

IPv6와 IPv4 환경에서 IPTV 전송 최적화 위한 SDN을 응용한 MPLS

황성규*

Convergence of MPLS applied SDN to Optimize IPTV Broadcast Transmission for IPv6 and IPv4

Seong-kyu Hwang*

Department of Information & Communication Engg., Chosun University, Gwangju 501-759, Korea

요 약

SDN(Software Defined Network)기술은 네트워크 제어기능이 기존의 스위치나 라우터 등의 하드웨어와 별도로 분리되어야 하고 데이터 전달기능과도 역시 분리되어 개발 및 실행될 수 있는 네트워크 구조를 말한다. 네트워크를 계획하고 디자인할 때에는 SDN 기술을 적용하여 반영해야 자원을 효율적으로 반영할 수 있다. 본 연구에서는 일명 Applications SDN을 제안하며 이는 MPLS(Multiprotocol Label Switching) IOS의 절대적으로 필요한 부분을 코딩한 프로그램과 기존 IOS를 결합한 이점(Combined Benefits)을 통해 Controller 부분을 완성하여 적용한 모델을 제안한다. OpenFlow Controller는 LER(Label Edge Router)에 명령을 하고 라우터는 그 명령에 따라 패킷을 EXP bit에 따라 목적지로 전송한다. TCP/IP를 코딩을 통해 최적화된 IOS로 구성한다. OpenFlow 프로토콜을 이용하여 컨트롤러는 MPLS의 정의된 헤더를 패킷에 캡슐화 하여 전송한다.

ABSTRACT

SDN (Software Defined Network) technology, hardware and network control functions should be separated independently of existing switches or routers and data transfer functions and are also refers to a separate network architecture that can develop and run. When planning a network and designed to reflect the network control function by applying the SDN may reflect the resources efficiently. In this study, one people suggest Applications SDN, which offer an absolutely necessary part of a coded program that combines the benefits of the existing IOS (Combined Benefits) applying the model to complete the Controller part through the MPLS IOS. The OpenFlow Controller is a command to the router LER(Label Edge Router), and transmits the packet in accordance with the command to the destination according to the EXP bit. To configure TCP / IP to the IOS through the optimized coding. Using the OpenFlow protocol controller transmits an encapsulation header as defined in the MPLS packet.

키워드 : SDN, MPLS, 멀티캐스트, 오픈플로우

Key word : SDN, MPLS, Multicast, OpenFlow

Received 10 February 2015, Revised 28 February 2015, Accepted 12 March 2015

* Corresponding Author Seong-Kyu Hwang (E-Mail: okhsk9@daum.net, Tel:+82-62-230-7247)

Department of Information & Communication Engg., Chosun University, Gwangju 501-759, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.4.819>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

현재의 인터넷이 가지고 있는 네트워크 구조는 단순하지만 관리가 복잡하고 인터페이스는 어려움이 있다. 그 이유는 인터넷의 코어(Core)를 구성하는 라우터나 스위치들이 이른바 완전히 닫혀 있어서 그 위에 새로운 소프트웨어나 프로그램을 실현하는 것이 원천적으로 봉쇄되어 있다[1]. 이에 SDN(Software Defined Network)기술은 네트워크 제어기능이 기존의 스위치나 라우터의 하드웨어와 별도로 분리되어야 하고 데이터 전달기능과도 역시 분리되어 개발 및 실행될 수 있는 네트워크 구조를 말한다[2]. 앞으로 네트워크를 계획하고 디자인할 때에는 네트워크 제어기능은 SDN을 적용하여 반영해야 자원을 효율적으로 반영할 수 있다[3]. 본 논문에서는 도래하는 IPv6 환경에 앞서 IPv6와 IPv4의 Dual Stack 구조에 대비하며 인터넷 기존망의 고유 특징인 Best Effect 환경에서 대용량 멀티미디어 데이터 전송의 QoS 보장과 하드웨어적 고속 스위칭을 위해 MPLS(Multiprotocol Label Switching) Label을 이용하여 packet을 forwarding 한다. 또한 최적화된 MPLS망에서 QoS 보장을 위해 IP Packet의 MPLS Label PUSH 과정에서 IP Header의 QoS Field(IP Pre/DSCP)의 첫 3bit를 MPLS Label의 EXP Field로 복사할 때 QoS 부분인 EXP Bit를 Expedited Forwarding으로 설정하는 방법과 앞으로 네트워크 구조가 SDN/OpenFlow로 발전해 가는 환경에서 SDN/OpenFlow를 응용한 MPLS를 제안한다.

II. SDN(Software Defined Network) 기술

SDN은 네트워크 제어 기능이 기존의 스위치나 라우터 등의 하드웨어와 별도로 분리 되어야 하고 데이터 전달 기능과도 분리되어 개발 및 실행될 수 있는 네트워크 구조를 말한다[4].

최근 이동통신 기지국시스템은 기지국의 디지털 신호처리부(DU)와 무선신호를 송수신하는 무선신호처리부(RU)를 분리해 DU는 전화국 사이에 집중배치하고 RU는 서비스 지역에 설치하는 무선망 기술로 무선데이터 트래픽을 효과적으로 수용하고 있다[5]. 캠퍼스 네트워크를 보면 백본에 각 교실의 스위치들이 연결되어

데이터들이 전송되고 있다. 그 구조 또한 SDN 구조의 일부라 생각된다. 백본에서의 제어를 통해 스위치들은 VLAN등의 정보를 받아 전달되기 때문이다.

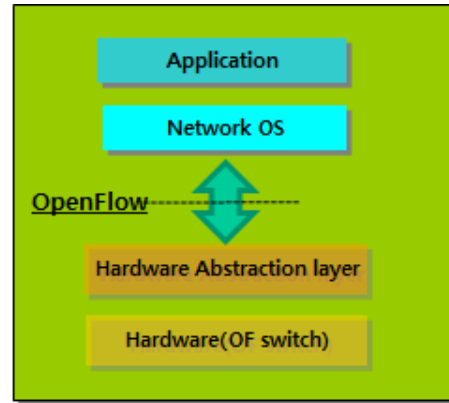


그림 1. SDN 개념
Fig. 1 SDN concept

OpenFlow 기술은 네트워크 라우터의 패킷 포워딩 기능과 제어 기능을 분리하고 이 두 기능간의 통신을 위한 개방형 표준 프로토콜을 말한다[6]. 그림 1은 머신과 네트워킹 스위치간의 통신을 담당하는 표준 인터페이스 기술인 OpenFlow를 기반으로 정의되고 있으며 일반적으로 SDN과 OpenFlow가 연결된 관계인 것으로 알려져 있으나 SDN은 그 하부 기술로 OpenFlow만을 한정하지는 않는다. SDN은 훨씬 더 큰 개념으로 네트워크 구조 또는 새로운 패러다임이며 OpenFlow는 SDN을 위한 인터페이스 기술 중 하나이다[7,8].

III. 제안하는 SDN/OpenFlow 기술을 응용한 MPLS

3.1. SDN을 응용한 MPLS

OpenFlow Controller는 LER(Label Edge Router)에 명령을 하고 라우터는 그 명령에 따라 패킷을 EXP bit에 따라 목적지로 전송한다. 그림 2는 LER에 OpenFlow 프로토콜을 펌웨어 추가하여 OpenFlow 라우터를 구성하고 Controller는 소프트웨어로 구현해야하는데 본 논문에서는 TCP/IP를 코딩을 통해 최적화된 IOS로 구성한다.

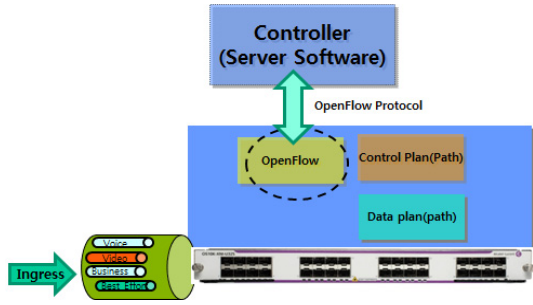


그림 2. OpenFlow controller 적용 MPLS LER 구성
Fig. 2 OpenFlow controller configuration applies to MPLS LER

OpenFlow 프로토콜을 이용하여 컨트롤러는 MPLS의 정의된 헤더를 패킷에 캡슐화 하여 전송한다. OpenFlow 컨트롤러의 주 역할은 MPLS 헤더정보의 EXP bit를 미리 코딩된 MPLS 헤더로 캡슐화 하여 MPLS 헤더기반으로 경로를 결정한다.

OpenFlow 컨트롤러에서 최단경로나 회선속도와 사용자가 지정하는 경로의 가중치나 여러 조건을 컨트롤러가 경로계산을 해야 하나 본 논문에서는 음성과 멀티미디어 데이터로 구성되어 민감한 음성은 EXP=5가 최적이어서 미리 코딩하여 정의해 둔다. 그 이유는 데이터의 우선순위에 따라 네트워크의 QoS를 제어할 경우는 네트워크에 장애나 음성의 데이터는 컨트롤러가 이를 판단하고 제어명령을 보낼 때까지 15ms를 초과하면 MPLS의 EXP bit가 서비스에 적용이 불가능해지기 때문이다.

3.2. Flow Table

OpenFlow 라우터는 다수의 플로우 테이블을 가질 수 있으나 본 논문에서는 하나의 플로우 테이블을 가진다. 플로우 테이블은 그림 3과 같이 플로우 엔트리(flow entry)로 구성되며 플로우 엔트리는 조건(Match fields)과 통계(counters)와 처리(action)의 정보로 구성된다.

헤더정보는 계층1의 물리적 포트 번호와 계층4의 TCP/UDP 포트 번호로 구성된다[5]. 처리에 실제 지정할 때에는 갯신 필드의 값을 구체적으로 적으며 패킷을 송출할 포트 번호와 MPLS EXP bit도 함께 지정할 수 있다. 통계는 트래픽의 양을 패킷의 수와 바이트 수로 나타낸다.

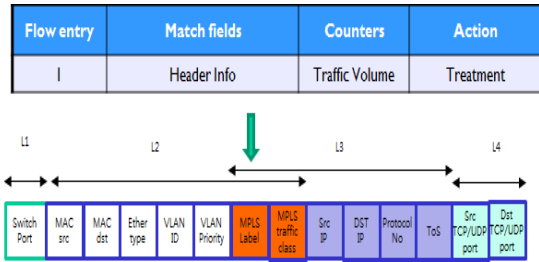


그림 3. OpenFlow v.1.1의 플로우 테이블 내용과 헤더 정보
Fig. 3 Flow table content and header information of the OpenFlow v.1.1

3.3. 파이프라인 처리(Pipelining)

LER내의 플로우 테이블에는 0의 번호가 부여되어 모든 입력 패킷은 0번의 플로우 테이블로부터 플로우 통계와 처리로 지정된 명령이 실행된다.

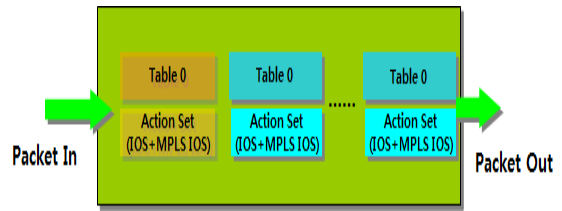


그림 4. 파이프라인 처리
Fig. 4 Pipeline processing

만약 다른 플로우 테이블이 있으면 엔트리와 비교하여 패킷 처리가 계속되도록 지시할 수 있다. 이것을 Pipelining 이라고 한다. 이 같은 그림 4의 처리는 모든 명령이 완수되어 최종적으로 패킷이 처리될 때까지 다음 테이블로 연속된다.

3.4. SDN을 응용한 MPLS (일명 Applications SDN)

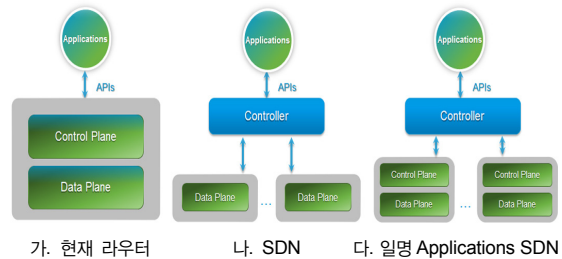


그림 5. SDN을 응용한 MPLS
Fig. 5 Application of SDN MPLS

그림 5의 현재 라우터모델을 나타내는 거의 그림은 SDN 개념을 도입하지 않는 현재의 라우터나 스위치의 구조모델이며 그림 나는 SDN개념을 적용한 간단한 프로비저닝과 중앙네트워크 토폴로지이다. 그리고 본문에서 적용한 일명 Applications SDN은 MPLS IOS의 절대적으로 필요한 부분을 코딩한 프로그램과 기존 IOS를 결합한 이점(Combined Benefits)을 통해 Controller 부분을 완성하여 적용한 모델이다.

IV. IPv6와 IPv4 환경에서 IPTV 방송전송 최적화 시뮬레이션 분석

4.1. 제안 구성 세부 구성도

제안 구성의 시뮬레이션 분석 세부 구성도는 재현 시뮬레이션 툴 GNS3로 그림 6과 같이 Dual Stack 환경에서 MPLS 구간, 가입자 구간, 멀티미디어 제공자 구간으로 구분하고 라우터는 총 9대를 사용하며 사용된 라우터의 IP Address를 설정하고 활성화 한다. 서버는 윈도우 2003 서버로 미디어 서비스를 구동하며 라우팅 프로토콜은 MPLS 망 내에서는 OSPF로 설정하였으며 CE-PE 구간 통신을 위해 라우팅은 BGP로 설정하였다. MPLS VPN에서 가입자 네트워크의 멀티캐스트를 지원하기 위해 백본망 자체, 즉 MPLS구간에서 멀티캐스트를 설정하고, 가입자망 내부에서도 멀티캐스트를 설정하였다. MPLS 구간에서는 멀티캐스트를 설정하고 PIM-SM으로 MPLS구간의 멀티캐스트를 구현한다.

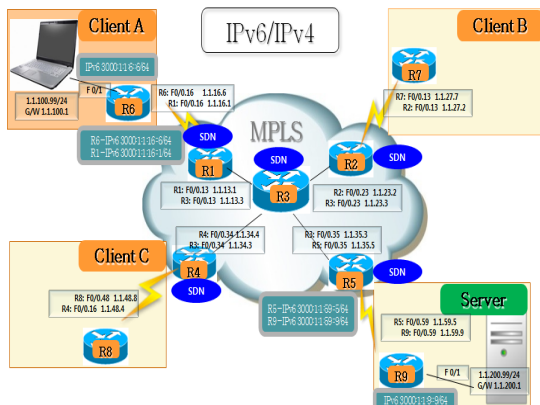


그림 6. 세부 구성도
Fig. 6 Advanced Configuration

MPLS 구간에 있는 라우터에 SDN 개념을 응용하여 MPLS EXP관련 Coding 프로그램을 적용한 라우터로 구현한다.

4.2. 최적화 분석

그림 7은 EXP값에 따른 R9에서 멀티캐스트 응답속도를 나타낸 것으로 EXP=5 일 때 라우터 9에서 멀티캐스트 평균 응답 속도는 239ms로 측정 되었다. 반면 EXP=0 일 때 라우터 9에서 멀티캐스트 평균 응답 속도는 289.4ms로 측정 되었으며, EXP 0과 5를 멀티캐스트 테스트 결과 응답의 일정한 안정성과 응답 속도의 차이가 있음을 확인할 수 있다.

그림 8과 9에서는 EXP 값이 5, 0 일 때 최소시간, 평균시간, 최대 시간을 나타내며 단독 전송일 경우와 동시 전송일 경우, 스트림 멀티미디어를 동시에 전송할 경우로 도식화 하였다.

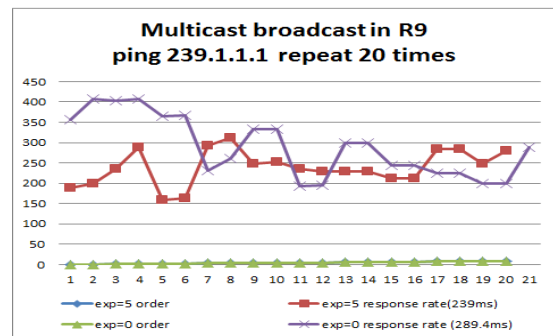


그림 7. EXP값에 따른 R9에서 멀티캐스트 응답속도
Fig. 7 Multicast response rate according to the EXP value in R9

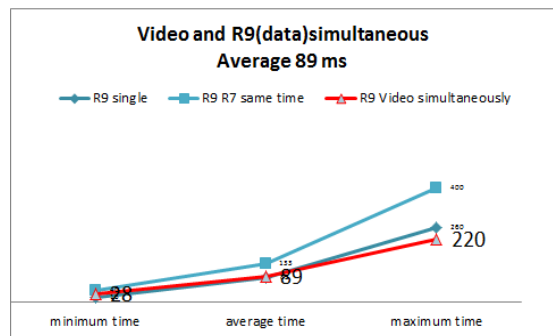


그림 8. EXP= 5
Fig. 8 EXP= 5

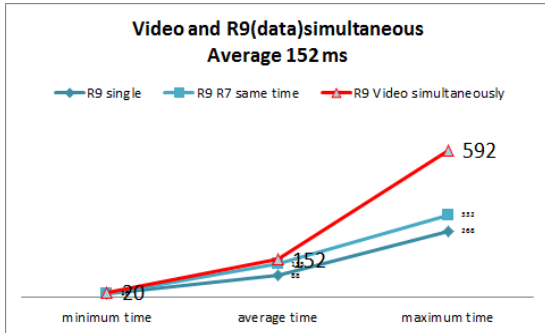


그림 9. EXP= 0
Fig. 9 EXP= 0

R9에서 단독으로 데이터를 전송할 경우 라우터의 자원에 영향을 미치지 않을 미비한 데이터 차이를 보이는 것을 확인하였다. 반면에 데이터를 동시 전송할 경우 데이터를 보낼 때부터 약간의 차이를 보이는 것을 확인할 수 있으며, 스트림 멀티미디어를 동시에 보낼 경우 평균시간은 EXP=5 일 때 89초와 EXP=0 일 때 152초의 차이가 확연히 나타났다. 실시간 데이터가 전송될 경우 EXP 값에 따른 차이는 없지만 스트림 멀티미디어가 전송되거나, 대역폭 혼잡이 발생할 경우 전송시간은 큰 차이가 발생한다. 이는 QoS가 적용되어 MPLS의 Label Switching 하여 우선순위로 전송 된다는 것을 확인할 수 있다

V. 결 론

멀티미디어 등의 실시간 멀티미디어 방송 전송과 고속 접속 서비스 증가에 따라 인터넷 트래픽이 급격하게 증가하고 있지만 인터넷 서비스 품질의 QoS가 보장되지 않고 있다. 더구나 앞으로 다가 올 IPv6 환경에서는 더 많은 모바일인터넷과 정보가전, 사물통신에 대한 수요로 자원효율화와 QoS 보장이 절실하다. 인터넷 서비스 품질의 QoS를 보장하기 위해 단순히 대역폭과 라우터의 증가와, 라우팅 테이블 수의 증가에 중점을 둔 망의 확장은 문제가 된다. 전통적인 IP Routing은 Layer 3 Routing 정보를 교환하며 Forwarding은 Destination Address만을 참조한다. 이로 인한 모든 라우터는 Full Internet Routing 정보 약 12만여개의 Route 정보를 필요로 한다.

따라서 본 논문에서는 IPv6/IPv4 Dual Stack 환경에서 Layer 2 Switching 기술인 패킷 forwarding과 Layer 3기술인 라우팅을 적용한 Label 패킷 forwarding 기법으로 하드웨어적 고속 스위칭이 가능해진 MPLS를 이용한 멀티캐스트와 융합하여 IPTV 방송 전송에 최적화를 고려하여 설계하였다. 그리고 라우터나 스위치에 새로운 소프트웨어나 프로그램을 실현하는 것이 가능 하는 SDN 개념을 응용하여 설계하였다. 본 논문에서는 QoS 부분인 EXP PHB Field 값의 MPLS IP Header 값에서 EXP 5와 EXP 0의 bit를 통하여 Utilization과 Throughput, Delay, Performance를 비교하여 QoS가 적용된 MPLS 중에서 멀티미디어 방송에 최적인 EXP= 5임을 제안하여 다음과 같은 결론을 도출하였다.

1. Delay, Utilization, Throughput : 일정시간이 경과 후 라우터의 자원이 어느 정도 사용되면 라우터가 처리할 수 있는 메모리나 큐 등의 사용으로 Performance가 떨어진다. 그러나 이와 상관없이 EXP= 5값으로 설정하여 QoS가 적용되면 4분 안에 라우터자원을 효율성 있게 확보하여 최적의 응답시간과 자원을 효율적으로 사용하여 전송되고 있어 지연과 사용율과 처리율 부분에서 QoS의 Performance를 상승시켰다.
2. Response Time : QoS의 가장 일반적인 항목인 응답시간을 최적화하기 위해 MPLS LER과 가입자 단말에서 FTP Upload의 평균 응답시간을 EXP =5로 설정하였을 때가 EXP=0 일 때 보다 빠르면서 안정적인 평균 응답시간을 산출하였다.
3. WFQ : WFQ Queuing Delay와 Buffer를 사용하여 인터페이스 처리율의 평균값이 EXP= 5로 설정하여 큐 처리와 버퍼 사용 율이 작아 자원 효율성을 해결하였다.

IPv6와 IPv4의 Dual Stack 환경에서 IPTV 전송 최적화 위한 SDN을 응용한 MPLS를 이용하여 다수의 사용자에게 실시간 스트림 전송을 멀티캐스트로 사용하여 대역폭과 자원의 효율화를 이루어 IPTV 방송 전송 최적화를 구현 하였다.

REFERENCES

- [1] Theophilus Benson et. al., "Network Traffic Characteristics of Data Centers in the Wild", *IMC* 10, Nov. 2010.
- [2] Draft Recommendation of Y.FNsdn-fm, "Requirement of Formal Specification and Verification Methods for SDN", *ITU-T*, 2012.
- [3] N. Handigol et al., "Where is the Debugger for My Software-defined Network?", *HotSDN*, Aug. 2012, pp. 55-60.
- [4] A. Voellmy and P. Hudak. "Nettle: Functional Reactive Programming of OpenFlow Networks", *PADL*, Jan. 2011.
- [5] X. Xiao, "Providing Quality of Service in the Internet", *Michigan State University*, Feb. 2000.
- [6] M. Canini et al., "A NICE Way to Test OpenFlow Applications", *NSDI*, Apr. 2012.
- [7] Glen Ragoonanan, "Infrastructure and SDN Transport Forum", *SDN & OPENFLOW World Congress 2013*.
- [8] Justin Dustzadeh, "End-to-End SDN: Carrier and Datacenter Networks," *SDN & OPENFLOW World Congress 2013*. 10.



황성규 (Seong-kyu Hwang)

2007 전주대학교 교육대학원 컴퓨터교육학과 석사
2012 ~ 2014.3 조선대학교정보통신과 공학박사
※ 관심분야 : 통신보안시스템설계, 네트워크 보안