CS6250 Project Proposal
Video Streaming Over Software Defined Networks

Ashwin Baliga (GT ID: 903221131)
Manyu Deshpande (GT ID: 903207185)
Rohit Varkey Thankachan (GT ID: 903206736)
Sanyukta (GT ID: 903232482)
Vibha Satya Narayan (GT ID: 903227774)

29 September 2016

Abstract

This document presents a proposal to simulate a network that uses a SDN controller to find the best video streaming server and then uses IP multicasting to stream it to multiple subscribers as proposed in [Alizadeh Noghani and Oguz Sunay, 2015].

1 Introduction

In recent years, online video streaming has become an important part of many consumers' lives. It is predicted that IP video traffic will increase from 70% in 2015 to 82% of all consumer internet traffic by the year 2020 [CISCO, 2015]. Video streaming generally uses application level IP multicast to stream video to several subscribers due to several issues including all routers in the path having to support the IP multicast protocol [Diot et al., 2000]. Implementing multicast[Tang et al., 2014] at the application layer introduces several delays that could be avoided if it was done at the network layer.

Software Defined Networking (SDN) is a relatively modern technology that focuses on decoupling the control plane and data plane traffic. The separation of control plane provides flexibility to the service provider to use routing algorithms as per the application requirements. SDN also offers bandwidth on demand, which gives control on carrier links to request additional bandwidth when necessary, which is required for video streaming applications. Using a SDN controlled network, we could realistically implement IP multicast at the network layer.

2 Scope

We intend to implement the SDN-based framework described in [Alizadeh Noghani and Oguz Sunay, 2015] for video streaming using IP multicast. The proposed framework will be implemented over the Mininet virtual network simulator. We will use OpenFlow protocols to control switches on the network, which will be implemented using the Open Virtual Switch (OVS). The video streaming application and its subscribers will be implemented and benchmarked with an SDN controller in various routing settings. We will attempt to run benchmarks on this framework and compare it with a non SDN video streaming framework.

3 Plan of Action

3.1 Mininet Setup

We will use the Mininet [Mininet-Team, 2012] network simulator to run our simulations on virtual networks. Switches on these networks will be implemented using the OpenV Switch virtual switch[OVS-Team, 2010]. We will create 4 topologies of 15-20 switches and run our implementation on all 4 of these topologies. Mininet will allow us to package our entire code into a virtual machine that we could then distribute for other to reproduce our results.

3.2 The Video Streaming Server

The video that is streamed will be encoded using the H.264 format with 2 multiple descriptor codes(MDC). This allows us to use either of these descriptors to decode the video, and to increase quality by using both. We will use the open source VideoLAN x264 encoder to encode video and receive the 2 MDC codes (also known as multiple reference frames).

We will have several of these video streaming servers distributed in our network. These are known as descriptor providers (DPs) and will be connected to the SDN controller. When they are created, these DP's will communicate with the SDN controller with the list of descriptors that they serve. The SDN controller maintains a mapping of DPs serving a particular descriptor.

3.3 Subscribers

Nodes in the network streaming the video are known as subscribers. Subscribers are connected to the SDN controller which routes video from DPs to it. Subscribers send certain messages to join and leave multicast groups to the SDN controller. We will also use these subscribers to collect statistics for isolating performance metrics.

3.4 Protocols to select the best Video Streaming Server

[Alizadeh Noghani and Oguz Sunay, 2015] focus on being able to provide the best video streaming rate possible to subscribers. This is done by choosing the best video streaming server for each subscriber. We intend to implement and use the following routing protocols to test the framework:

- Minimum Hop - hops
- Shortest Path (Dijkstra) - sum of cost of each link
- MinMax - minimum of maximum costs of links between source and destination

3.5 SDN controller

The SDN controller will be implemented using the open source POX [NOXRepo, 2013] python library. This is a crucial component of the framework. It has several responsibilities:

- Manage "Join" and "Leave" requests from subscribers.
- Isolate the best DP to serve a subscriber using the routing algorithm used.
- Update data structures depending on the entry/exit of subscribers.
- Push the required new rules to the network using the OpenFlow protocol [McKeown et al., 2008].

- Query switches in the network periodically and update link weights.
- Keep track of descriptors actively being streamed and the servers that have the capability to provide them.
- Allocate multicast trees to subscribers based on Premium/Standards users.

3.6 Performance Analysis

We will collect QoE metrics from the subscribers to use as performance analysis. The metrics that we will be evaluating would broadly include the following:

- Packet Loss
- Latency
- Pre-Roll Delay

We will use a Application Level Multicast(ALM) protocol to stream video on a non SDN network to simulate the current method of streaming video and compare it with the results obtained using the SDN controller. The effect of the choice of the routing table will also be examined.

4 Division of Labor

This project can be divided into several chunks of work that can be completed in parallel. The responsibilities that are to be taken up by each team member is listed below.

Name	Responsibility Taken
Ashwin Baliga	Mininet Topology Setup and Switch configuration
Manyu Deshpande	Routing protocol implementation
Rohit Varkey Thankachan	Video streaming and encoder implementation
Sanyukta	SDN controller implementation
Vibha Satya Narayan	SDN

References

[Alizadeh Noghani and Oguz Sunay, 2015] Alizadeh Noghani, K. and Oguz Sunay, M. (2015). Streaming multicast video over software-defined networks. *arXiv preprint arXiv:1508.03592*.

[CISCO, 2015] CISCO (2015). Cisco visual networking index:forecast and methodology, 2015–2020 (white paper). Technical report.

[Diot et al., 2000] Diot, C., Levine, B. N., Lyles, B., Kassem, H., and Balensiefen, D. (2000). Deployment issues for the ip multicast service and architecture. *IEEE network*, 14(1):78–88.

[McKeown et al., 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

[Mininet-Team, 2012] Mininet-Team (2012). Mininet: An instant virtual network on your laptop (or other pc). Available at http://mininet.org/ (2016/09/29).

[NOXRepo, 2013] NOXRepo (2013).

[OVS-Team, 2010] OVS-Team (2010). Open vswitch, an open virtual switch.

[Tang et al., 2014] Tang, S., Hua, B., and Wang, D. (2014). Realizing video streaming multicast over sdn networks. *9th International Conference on Communications and Networking in China*.