



HAL
open science

Performance optimization of media distribution in overlay network using Open Flow

Shimin Sun, Li Han, Sungchol Cho, Sunyoung Han, Junfeng Wang, Béatrice
Paillassa

► **To cite this version:**

Shimin Sun, Li Han, Sungchol Cho, Sunyoung Han, Junfeng Wang, et al.. Performance optimization of media distribution in overlay network using Open Flow. 28th International Conference on Information Networking (ICOIN 2014), Feb 2014, Phuket, Thailand. pp.276–281, 10.1109/ICOIN.2014.6799481 . hal-03256417

HAL Id: hal-03256417

<https://hal.science/hal-03256417>

Submitted on 14 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Performance Optimization of Media Distribution in Overlay Networks Using OpenFlow

Shimin Sun, Li Han, Sungchol Cho, Sunyoung Han*
Dept. of Computer & Information Communication
Engineering
Konkuk University
Seoul, Korea
{shimin, hanli, syhan}@cclab.konkuk.ac.kr

Junfeng Wang
College of Computer Science
Sichuan University
Chengdu 610064, P.R. China
wangjf@scu.edu.cn

Béatrice Paillassa
Institut National Polytechnique
University of Toulouse
Toulouse, France
Beatrice.Paillassa@enseiht.fr

Abstract—Throughout the last decade, overlay multicast is a hot issue for media distribution services. High bandwidth consumption and multi-source multicasting in widely distributed networks is a critical issue for a wide range of applications including video conferencing, multi-party games, content distribution etc. A number of research projects have explored to use multicast as an efficient and scalable mechanism to deal with group oriented communications. However, there are still some limitations, such as long latency, duplicated packet transmission at certain points and lack of network equipment functionality support. With the appearance of Software Defined Networking, represented by OpenFlow devices, network control and management becomes possible to remote administrators. With the support of OpenFlow, it is possible to do some optimizations of existing media distribution mechanisms in overlay networks. In this paper, we proposed a novel method to improve the traditional overlay multicast relay networks with available limited number of OpenFlow devices, to cut down the bandwidth expenditures and perform a good control and management of media distribution services.

Keywords—*Overlay Networks; OpenFlow; Media Distribution; Multicast*

I. INTRODUCTION

With the progress of Internet infrastructure and media streaming technology, Internet video applications occupy more than 50% traffic in 2012 and will up to 70% before 2017 based on Cisco Visual Networking Index [1]. Those Internet video services, such as IPTV, P2P, Webcasting, Peercasting, are usually using one-to-many or many-to-many packet transmission pattern. Currently, the basic transmission method is usually based on server-client mode. This mode can easily cause bandwidth bottleneck on the server side. For example, if the server connects with 1000 clients at the same time, and provides 1Mbps streaming service for each client, then the total outgoing bitrate of the server is 1Gbps. It requires the downlink

to be more than 1Gbps. To alleviate server load and reduce bandwidth consumption in their networks, service providers usually deploy many servers in different locations.

There are other efficient ways to significantly reduce server load by adopting multicast technique. Multicast is used to deliver messages to a group of destination simultaneously in a single transmission from the source. The traditional multicast approaches contain IP multicast [2], overlay multicast [3] and application layer multicast [4].

IP multicast is the most common implemented multicast method, where routers create optimal global distribution trees and send to a multicast address, such as 239.1.1.1. IP multicast requires every joined routers are multicast enabled, and should perform group management, address allocation, and security, which bring unfavorable scalability issues. Today the scalability aspect of IP multicast is still an active research topic [5]. Meanwhile, the most current deployed routers around the world are not multicast enabled, which is the major reason why IP multicast is not widely used for media distribution system.

Overlay multicast and application layer multicast construct virtual networks for multicast groups. Unlike IP multicast, they replicate data packets at end hosts (or agents) [6], which is approach similar to P2P. A set of distributed Multicast Service Agent (MSA) are allocated in networks, which communicate to terminal hosts using multicast approach and connect with server using standard unicast approach with tunneling. Overlay networks effectively use the Internet as a lower level infrastructure, to provide higher level services to end users. Unfortunately, the tunnels between server and many MSAs still cause duplicated packets transmission in certain points especially at the outgoing link of the server, which will be described in detail in section II-A.

Around 2008, Software Defined Networking (SDN) [7] was proposed by clean slate research team of Stanford

* Corresponding author

This work was supported by National Research Foundation of Korea - Grant funded by the Korean Government (Ministry of Science, ICT and Future Planning), grant No. 2013K1A3A1A21000328.

This work was supported by the PHC program-STAR project, grant funded by Frenches ministry MAE and MER, grant No 299939XM.

University. SDN allows network administrators to manage network services through abstraction of lower level functionality. It divides the data transmission system with control plane and data plane. OpenFlow protocols [8] are used to communicate between control plane and data plane. Normally, in traditional routers or switches, the high level routing decisions and packet forwarding occur on the same device, as in Fig. 1. OpenFlow technique provides a much more flexible network management and control approach to networks. It enables networks to evolve by giving a remote controller the ability to manage the behavior of network devices. The forwarding decision is executed at a central software system, the OpenFlow controller, as in Fig. 2.

OpenFlow have been proposed to improve Internet routing, such as afford Quality of Service (QoS) guarantees to achieve higher-quality streaming, load balance. The flexibility of the definition of flow table in OpenFlow brings possibility for service administrators to control packets forwarding behavior and routings. For example, OpenFlow controller can set rules to specific packets, such as send to certain ports, rewrite IP headers, etc. For multicast streaming, OpenFlow devices can build tunnels among each other or rewrite header to change transmission pattern and path. These approaches could be easily used to solve the repetitive problem of media transmission. Many researches [14-16] have been proposed to provide a better media distribution service, but most of them are based on OpenFlow devices widely deployed. However, OpenFlow has just start its step and might take a long time to be available anywhere, like the hardship of IPv6 allocation.

This study introduce OpenFlow concept into overlay networks, and lend a hand to multicast services. Our system build on the existed overlay multicast architecture with the support of a limited number of OpenFlow devices. OpenFlow are used to set up connection among OpenFlow devices and perform multicast behavior in the last hop OpenFlow devices.

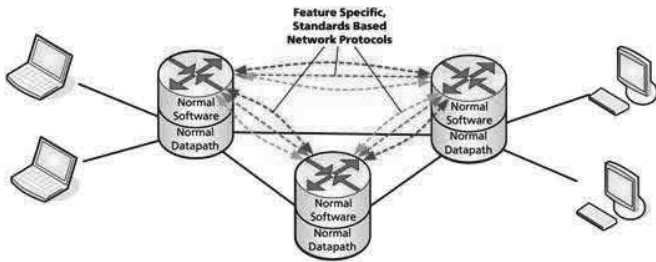


Fig. 1. Traditional networking scheme

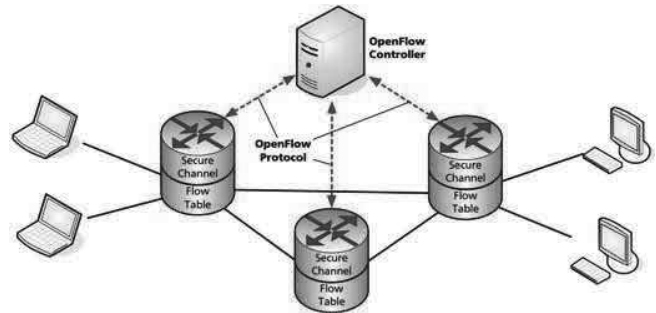


Fig. 2. OpenFlow networking scheme

This paper is organized as follows. In Section II, we introduce the overlay network, multicast technique and basic functions of OpenFlow. Section III details the design of the new proposed model for performance improvement based on OpenFlow technique. Then the implementation and evaluation are presented and discussed in Section IV. Finally, Section V concludes the paper by briefly summarizing the main points.

II. RELATED WORK

A. Overview of Overlay Network and Overlay Multicast

Overlay network is a layer of virtual or logical network that build on top of physical network, as in Fig. 3. Service providers run their application (IPTV) on their virtual networks, even though the real data packets are sent through the physical infrastructure.

Some protocols and services have been proposed to implement overlay multicast over the Internet. A recursive unicast scheme REUNITE [9] provides for network layer multicast, where unicast IP addresses are used for forwarding and delivery of packets. Some routers act as branching points of multicast tree, only branch routers perform multicast and other routers forward packets based on unicast routing. This approach needs service providers have the privilege to control some routers in the network. An OSPF protocol extension was proposed in [10] to achieve scalable multicast, which introduces new protocols for intra-domain multicasting mechanism. This approach still needs to maintain a multicast forwarding entry in some tree branch routers. Xcast (Explicit Multi-unicast) [11] is another approach that adds an additional header field contains all unicast IP of multicast group. Each router along the way parses the header, partitions the destinations based on each destination's next hop, and forwards a packet with an appropriate Xcast header to each of the next hops. This approach introduces a big workload to routers and need routers support new headers.

Different from above mentioned approaches which need the modification of routers or introduce new protocols, application layer overlay multicast usually deploys some agents in the network performs multicast relay function instead of original Media Server. Chainsaw [12] presents a p2p overlay multicast system, where peers are notified of new packets by their neighbors and need explicitly to send request to the neighbor in order to receive it. This approach can effectively eliminate distribution trees, but packets reach peers asynchronously and

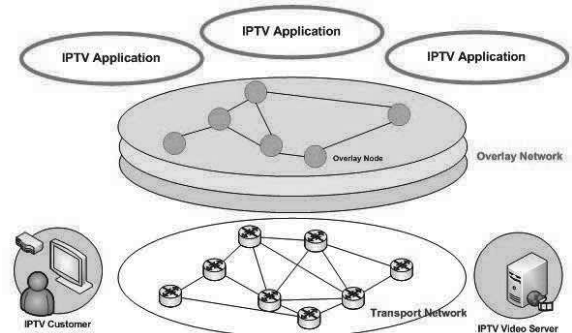


Fig. 3 Overlay network

management this kind of distribution system is difficult. A set of service nodes (MSNs) is used to implement media streaming applications in [13]. These MSNs are organized into an overlay and act as application layer multicast forwarding entities for a set of clients. This scheme brings beneficial to real-time applications since those MSNs receive packets simultaneously and perform relay at nearly same time.

Since IP multicast lacks of network devices' support, an overlay multicast is a better option for media distribution. Most of overlay multicast build unicast tunnels instead of multicast on the path from Media Server to agents (which perform multicast behavior for clients). Therefore, if one remote server has to provide the same service to multiple multicast groups which located in adjacent networks, it has to setup tunnels between server and each agent. For example, in Fig. 4, the Media Server has to setup 3 tunnels with each agent independently. Each tunnel transmits same packets and consumes same bandwidth in backbone networks. In this situation, repetitive problem exists in part of the path.

B. OpenFlow Use Cases for Multicast

SDN and OpenFlow technique brought new opportunities to overlay multicast approaches. Some researchers have been carried through. In [14], OpenFlow switches were used to achieve IP multicast and it provides calculation algorithm of different multicast trees spanning all switches of the network in case of fast recovery for path failure. Multicast ID assigned for each tree and switches need to rewrite packet header. In [15], a management method of IP multicast in overlay networks using OpenFlow was presented and implemented in VXLAN (Virtual eXtensible LAN). In [16], network layer agents were used for the services distribution and administrators have full control of the service and network routing. The existing studies usually need whole network OpenFlow configured with full control of the overlay network.

III. OVERLAY MULTICAST BASED ON THE SUPPORT OF LIMITED NUMBER OF OPENFLOW DEVICE

Currently, OpenFlow devices are still very rare deployed around the world and the price is much higher than legacy

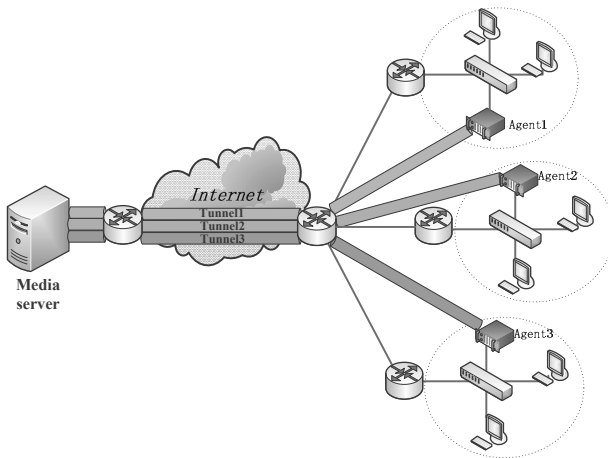


Fig. 4. Media distribution service with multi-agents

network devices. Even OpenFlow devices are universally deployed, the available OpenFlow devices that we can control may still handful. Our proposal bases on this status, and focuses on the optimization of media distribution performance with a limit number of OpenFlow devices' support.

In our consideration, at least two OpenFlow devices exist in the distribution system; one should locate at the outgoing link of Media Server and the other one exists in Internet which is close to several terminal multicast groups. As in Fig. 5, OF1 connects with Media Server and OF2 in the intermediate point of the network between Server and terminal networks. If no OpenFlow devices in terminal network, agents should be assigned to perform application layer overlay multicast. If an available OpenFlow device locates in terminal network, then OpenFlow can carry out the multicast behavior instead of agents. Comparing Fig. 5 with Fig.4, we can see that no repetitive tunnels exist between Media Server and OF2. This approach can reduce more than 1/3 of occupied traffic on the link between Media Server and OF2.

A. Virtualized View of Overlay Multicast Networks

The unique feature of OpenFlow can be used to achieve network traffic isolation [17]. By gathering specific network components, we can create virtual connection on top of physical network infrastructure. By mapping groups of flows into different virtual networks, we can create flow-based virtual network abstraction as in Fig. 6.

This virtualized network consists of Media Server, OpenFlow devices (OF1, OF2) and multicast agents (Agent1, Agent2, Agent3). These agents are the members of multicast group that managed by controller. It contributes to simplify the physical network topology, improve scalability and better management of the network.

B. Working Principle

We regard one virtual network as a multicast tree and a tree ID is maintained for each virtual network. We only focus on the media distribution in this virtual network. Media Server

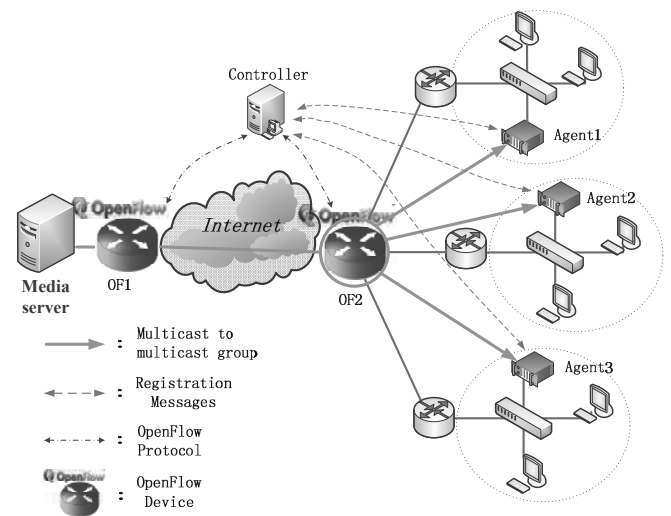


Fig. 5. Multicast relay with support of OpenFlow

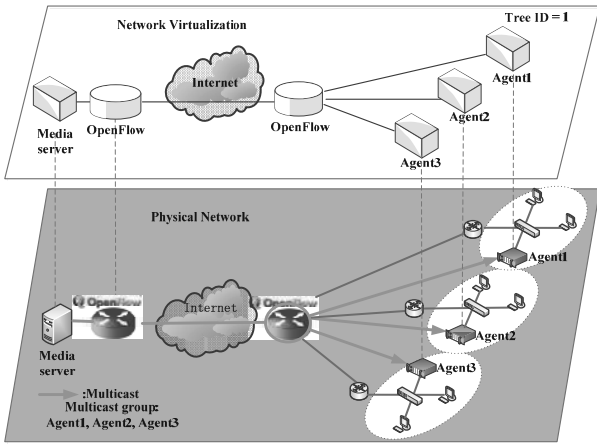


Fig. 6. Network virtualization using OpenFlow

sends streaming with specified port number, to OF1. The port number is the indication of streaming IDs in this system. Agents who want to acquire streaming from the Media Server should send registration messages to Controller. Controller verifies the requested agent, records the information of this agent in its tree table. Then it calculates the optimal transmission path and set rules to OF1 and OF2. OF1 and OF2 perform the forwarding of the required streaming based on its forwarding table that was set by controller. After that, the data transmission can be finished from Media Server to agents. Agents send the streaming to its multicast group in local network. The workflow of the system is as Fig. 7.

OpenFlow Controller is the core of the system. It should maintain a tree table for each agent. Such as in Table I, the tree information of three agents has been recorded. Different streaming may be transmitted to agents, which is marked by different port number. For different agents and streaming, Controller should set different rules to OFs.

1) *Set rule 1 to OF1:*

If OF1 receives packet meet: Source IP = Media Server's IP, Destination IP = OF1's IP and Source port number is specified corresponding with streaming IDs (such as 35111). Then, the packet will be rewritten to: Source IP = OF1's IP, Destination IP = OF2's IP and Source port number will not be changed.

2) *Set rule 2 to OF2:*

If OF2 receives packet meet: Source IP = OF1's IP, Destination IP = OF2's IP and Source port number = 35111. Then, the packet will be rewrote, replicated and sent to each agent in unicast mode: Source IP = Media Server's IP and Destination IP = Agent's IP.

To efficiently manage the virtual network and save bandwidth, status monitoring should be cared. If no client request for streaming service in one local network, the agent responsible for the streaming in this network should report to controller. Controller should reset the rule in OF1 and OF2 to stop sending streaming to this network. Meanwhile, beacon message is needed for agents to notify Controller its active status. We mark three status (active, stale and leave) in tree table of Controller.

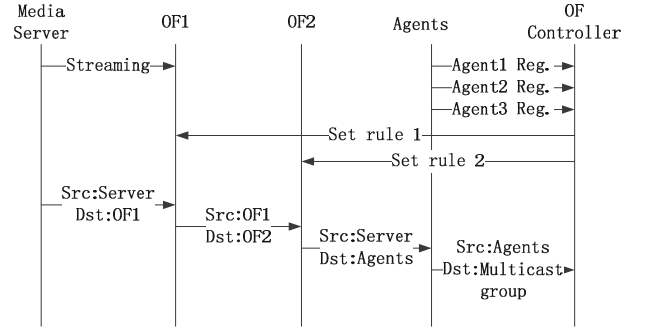


Fig. 7. Workflow of streaming transmission

TABLE I. TREE TABLE IN CONTROLLER

Agent	Streaming ID	Tree ID	OF1	OF2	status
1	1	1	Rule1	Rule2	active
2	1	1	Rule1	Rule2	stale
3	2	1	Rule3	Rule4	leave

IV. IMPLEMENTATION AND EVALUATION OF FUNCTIONALITY

In this section, we present a functionality implementation and simple performance evaluation based on the design explained in Section III.

A. Implementation

We implemented a virtual network that topology similar to Fig. 5, but only two agents configured. The system was implemented in the real international research networks. Media Server and OF1 were allocated in China connecting with CERNET [18]. OF2 connected with two agents in Korea using KOREN [19] and each agent connected with two PCs. In those machines, network protocol analyzer wireshark was installed to monitor flow changes. Each component of the system was deployed as follow and successfully achieved the functionality that proposed in Section III.

1) *Media Server*

VLC player version 0.8.6 was used, which is an open source software and provides multicast and streaming function with TCP/UDP/RTP/HTTP protocols support.

2) *OpenFlow Devices*

Open Vswitch 1.10.0 was installed in corresponding PCs which has two NIC card. The standard of OpenFlow switch specification is defined in [20].

3) *OpenFlow Controller*

Ovs-controller, which is a simple OpenFlow controller contained in Open Vswitch package, was used for setting the rules to Open Vswitch. The tree table in Controller that was defined and implemented manually.

4) *Multicast Agents*

VLC player version 0.8.6 was used as a streaming receiver and as a multicast streaming server for local network.

B. System testing

To compare the performance, we first built the network without the OpenFlow support. Media Server sent same streaming to two remote agents using VLC streaming function in TCP unicast transmission mode. Secondly, we built the network with OpenFlow existed. In this scheme, Media Server just sent one streaming to OF1 and the packets transmission following the flow as in Fig.7. By comparing the throughput of Media Server, we got the result that the OpenFlow enabled network saves nearly half of the throughput that no OpenFlow supported network.

C. Performance Evaluation

To simplify the evaluation process, some assumptions are defined.

- 1) Assume the topology from the Media Server to clients is a single tree.
- 2) Assume the topology between Media Server and clients is a simple hierarchical tree, where each router connects with two low level routers, as in Fig. 8.
- 3) Assume only two available OpenFlow devices exist in this network. One connects to the Media Server and the other one may be presented at any level and any point of this network.
- 4) Assume under each bottom router, one multicast agent is deployed.

We wish to compare the changes of bandwidth consumption between the network with OpenFlow devices' support and the network without OpenFlow devices. However, since OF2 deployed in the intermediate point of the transmission path, bandwidth consumption is not uniform for different part of the same path. Therefore, to quantify the optimization, we count every link between two routers as 1 link. We calculate the total number of links that same streaming go through the whole network.

For example, as in Fig. 8, if there are M agents receiving same streaming, then M flows will go through the top level to bottom level - level N . Then, the total link that the streaming go through is $M \times N$. Based on our assumption, one router extend to exactly two lower level routers, and for each bottom router there is one agent accompanied by. Then the total number of

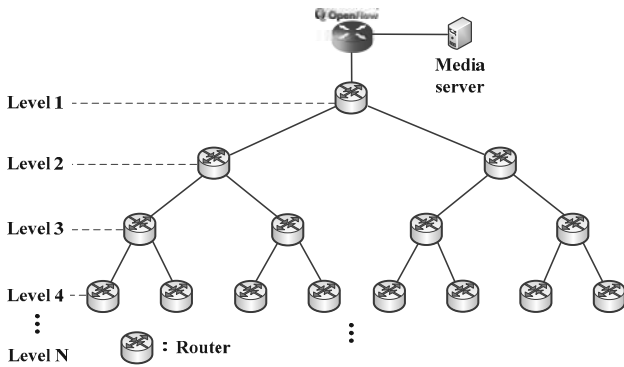


Fig. 8. Simple hierarchical tree topology

agent should be $2 \times (N-1)$ and the total links become $2(N-1) \times N$, which is the amount of link in no OpenFlow devices' network.

In OpenFlow supported network, if one OpenFlow device is located in level K , the number of links that above level K will be reduced based on the number of agents under the router. The number of links that reduced can be calculated by Eq. 1.

$$Num_{link_reduced} = (2^{(N-K)} - 1) \times K \quad (1 < K < N) \quad (1)$$

The reduction percentage between the number of reduced links and the number of total links can be calculated as Eq. 2.

$$Per_{link_reduced} = \frac{(2^{(N-K)} - 1) \times K}{2^{(N-1)} \times N} \quad (1 < K < N) \quad (2)$$

We evaluated the scenario with total tree level from level 2 to level 8, and OpenFlow device located in level 1 to level 5 respectively. The results are shown in Fig. 9 and Fig. 10. From Fig. 9, we can see that the number of reduced links increases exponentially along with the total tree level increasing. The location of OpenFlow device also impact the performance, higher level location is better than lower level location. However, more level means more routers, agents and more links. The percentage on the total number of links is decreasing along with the total tree level increase, as in Fig. 10. If the number of level $N \rightarrow \infty$, the result of percentage tends to zero, as in Eq.3:

$$\lim_{N \rightarrow \infty} \frac{(2^{(N-K)} - 1) \times K}{2^{(N-1)} \times N} = 0 \quad (1 < K < N) \quad (3)$$

The results indicate that higher level OF2 allocation can reduce more number of links, but along with the increasing of total tree levels, the influence of OpenFlow decreasing. In that situation, more OpenFlow devices should be deployed in the virtual network. Under efficient management and allocation of more OpenFlow devices, the performance can be improved more.

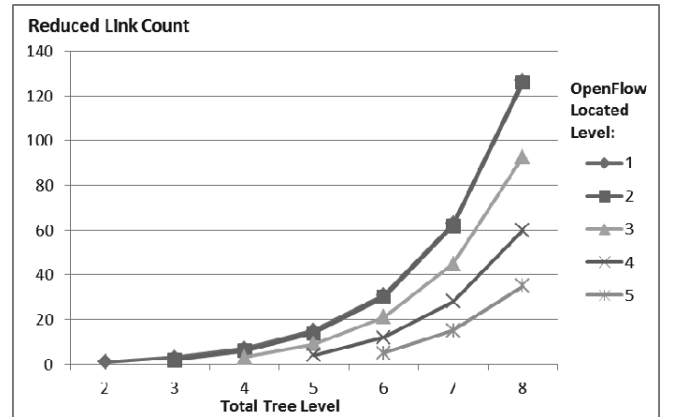


Fig. 9. The number of reduced links

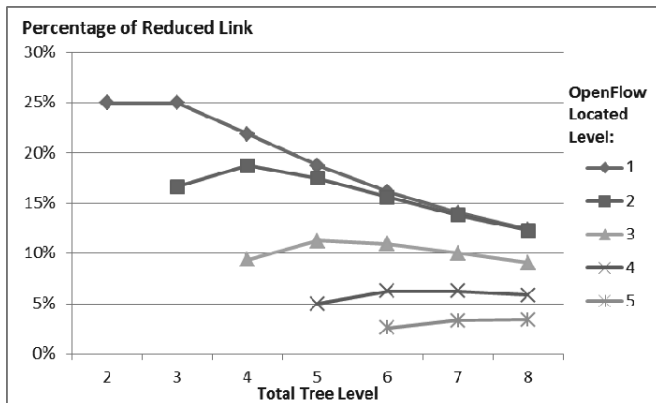


Fig.10. The percentage of reduced links

V. CONCLUSION

In this paper, we presented an optimization of media distribution system with the support of OpenFlow in overlay network. The OpenFlow Controller is responsible for the establishment of the virtual network and management of multicast trees. This approach can not only alleviate the repetition problem at certain point, but also provide the whole control of the transmission path and management of agents in terminal networks. For future work, multiple multicast trees will be introduced into the system. Security of streaming and authentication of agents and clients should also be considered. Multiple OpenFlow devices allocation will be evaluated to analyze the performance and look for the optimum location on the network.

REFERENCES

- [1] "Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," white paper, Cisco, May 2013.
- [2] Xing Jin, Wangqing Tu, Chan and S.-H.G., "Challenges and advances in using IP multicast for overlay data delivery," *IEEE Comm. Mag.*, vol. 47, issue 6, pp. 157-163, June 2009.
- [3] D. M. Moen, "Overview of Overlay Multicast Protocols," Available: <http://bacon.gmu.edu/XOM/pdfs/Multicast%20Overview.pdf>
- [4] S. Banerjee, Bobby Bhattacharjee and Christopher Kommareddy, "Scalable application layer multicast," In *SIGCOMM'02: Proc. 2002 conf. Applicat., technologies, architectures, and protocols for comput. Commun.*, NY, USA, 2000.
- [5] K. T. Phan, Joanna Moulhierac, Cuong Ngoc Tran and Nam Thoi, "Xcast6 treemap islands: revisiting multicast model," *Proc. of the ACM Int'l Conf. Emerging Networking Experiments and Technologies (CoNEXT)*, 2012.
- [6] Chih-Chao Wen, Chi-Ju Chiu, Cheng-Shong Wu, Hui-Kai Su and Yuan-Sun Chu, "An Integrated Two-Tier Multicast-Agent Architecture for All-IP Multicast Transport Services," *Int'l Conf. Broadband and Wireless Computing, Comm. and Applicat. (BWCCA)*, 2011.
- [7] "Software-Defined Networking: The New Norm for Networks," White paper, Open Networking Foundation, April 2012.
- [8] N. McKeown, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Comput. Comm. Review*, vol. 38, no. 2, pp. 69-74, Mar. 2008.
- [9] I. Stoica, T. S. E. Ng, and H. Zhang, "REUNITE: a recursive unicast approach to multicast," *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.
- [10] B. Zhang and H. T. Mouftah, "Providing multicast through recursive unicast," *IEEE Comm. Mag.*, vol. 43, no. 1, pp115-121, Jan. 2005.

- [11] Y. Imai, et al., "Explicit Multicast (Xcast) Concepts and Options," *IETF RFC5058*, Nov. 2007.
- [12] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," *Proc. IPTPS*, 2005.
- [13] Banerjee, S., Kommareddy, C.; Kar, K.; Bhattacharjee, B.; Khuller, S., "Construction of an efficient overlay multicast infrastructure for real-time applications," *Proc. Twenty-Second Annu. Joint Conf. the IEEE Comput. and Comm. (INFOCOM 2003)*, Mar. 2003.
- [14] D. Kotani, K. Suzuki, and H. Shimonishi, "A Design and Implementation of OpenFlow Controller Handling IP Multicast with Fast Tree Switching," *IEEE/IPSJ Int'l Symp. on Applicat. and the Internet (SAINT)*, 2012.
- [15] Yukihiko Nakagawa, Kazuki Hyoudou and Takeshi Shimizu, "A management method of IP multicast in overlay networks using openflow," *Proc. the first workshop on Hot topics in software defined networks*, pp.91-96, 2012.
- [16] Diplomarbeit, "Cross-layer Optimization of Peer-to-Peer Video Streaming in OpenFlow-based ISP Networks," M.S. thesis, Darmstadt University, Germany, April 2013.
- [17] Dmitry Drutskey, Eric Keller and Jennifer Rexford, "Scalable Network Virtualization in Software-Defined Networks," *Internet Computing*, vol. 17, no. 2, pp20-27, Mar. 2013.
- [18] "Korea Advanced Research Network," KOREN, Korea Communications Commission, Korea.
- [19] Ben Pfaff et al., "OpenFlow Switch Specification Version 1.3.1", Open Networking Foundation, Sep. 2012.
- [20] "China Education and Research Network," CERNET, Ministry of Education, China.