Research on Video Traffic Control Technology Based on SDN

Ziyan Lin

Communication University of China, Beijing China

emmalinziyan@163.com

Keywords: video traffic control; software defined network; routing

Abstract. With mobile internet development, new video business appears continuously that bring forward higher demand for quality of services of the traditional network. Software defined network (SDN) can satisfy such demand by decoupling the control plane and data plane, and promote bandwidth availability, as well as implement flexible traffic control and provide independent bandwidth guarantee. In the paper, a video traffic control scheme is designed based on SDN, which distinguishes user priority and improve network availability by means of multi-path. Experiments show that the scheme can provide bandwidth guarantee, and promote the quality of video business.

Introduction

In order to guarantee the communication quality, the Quality of Service (QoS) technology has been developed in traditional network which is the overall performance of a network, and is measured by error rates, bit rate, throughput, transmission delay, etc. QoS is particularly important for the traffic with special requirements such as VoIP, IPTV and video applications. A number of attempts for layer 2 technologies that add QoS tags to the data have gained popularity in the past, such as frame relay, asynchronous transfer mode (ATM) and multiprotocol label switching (MPLS). Nevertheless, the traditional network architectures are virtually impossible to meet current market requirements because traditional network architectures were not designed to meet the requirements of today's users, enterprises, and carriers, which are hierarchical frame built with switches and routers in a tree structure. This kind of design made sense when client-server computing was dominant, but is ill-suited to the dynamic computing and storage needs of today's enterprise data centers, campuses, and carrier environments.

In order to solve such problems, the OpenFlow was presented by the researchers of Stanford University in 2008, and gradually developed into Software Defined Network (SDN) that allows network administrators to manage network services through abstraction of lower-level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) and forward traffic to the selected destination (the data plane), in this way, simplifies networking and promote the quality.

The paper is organized as following: study background and relevant technologies are introduced in section 2; the experiment scheme and design idea are described in section 3; the experimental results that prove the designing are illustrated in section 4; and the work summary and next step's work are given in section 5.

Study Background and Relevant Works

Integrated services. Integrated services or IntServ is an architecture that specifies the elements to guarantee quality of service (QoS) on networks. The idea of IntServ is that every router in the system implements IntServ, and every application has to make an individual reservation, which uses RSVP is the underlying mechanism to signal it across the network. IntServ works well in small-scale network, but for a system the size of the Internet, it is hard for all routers along the traffic path to support it and many states to be stored in each router. As a result, it is hard to keep track of all of the reservations.

Differentiated services. The idea of differentiated services or DiffServ is that it specifies a simple, scalable and coarse-grained mechanism to classify and manage network traffic on IP networks, and provide quality of service with low-latency for critical network traffic such as voice or streaming media.

In order for packet classification, DiffServ uses a 6-bit differentiated services code point (DSCP) in the 8-bit Differentiated services Field (DS field) in the IP header. In contrast to IntServ, DiffServ requires no advance setup, no reservation, and no time-consuming end-to-end negotiation for each flow, but how individual router deals with the DS field of the specific configuration is not stipulated, therefore it is difficult to predict end-to-end behavior. DiffServ provides QoS in each domain. It is complicated further if a packet crosses two or more DiffServ domains before reaching its destination. DiffServ or any other IP based QoS marking does not ensure quality of the service or a specified service-level agreement (SLA).

Multiprotocol Label Switching (MPLS). Multi-protocol Label Switching (MPLS) is a mechanism in high-performance telecommunication networks to direct data from one network node to the next based on short path labels rather than long network addresses, which avoids complex lookups in a routing table. MPLS encapsulates packets of various network protocols and supports various access technologies, and carries many different kinds of traffic, including IP packets, as well as native ATM, SONET, and Ethernet frames.

MPLS operates at between traditional definitions of layer 2 (data link layer) and layer 3 (network layer), and thus is often referred to as a "layer 2.5" protocol, and provides a unified data-carrying service for both circuit-based clients and packet-switching clients. At the same time, MPLS attempts to preserve the traffic engineering (TE) and out-of-band control for deploying large-scale networks. Due to the traffic management benefits of MPLS, such as better reliability, increased performance, in our experiments designing, the global quality of service is promoted by TE.

Nowadays, SDN has been used to implements QoS. Egilmez et al. [5] [6] proposed an OpenFlow protocol based controller, namely OpenQoS to achieve end-to-end QoS for multimedia-based applications. It is extended to a distributed architecture in [7]. OpenQoS classifies the traffic into data and multimedia flows. The multimedia flows are following the dynamic QoS guaranteed routing algorithm, whereas the other data flows are following the shortest routing path algorithm. Unlike another OpenFlow-based QoS framework in [4], OpenQoS does not use the queue mechanisms that uses OpenFlow to realize QoS guarantees for different service types. When the traffic of certain types exceeds the threshold, it can still cause link congestion.

In our designed experiments, the fine-control for video flow is realized based on user priority, and the queue mechanism of Openflow as well as Multi path forwarding technology.

Experiment Designing

The experiment designing is aimed to improve quality of service of network video, and ensure the video packets are sent to the user without lose and with lower delay, In addition, the experiment designing is also distinguished different users to ensure the bandwidth of high priority user, in order to provide business applications.

Designing based on DiffServ. For video application, the designed scheme can both distinguish different videos and different client (DiffClient) by IP address and port. In the simulation experiment, the SDN controller is responsible for quickly looking up the IP address and port of packets received in this way. The OpenFlow switch takes different forwarding strategy according to IP addresses of packets received under the controlling of the controller.

Independent bandwidth guarantee. It needs enough bandwidth for video business, especially live video business, to ensure the video received by user without delay and jitters, so it is crucial for video business to provide independent bandwidth guarantees. In this scheme, the queue mechanism of OpenFlow is employed to provide independent bandwidth guarantee for different priority user distinguished by the DiffServ. One OpenFlow switch provides QoS guarantee by the queue mechanism. Each port of the switch is built up one or several queues, which divides one linkage into multi-path. Each packet is forward to a corresponding queue. The transmission speed of a video flow is determined by the configure parameter of the queue.

In our experiment scheme, data packets with different priority are forward into different queues, the transmit bandwidth is determined by the configure parameters of the queue, which does not affect other users so that guarantees the higher priority users are not affected by lower priority users, the competition only happens in the users with same priority.

Multi-path. In our experiment scheme, the traffic engineering technology is introduced to solve the queue congestion and promote the reliability of data conveying. The traffic engineering (TE) is a kind of optimizing technology to distribute traffics into network. The key function of TE is route optimization that improves network performance by searching optimized routing. But for a large scale network with complicated structure and multi-path among servers, it is hard to improve network performance according to traditional routing protocol to choose one route between the destination and the source, moreover in this case it will waster network resources.

In order to solve such problem, our experiment scheme, all routes with same cost are gotten by the optimizing algorithm of the shortest path. In computing and finding module of our scheme, the controller computes all path costs and select minimized one and stores it in a HashMap. After a program running, the controller will get the statistical data (bandwidth occupied by a queue) from these paths periodically (including all queues of all ports), as soon as finds the traffic congestion, the equivalent routing is started up. In the mode of equivalent path, once there is the data packet arrived to the OpenFlow switch, it will be forwarded to the controller, then the controller will computes optimized path from multi-paths and dispatches routing table, and transmits the packets. By this way, it is ensured that packets will be forwarded to optimized path and avoid traffic congestion, at same time, promote the network performance and reliability, fully utilize the resources.

Scheme implementation. The route computing module is responsible of routing finding and path storing and carrying out multi-path algorithm. In our scheme, all resources are concentrated one path without occupying other paths when queue bandwidth is not reached the upper limitation. As soon as the controller running, it will measure the bandwidths occupied by users, and compute the bandwidths used by different priority user, then estimate if there is any user whose bandwidth is up to the limitation, if so, the multi-path routing will be used, packets will be forwarded to the queue of other path. All equivalent paths are got via route computing module, and the controller will compute the path according to bandwidths of queues, then start-up routing selection module that will schedule packet transmission.

The destination switch Dpid and port number can be got from Flowld in Floodlight controller [8]. In our scheme, routing selection module decides which user the path belongs to by checking the Flowld ‘s destination switch Dpid and port number. In the experiment, two equivalent paths are built labeled by 0 and 1 respectively. When the controller has determined user path, it will select path 0 for forwarding the packets in the case that queue bandwidth does not exceed rated bandwidth, which is determined by the value of the flag variable-vflag, else another path will be selected, vice versa. In this way, once one path is congestion, another path will be used, so that promotes availability of the network. The flow chart of forwarding packets is described in Fig. 1.

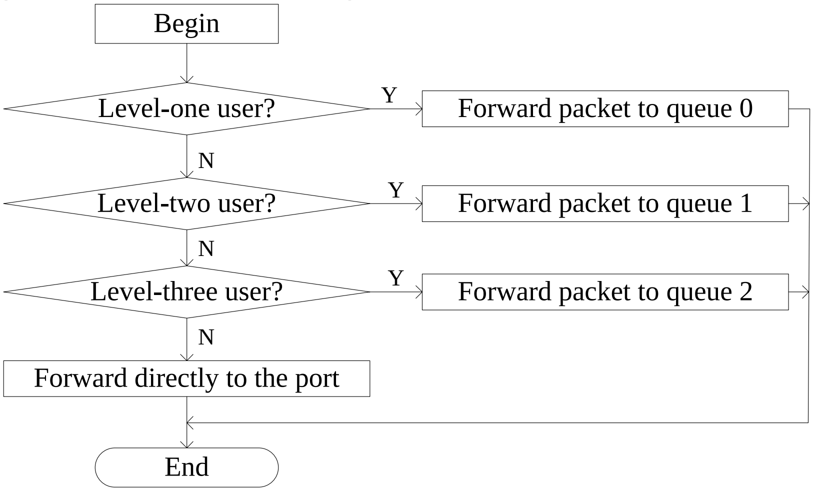
Begin
Level-one user?
Y
Forward packet to queue 0
N
Level-two user?
Y
Forward packet to queue 1
N
Level-three user?
Y
Forward packet to queue 2
N
Forward directly to the port
End

Figure 1. Flow chart of forwarding packets

Through routing finding, selecting, and fully using equivalent paths, the video traffic can be adjusted dynamically, and can fit burst traffic and guarantee the user’s quality of service. Generally speaking, the controller finds the whole network topology, and computes equivalent paths and determines the routing; routing selection module selects routing path, and implements the multi-path function; at last, the controller provides forwarding strategy and guarantees the user’s independent bandwidth.

Experiment design and result analysis

The experiment adopts simulation software for cost reason. The operation system is Linux, SDN network is built with Mininet [9], and the controller is widely used Floodlight . The experiment topology is illustrated in Fig. 2.

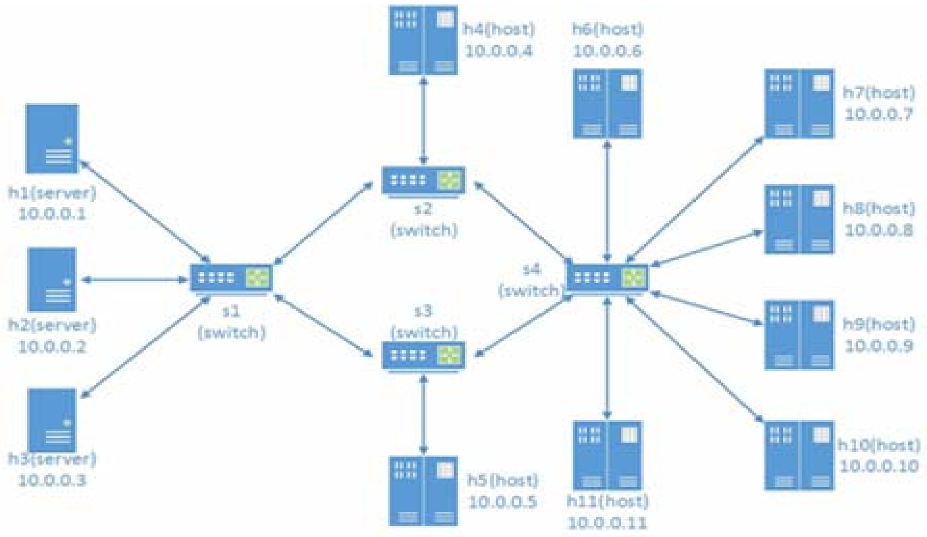

Figure 2. Experiment Network topology

In this experiment, 3 video servers named h1, h2 and h3 are used to provide same video contents in case one server is overload. 8 users are labeled as h4 to h11, among them h6 to h11 are the users of attending experiment, and h4 and h5 is the users simulating other network businesses. s1 to s4 are 4 switches that constitute two equivalent paths, i.e. s1-s2-s4 and s1-s3-s4. The bandwidth of each path is defined as 4.5Mbps. In the experiment, firstly set up virtual topology then start-up the controller program, and finally configure 3 queues at the ports of switch 1, switch 2 and switch 3, which upper bound bandwidths are 1.5Mbps. The main purpose of the experiment is to simulate video traffic controlling, therefore server h1, h2 and h3 are deployed with Darwin Streaming Server, and 6 users h6 to h11 are also configured with VLC to request streaming media.

There are total 3 experiments to simulate video services for traditional network, differentiated services and differentiated clients with different queue schedules and route methods and multi-path.

Experiment 1. This experiment simulates traditional network that doesn’t distinguish users and only one path from source to destination. In this experiment, one queue is built up at output port of the switch with bandwidth 4.5Mbps. Packets requested by user h6 to user h11 are forward through this queue, and in QoS module, there isn’t multi-path routing, therefore all packets are forward through path 0 (s1-s2-s4), regardless of bandwidth limitation 4.5Mbps.

The data rate of each experiment stream media is 1Mbps, once users exceed 4, the total data rate are larger than 4.5Mbps. In this case, packets are congested and frame drop happens that result in jitters of the video display shown in Fig. 3.

Experiment 1 indicates that traditional network can't cope with large users, if the video traffic exceeds the bandwidth of one path, all video images will be affected, which doesn't tolerated for video business.

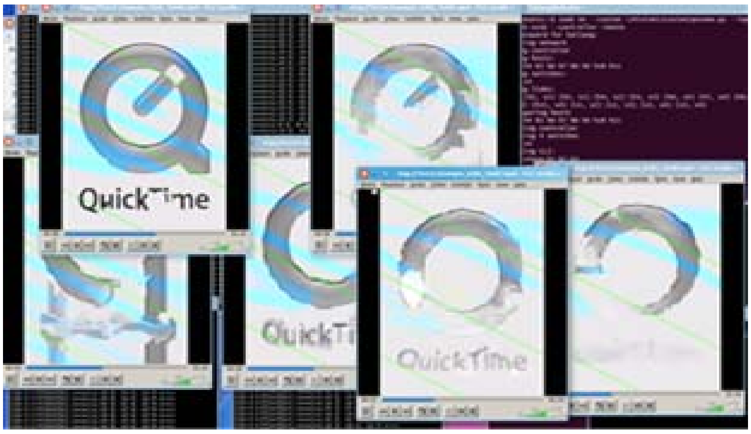

Figure 3. Results of Experiment 1

Experiment 2. Experiment 2 simulates one path transmission network with differentiated clients. The controller builds up 3 queues with bandwidth 1.5Mbps at each switch output port. The QoS module is forward packets of user h6 via queue 0, user h7 and h8 via queue 1, and user h9, h10 and h11 via queue 2. The routing selection is as same as experiment 1, all packets are forward through path 0 (s1-s2-s3) regardless of congestion.In experiment 2, the video of user h6 isn't jitter because of user-level one with bandwidth 1.5Mbps. The videos of user h7 and h8 as well as user h9,h10 and h11 are both jitter as soon as two or above two users request the video (data rate has above 1.5Mbps), because they are of user-level two and three respectively, this is shown in Fig. 4.

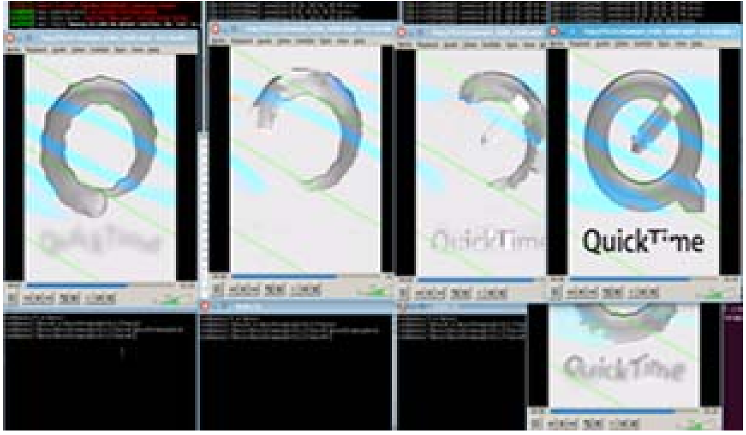

Figure 4. Results of Experiment 2

Experiment 2 indicates that Diffclient can obtain higher video quality for high level users. As for low level user, the video quality is poor for another equivalent path (s1-s3-s4) being laid aside.

Experiment 3. In experiment 3, Both Diffclient and multi-path technology are used. The controller measures periodically the quantity arrived packets at each user port, and computes the user bandwidth, as soon as data rate exceed rated bandwidth, starts-up multi-path routing, and forward packets through another equivalent path. In experiment 3, the strategy of putting packets into queue is the same as experiment 2. Therefore, video of user-level one is fluent without jitter. For user-level 2 h7 and h8, when both request video, the controller starts-up multi-path routing, the packets are forwarded through two equivalent paths, the videos of two users are fluent without jitter as soon as the system has been stable. As for user-level 3 h9, h10 and h11, so long as no more than two users request video, it is displaying well as user-level 2. If 3 users request video at same time, the video will be jitters for data rate exceeds the total bandwidth of two paths, which is shown in Fig. 5.

When 6 users request video at same time, the videos are fluent without jitter for level-one users and level-two users, but there are jitters for level-three users. The results of 3 experiments show that the scheme in this paper can implement user priority and guarantee the user bandwidth, as well as be fully used of equivalent paths, which promote network performance and broaden space of video business.

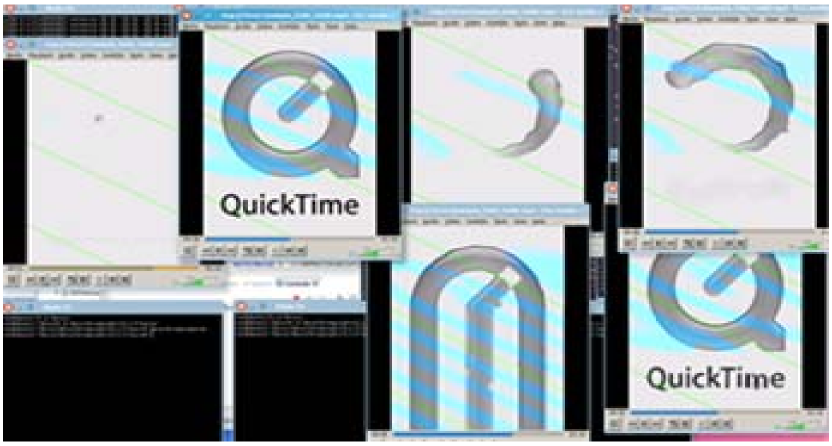

Figure 5. Results of Experiment 3

Conclusions

The fast development of video business need more robust QoS, although many approaches have been presented to solve this problem in traditional network, it is far from being solved due to present network frame. SDN presents a new idea and way. In this paper, we present a video traffic control scheme based on SDN, which can provide independent bandwidth guarantee for different priority users by means of the concentrated control and programmable feature of SDN. Besides that, our scheme can adjust packet forward path by measure the payload of the queue. The simulating experiments show that the scheme can promote the network performance, increase network throughput capacity and reduce video jitter, and is a better solving scheme on video traffic control in SDN.

References

[1] Thomas D. Nadeau and Ken Gray, Editor, “SDN: Software Defined Networks”, O'Reilly Media, Inc, USA (2013).

[2] N. McKeown, T. Anderson, H. Balakrishnan, et al. “OpenFlow: Enabling Innovation in Campus Networks”. ACM SIGCOMM Computer Communication Review, (2008) April 2; New York, USA.

[3] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, et al. “OverQoS: offering Internet QoS using overlays”, SIGCOMM Comput. Commun. Rev., Vol. 33, No. 1. (January 2003), pp. 11-16.

[4] W. Kim, P. Sharma, J. Lee, et al. “Automated and Scalable QoS Control for Network Convergence”. Proceedings of the 2010 internet network management conference on Research on enterprise networking, (2010) April 27; Berkeley, USA.

[5] H. E. Egilmez, S. T. Dane, K. T. Bagci, et al. “OpenQoS: An OpenFlow Controller Design for Multimedia Delivery with End-to-End Quality of Service over Software-Defined Networks”. In Signal & Information Processing Association Annual Summit and Conference, (2012) December 3-6; Hollywood, USA.

[6] H.E. Egilmez, S. Civanlar, A.M. Tekalp, “An Optimization Framework for QoS-Enabled Adaptive Video Streaming over OpenFlow Networks”, IEEE Trans. on Multimedia, 2013.

[7] Egilmez, H.E. "Distributed QoS Architectures for Multimedia Streaming over Software Defined Networks," Multimedia, IEEE Transactions on, vol.16, no.6, pp.1597, 1609, Oct. 2014.