể thông qua SSH để truy cập vào các Server.



Hình 2.10. Màn hình khởi động MobaXterm

e) Editcap

Editcap là một công cụ đi kèm với Wireshark nhưng công dụng của nó cũng rất lớn. Với tác dụng chỉnh sửa file pcap theo nhu cầu của người dùng như: chia nhỏ file pcap, ghép file pcap,…



Hình 2.11. Màn hình trợ giúp của công cụ editcap

Ví dụ, để tách file dns.pcap thành những file nhỏ, mỗi file dài 20s. Ta sử dụng câu lệnh như sau:

Sudo editcap -i 20 dns.pcap dsn_output.pcap

Một số các thông số quan trọng của editcap là:

-i: chia file pcap sau bao nhiêu giây

-c: chia file pcap thành các file nhỏ, mỗi file nhỏ chứa bao nhiêu packet, …

f) Speedometer

Speedometer là một công cụ nhỏ trong các phiên bản khác nhau của hệ điều hành Ubuntu. Do kích thước nhỏ gọn nên việc cài đặt vô cùng dễ dàng và nhanh chóng, phù hợp với các điều kiện đo đạc khác nhau. Công dụng chính của công cụ này là đo đạc lưu lượng qua một cổng nào đó trên máy tính đang được cài đặt

Speedometer. Để thuận tiện cho người dùng, Speedometer cho phép người dùng đo cả lưu lượng ra và vào một cổng và thậm chí cả cổng ảo (tạo ra bởi các mô hình chứa OpenFlow Switch). Chính vì vậy khi không cần quan tâm quá nhiều đến các thông số khác ngoài lưu lượng qua cổng thì Speedometer là một công cụ hoàn hảo.

Không giống như Wireshark chạy trên nền tảng các hệ điều hành có giao diện, Speedometer chạy được trên cả các nền tảng chỉ có màn hình commandline (cụ thể là Ubuntu Server, hay là SSH qua MobaXterm).

Tương tự như các công cụ khác chạy trên Ubuntu, Speedometer cũng cho phép người dùng truy vấn trợ giúp thông qua câu lệnh: *sudo speedometer -help,* kết quả thu được như trên Hình 2.12.

```
Usage: speedometer [options] tap [[-c] tap]...
Monitor network traffic or speed/progress of a file transfer.  At least one
tap must be entered.  -c starts a new column, otherwise taps are piled
vertically.

Taps:
  -f filename [size]          display download speed [with progress bar]
  -r network-interface        display bytes received on network-interface
  -t network-interface        display bytes transmitted on network-interface
  -c                          start a new column for following tap arguments

Options:
  -b                          use old blocky display instead of smoothed
                              display even when UTF-8 encoding is detected
                              (use this if you see strange characters)
  -i interval-in-seconds      eg. "5" or "0.25"   default: "1"
  -k (1|16|88|256)            set the number of colors this terminal
System Settings               supports (default 16)
  -l                          use linear charts instead of logarithmic
                              you will VERY LIKELY want to set -m as well
  -m chart-maximum            set the maximum bytes/second displayed on
                              the chart (default 2^32)
  -n chart-minimum            set the minimum bytes/second displayed on
                              the chart (default 32)
  -p                          use original plain-text display (one tap only)
  -s                          use bits/s instead of bytes/s
  -x                          exit when files reach their expected size
  -z                          report zero size on files that don't exist
                              instead of waiting for them to be created

Note: -rx and -tx are accepted as aliases for -r and -t for compatibility
with earlier releases of speedometer.  -f may be also omitted for similar
```

Hình 2.12. Giao diện trợ giúp của Speedometer

Trong đó, thông thường chúng ta quan tâm đến:

-t: đo lưu lượng đi ra khỏi cổng

-r: đo lưu lượng đi vào cổng

Ví dụ:

Speedometer -t eth3 -r eth5

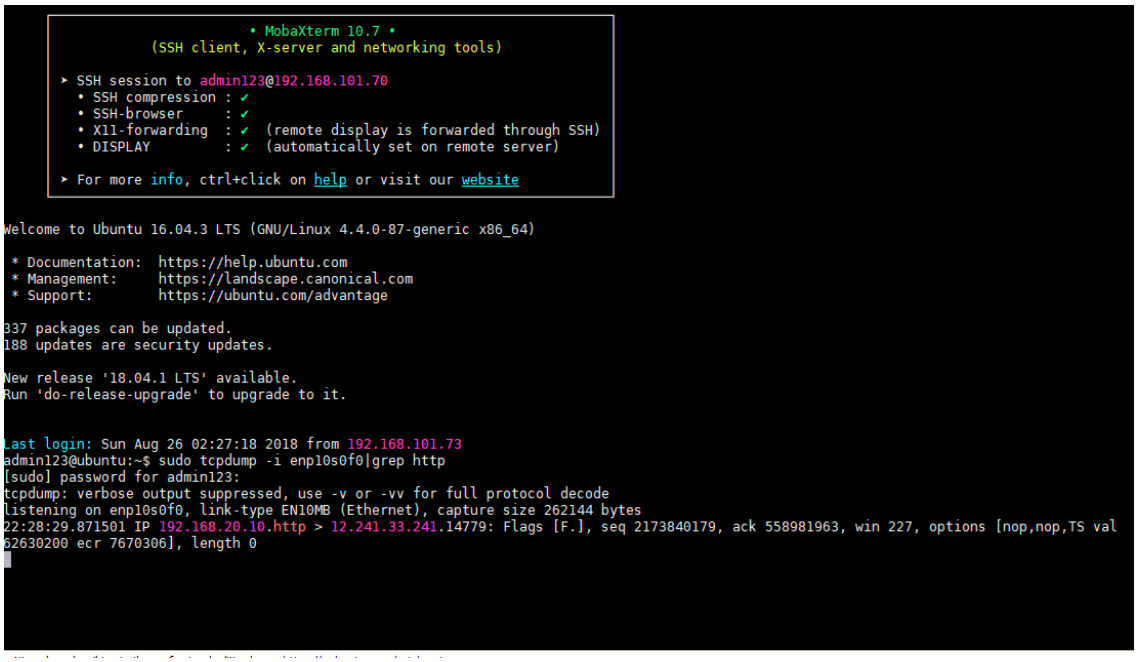
Câu lệnh trên có nghĩa là chúng ta đang muốn đó lưu lượng ra ở giao diện cổng eth3 và lưu lượng vào ở giao diện cổng eth5.

g) Tcpdump

Tcpdump là một công cụ đo lưu lượng và các thông số của gói tin khi đi qua một hay nhiều cổng trên một máy nào đó. Chúng ta có thể coi Tcpdump là kết hợp giữa Wireshark và Speedometer bởi vì Tcpdump không cần chạy trên các hệ điều hành có giao diện giống như Speedometer nhưng cũng mang rất nhiều các thông số của gói tin giống như Wireshark. Chính vì vậy, trong những trường hợp cụ thể, người dùng có thể linh hoạt giữa ba phần mềm này để thu được hiệu quả đo đạc cao nhất. Câu lệnh hoạt động của Tcpdump cũng rất đơn giản, ví dụ như sau:

Tcpdump -i eth3 | grep dns

Tức là chúng ta muốn thống kê các gói tin qua giao diện cổng eth3 và những gói tin chứa thông tin "dns".



Hình 2.13. Giao diện hoạt động của Tcpdump

Trên Hình 2.13 là giao diện các gói tin được bắt bằng tcpdump. Có rất nhiều thông tin khác nhau mà người dùng có thể lựa chọn để thu thập, tuỳ vào mục đích sử dụng.

2.3.3. Kịch bản phát tấn công

Sử dụng công cụ Bonesi để phát tấn công trực tiếp vào FPT Server đóng vai trò là máy nạn nhân, đồng thời dùng Wireshark để thu thập dữ liệu lưu lại dưới dạng file pcap. Tác giả đã có được một bộ dữ liệu tấn công giống đặc tính của mạng Botnet. Sử dụng TCPReplay để phát lại bộ lưu lượng đó từ máy phát lưu lượng vào hệ thống giả lập đã xây dựng, tiến hành tấn công máy FPT Server. Lưu lượng phải được chuyển tiếp qua OpenFlow Switch và được kiểm soát bởi controller. Tiến hành phát và ghi lại kết quả trong hai trường hợp : không chạy giải pháp giảm thiểu tấn công và trường hợp có chạy giải pháp giảm thiểu tấn công trên Floodlight Controller. Sử dụng công cụ Speedometer để đo thông lượng của lưu lượng ở đầu vào cũng như đầu ra của khối chuyển mạch và công nghệ sFlow để giám sát lưu lượng. Từ đó có thể thấy được rõ hình thức tấn công và hiệu năng của giải pháp được sử dụng.

Giải pháp được đưa ra ở đây là dùng ngưỡng xác định, khi phát hiện tấn công, Controller sẽ gửi bản tin Flowmod để drop tất cả các bản tin DNS Response. Trên khối Floodlight Controller có phát triển module phát hiện tấn công DNS với cơ chế là dùng ngưỡng xác định. Port 53 (DNS sử dụng giao thức TCP và UDP với Port giao tiếp là Port 53) sẽ được đóng để chặn các bản tin DNS Response trong khoảng time idle time out của mội flow entry (thường là 5s). Trong vòng 5s từ lúc đóng port, nếu có bản tin DNS Response đi tới thì chu kỳ 5s lại được lặp lại từ đầu, vì thế khi có tấn công xảy ra sẽ đảm bảo tất cả các bản tin DNS Response bị chặn lại và drop hết ở port 53, lưu lượng ra được đảm bảo.

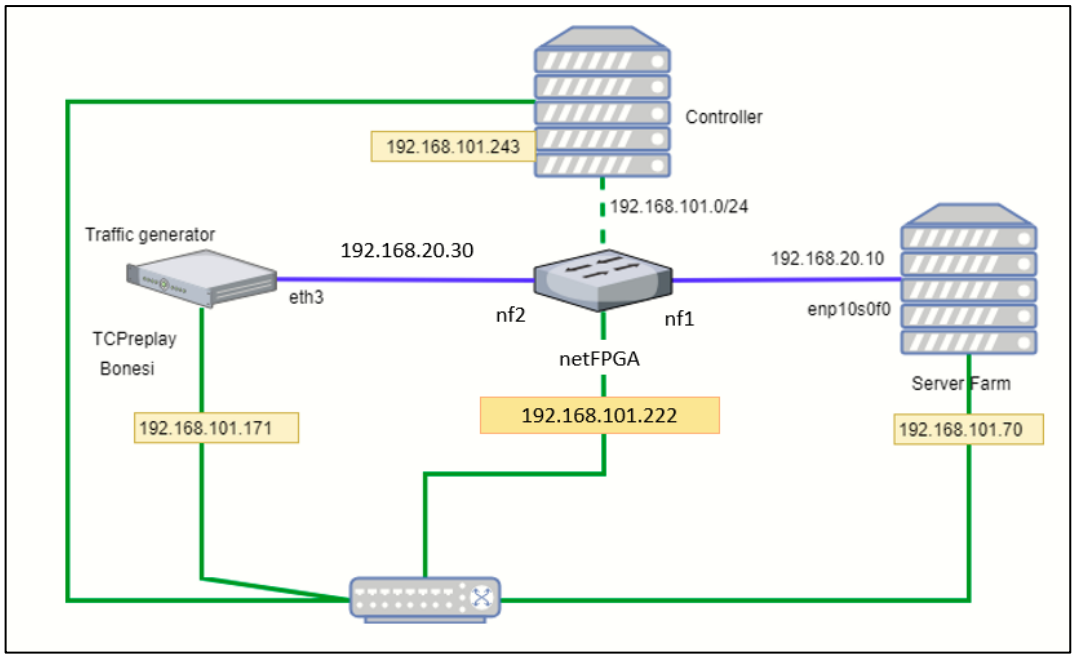
CHƯƠNG 3. KẾT QUẢ MÔ PHỎNG CHỐNG TẤN CÔNG TRONG SDN

Trong chương này, tác giả sẽ trình bày về những kết quả đạt được trong việc thực hiện mô phỏng trong chương 2 về việc xây dựng kiến trúc mạng SDN/OpenFlow thực hiện phòng chống tấn công DNS và các giải pháp được thực hiện. Đồng thời đề xuất hướng phát triển sau này.

3.1. Mô hình xây dựng hệ thống

3.1.1. Tổng quan hệ thống

Tác giả đã xây dựng được mô hình giả lập tấn công như đã nghiên cứu phần lý thuyết. Hình 3.1 cho thấy được các khối chức năng trong hệ thống giả lập.



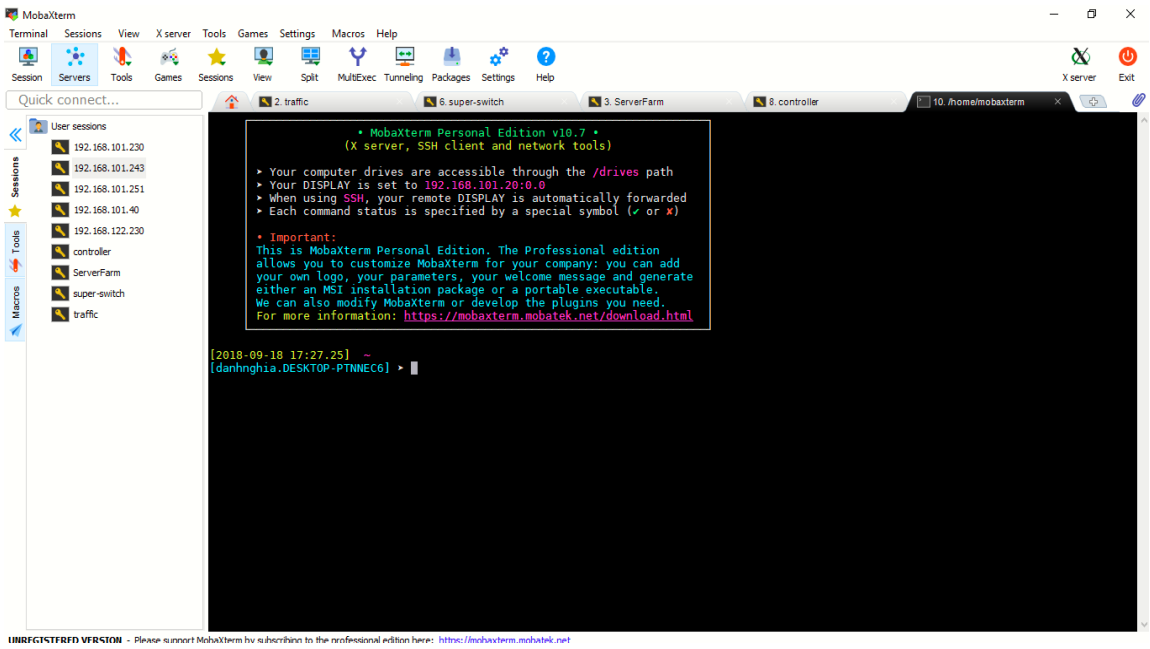
Hình 3.1. Mô hình lý thuyết

Như ta thấy chú thích cụ thể trên hình là các khối thiết bị chức năng trong hệ thống bao gồm :

- Floodlight Controller.
- OpenFlow Switch.
- Traffic Generator.

- Victim Server.

Các khối này nằm tập trung tại Security Rack, chính vì vậy để việc truy cập và điều khiển vào từng khối chức năng trong hệ thống được dễ dàng, tác giả đã sử dụng phần mềm Moba Xterm hoạt động dựa trên giao thức SSH trong dải mạng LAN, trực tiếp SSH vào các khối, điều khiển tập trung tất cả các khối ngay trên một máy tính laptop hoặc PC rời khối, có cùng kết nối mạng với hệ thống. Dễ dàng điều khiển và giám sát hệ thống, giảm thiểu đc thiết bị phần cứng và không gian.



Hình 3.2. Giao diện phần mềm Moba Xterm

Các khối trong mô hình có một địa chỉ IP riêng, nhờ vào đó sử dụng giao thức SSH ta có thể truy cập và điều khiển một cách tổng quát tất cả các khối trong hệ thống. Địa chỉ IP ứng với từng khối:

- Controller Floodlight (controller): 192.168.101.243
- OpenFlow Switch (super-switch): 192.168.101.222
- Traffic Generator (Traffic): 192.168.101.171
- Victim (Serverfarm): 192.168.101.70

3.1.2. Triển khai hệ thống

Traffic Generator: Chúng ta tiến hành bật traffic generator lên để chuẩn bị cho việc phát lưu lượng mẫu. Kiểm tra lại các kết nối mạng từ traffic generator ra mạng internet để bắt đầu dùng MobaXterm để SSH vào điều khiển máy này.

OpenFlow Switch (super-switch): Sau khi tiến hành bật máy lên, chúng ta phải tiến hành nạp code cho NetFPGA, bước đầu là để máy tính có thể nhận các cổng của NetFPGA là cổng của máy.

$: sudo -i

$:./nic1.sh

$: reboot

Sau khi tiền hành nạp file code đầu tiên, chúng ta phải khởi động lại máy để máy hoàn thành tác vụ chuyển các cổng của FPGA thành cổng của máy. Sau khi máy bật lên trở lại, chúng ta tiến hành nạp file code thứ hai

$: sudo -i

$:./nic2.sh

Sau khi chạy xong file nạp code trên, chúng ta đã cấu hình để máy tính này có thể hoạt động như một OpenFlow Switch. Chúng ta kiểm tra lại hệ thống bằng lệnh:

$: sudo ovs-vsctl show

```
526d885c-cb18-4204-9a13-04218e171a9f
    Bridge ovs
        Controller "tcp:192.168.101.243:6653"
            is_connected: true
        Port "nf2"
            Interface "nf2"
        Port "nf1"
            Interface "nf1"
        Port "nf3"
            Interface "nf3"
        Port "nf0"
            Interface "nf0"
        Port ovs
            Interface ovs
                type: internal
    ovs_version: "2.5.4"
```

Hình 3.3. Kết quả sau khi nhập code

Nếu kết quả chạy ra đúng các cổng như mô hình đã định sẵn (như trên Hình 3.3) thì chúng ta đã nạp code thành công. Giờ chúng ta có thể dùng MobaXterm để truy cập và điều khiển máy này.

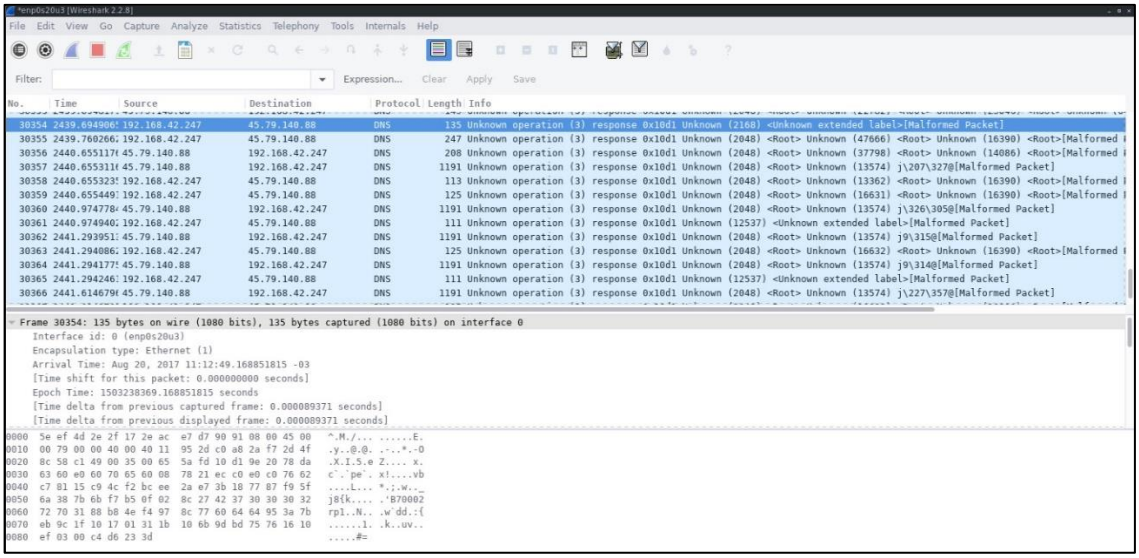
ServerFarm: Sau khi máy ServerFarm được bật, chúng ta tiến hành bật Wireshark lên và chuẩn bị tiến hành đo lưu qua cổng enp10s0f0.

3.2. Mô phỏng tấn công và biện pháp giảm thiểu tấn công

Trước tiên sử dụng Bonesi để phát lưu lượng giả lập, trong đó chứa các nguồn địa chỉ IP khác nhau từ file 50k-bots với tối độ 1500 gói/s tới địa chỉ của Victim 192.168.20.30

Câu lệnh phát : *$:sudo bonesi –i 1k-bots –d eth3 –r 1500 –p udp 192.168.20.30:80*

Đồng thời dùng phần mềm wireshark thu lại lịch sử của những kết nối trong khi phát tấn công thành file dns.pcap.



Hình 3.4. Các gói tin thu được trên Wireshark

Sau khi thu được file dns.pcap đóng vai trò như file chứa lưu lượng tấn công cho mô hình giả lập. Tiếp theo tác giả sẽ sử dụng công cụ TCPReplay để phát lại file dns.pcap kia vào mô hình.

3.2.1. Phát lưu lượng bình thường không có giải pháp giảm thiểu

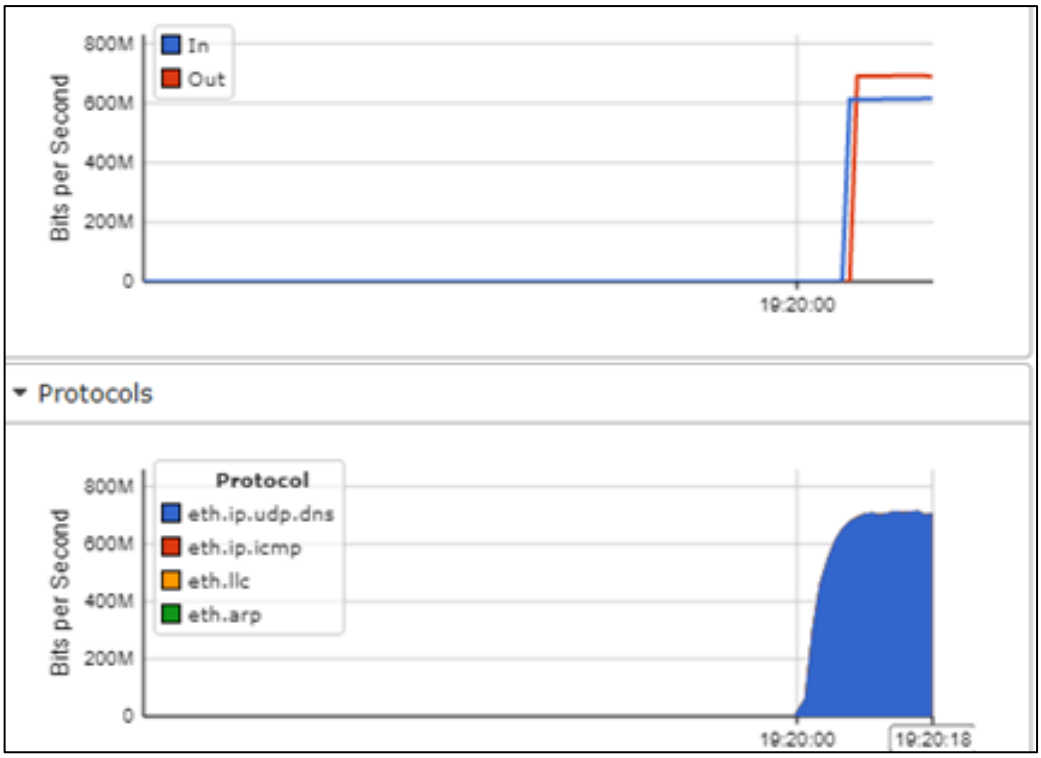
Trước khi tiến hành phát lưu lượng, chúng ta tiến hành bật controller và tiến hành gửi lưu lượng từ OpenFlow Switch lên controller. Trước tiên, trên controller, ta chạy controller bằng cách:

$: cd /Downloads/idea-IC-173.4548.28/bin

$:./idea.sh

Sau đó, chúng ta tiến hành chạy controller mà trên đó không có bất cứ giải pháp giảm thiểu nào.

Thực hiện phát lưu lượng vào hệ thống giả lập từ máy traffic gerenator, đồng thời chưa chạy các chương trình giảm thiểu tấn công trên controller. Lúc này luồng lưu lượng sẽ đi từ máy phát tới OpenFlow Switch, trao đổi với controller. Vì hiện tại trên controller chưa bật chức năng phát hiện và giảm thiểu tấn công cho nên toàn bộ lưu lượng đi vào sẽ được chuyển tiếp hoàn toàn sang phía Victim, khi đó máy nạn nhân đang hứng chịu tấn công.

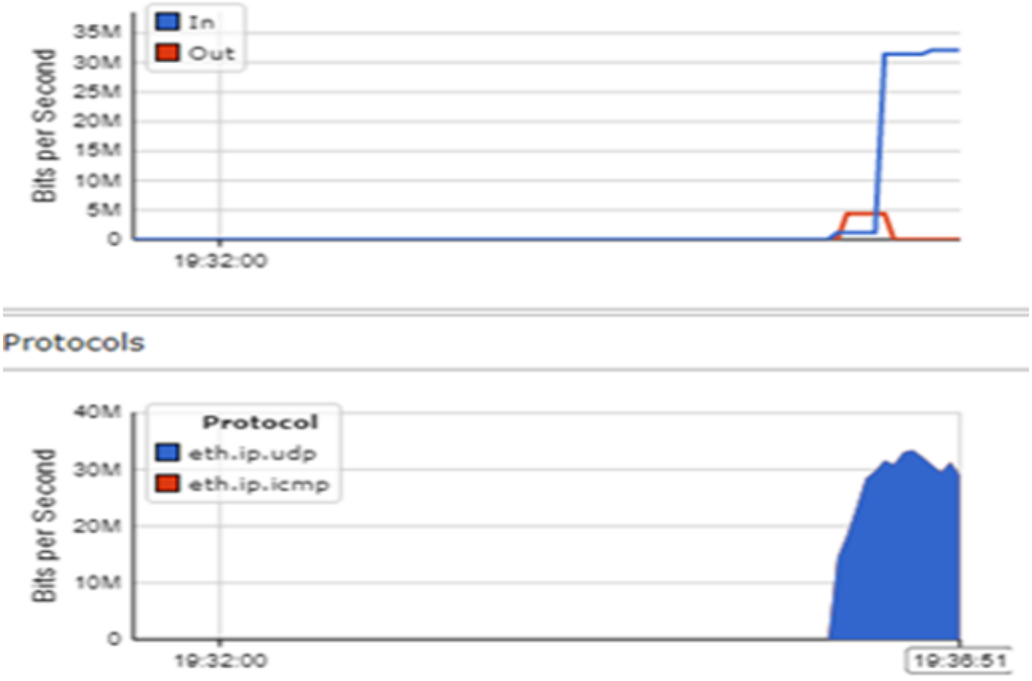


Hình 3.5. Lưu lượng tấn công khi chưa chạy giải pháp giảm thiểu

3.2.2. Hệ thống khi sử dụng giải pháp giảm thiểu

Quy trình tương tự như khi phát lưu lượng bình thường, nhưng lúc này trên SDN Controller ta bắt đầu chạy giải pháp phát hiện và giảm thiểu tấn công. Như đã trình bày ở chương trước, giải pháp được đưa ra ở đây khi có tấn công là đóng port 53 khi luồng lưu lượng tới vượt quá ngưỡng cho phép, cứ 5s lấy mẫu theo dõi một lần.

Từ hình 3.5 và 3.6 cho ta thấy sự khác biệt giữa trường hợp sử dụng giải pháp phát hiện và giảm thiểu tấn công và trường hợp không sử dụng. Ta thấy rằng, đối với file 50k-bots với tối độ 1500 gói/s khi không sử dụng giải pháp giảm thiểu tấn công, số lượng flow-entry trên switch rất lớn và lớn nhất khoảng 700Mb/s và thời gian tồn tại của flow entry trên Switch kéo dài. Điều này có thể gây tốn tài nguyên trên Switch và nếu tấn công với cường độ lưu lượng lớn có thể dẫn tới Switch sẽ không còn khả năng xử lý. Trong đó khi sử dụng giải pháp giảm thiểu tấn công với file 50k-bots với tối độ 1500 gói/s thì số lượng flow-entry trên Switch rất ít khi tấn công xảy ra và đạt giá trị lớn nhất khoảng 5Mb/s. Ngoài ra trạng thái flow tồn tại trên Swich cũng ngắn hơn so với trường hợp không sử dụng giải pháp giảm thiểu.



Hình 3.6. Lưu lượng tấn công khi chạy qua giải pháp giảm thiểu

3.3. Nhận xét và kiến nghị

Hệ thống giả lập được xây dựng khá hiệu quả trong việc phát hiện và phòng chống tấn công. Dựa trên kiến trúc mạng SDN/OpenFlow cùng với công nghệ sFlow, hệ thống đã phát hiện được các truy vấn giả mạo để thực hiện ngăn chặn tấn công trong một thời gian ngắn, nhanh chóng giảm thiểu được các flow entry đi vào bộ chuyển mạch OpenFlow Switch, giúp Server nhanh phục hồi để phục vụ các dịch vụ thông thường ngay trong khi cuộc tấn công xảy ra.

Qua đề tài luận văn này, ta nhận thấy hiệu quả trong việc phát hiện và phòng chống tấn công của hệ thống giả lập. Tác giả xin được kiến nghị được thử nghiệm mô hình hệ thống này vào các hệ thống SDN đang hoạt động tại các đơn vị trong thời gian tới.

KẾT LUẬN

Các kết quả chính của đề tài luận văn:

- Đưa ra cái nhìn tổng quan về kiến trúc mạng SDN, giao thức OpenFlow và các bản tin trao đổi giữa OpenFlow Switch và Controller.

- Xây dựng kiến trúc mạng SDN/OpenFlow trên server testbed. Đồng thời giả lập một cuộc tấn công trên hệ thống mô phỏng, các kỹ thuật giám sát lưu lượng và giảm thiểu tấn công đã được tích hợp.

- Hệ thống mô phỏng đã phát hiện và ngăn chặn cuộc tấn công trong thời gian ngắn, giúp server nhanh chóng phục hồi để phục vụ các dịch vụ ngay trong khi cuộc tấn công xảy ra.

Hướng phát triển của đề tài:

Thử nghiệm mô hình với các bộ dữ liệu đã được thu thập từ thực tế chứa nhiều hình thức tấn công hơn, để có thể đánh giá hiệu năng của hệ thống giả lập cũng như phát triển các giải pháp phòng chống các loại tấn công khác trên SDN Controller. Đồng thời, nâng cấp cấu hình Controller để tăng tốc độ xử lý các gói tin trong các cuộc tấn công, giảm thời gian đáp ứng của hệ thống, tối ưu hóa hiệu năng của hệ thống, cung cấp một hệ thống hoàn thiện.

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. Thuyết minh dự thảo tiêu chuẩn quốc gia – Các yêu cầu và hướng dẫn bảo mật DNS (DNSSEC)-2016.

[2]. Open Networking Foundation, Software-Defined Networking: The New Norm for Networks, April 13, 2012

[3]. OpenFlow Specification v1.3.0, June 25, 2012

[4]. Marios Anagnostopoulos, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, Stefanos Gritzalis, DNS Amplification Attack Revisited, An article in Computer&Security, December 2013

[5]. FERGUSON, P., AND SENIE, D.BCP 38 - Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827, May 2000.

[6]. Muhammad Afaq, Shafqat Rehman, Wang-Cheol Song, Large Flows Detection, Marking, and Mitigation based on sFlow Standard in SDN, Journal of Korea Multimedia Society Vol. 18, No. 2, February 2015 (pp. 189-198).