



HAL
open science

SDN Based Architecture to Improve Video Streaming in Home Networks

Rabee Mustapha Abuteir, Anne Fladenmuller, Olivier Fourmaux

► **To cite this version:**

Rabee Mustapha Abuteir, Anne Fladenmuller, Olivier Fourmaux. SDN Based Architecture to Improve Video Streaming in Home Networks. The 30th IEEE International Conference on Advanced Information Networking and Applications (AINA-2016) , Mar 2016, Crans-Montana, Switzerland. 10.1109/AINA.2016.58 . hal-01365329

HAL Id: hal-01365329

<https://hal.sorbonne-universite.fr/hal-01365329>

Submitted on 13 Sep 2016

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.

SDN based architecture to improve video streaming in home networks

Rabee Mustapha Abuteir, Anne Fladenmuller and Olivier Fourmaux

Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris

Email: {mustapha.abuteir, anne.fladenmuller, olivier.fourmaux}@lip6.fr

Abstract—Nowadays, Internet video is the dominant internet traffic. DASH is an adaptive video streaming technique introduced to enable high quality video delivery over HTTP. In home networks, multiple video streams will compete for bandwidth, thus leading to poor performance and impacting the received quality of experience. In this paper we introduce a new technique to address this issue at the home network gateway without modifying neither the client player nor the video server. We design our framework NAVS (Network Assisted Video Streaming) relies on the deployment of Software Defined Networking (SDN). NAVS performs a dynamic traffic shaping based on the collected network traffic statistics and monitoring of video flows. NAVS dynamically allocates bandwidth for each video flow in real time. NAVS scheme has been evaluated over several metrics: bandwidth utilization, instability of players as well as the average video quality received by the clients. Our results demonstrate an improvement for all these parameters.

Index Terms—DASH, Home networks, SDN, Video streaming, Traffic shaping.

I. INTRODUCTION

Internet has been originally designed to transfer data but with the improvement of technologies, new services were introduced such as voice and video. Nowadays, video streaming is the dominant Internet traffic. Last report from Cisco Systems [1] shows the Internet video traffic represents 59% from the global internet traffic. Although, it will reach 77% by 2019. Over the top (OTT) video services such as Hulu, Netflix and Youtube became the most used services in the Internet. As an example currently Netflix [2] increases its domination of North America traffic to reach 36.5% of downstream traffic in the peak evening hours. In spite of the fact that there is no standard metrics for Quality of Experience (QoE), OTT spends a lot of investments to increase the users engagement with the video services. All these aspects led to develop new protocols for video streaming on the Internet such as DASH.

Dynamic Adaptive Streaming over HTTP (DASH) [3], an ISO standard also known as MPEG-DASH, was developed as a cooperation between industries and standardized organizations aiming to high quality video delivery. DASH run over HTTP due to the support of HTTP by the servers, middleboxes and clients applications. Each video is available on the server with multiple copies and each copy with different encoding. Each copy is divided into chunks with equal duration. The chunks meta data is available in the Media Presentation Description (MPD) which contains the video length, available bitrates and chunks URLs. First, the client requests the MPD file then it chooses the most suitable bitrate and starts downloading

chunks. The adaptation logic, to choose the next chunk, is kept open to the implementer. However, many implementations of DASH such as DASH-JS [4] and DASH.as [5] use bandwidth estimation based adaptation logic to select the next chunk bitrate. Further, when link status changes between the client and server then the client player requests either higher or lower bitrate chunks based on the last estimated bandwidth.

DASH runs over HTTP which uses TCP as transport layer, Thus leading to mismatch between the DASH adaptation logic which runs at the client side and the TCP congestion control which runs at server side. Further, the DASH player uses a buffer to minimize effect of packet jitter delays and smooths the playout. In the start of video streaming DASH starts filling the buffer and after a threshold it starts the playout. When the buffer fill ups an on-off pattern starts to appear. During ON periods the DASH player downloads new chunk while in the OFF periods it waits for the buffer drain. When two or more DASH clients compete for bandwidth [6] thus leads to instability of the players, unfairness between the players requested bitrates and network bandwidth under-utilization. This situation is common in shared access network such as home networks.

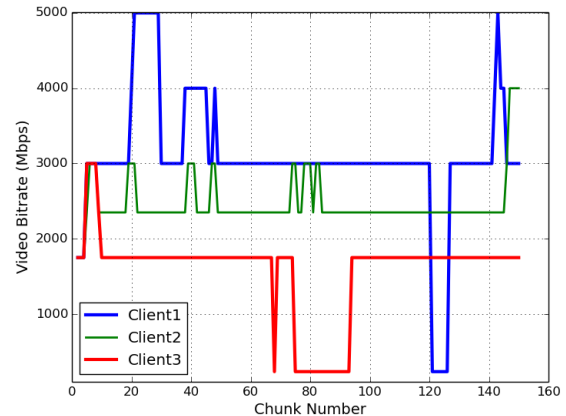


Fig. 1. Video bitrate for three DASH video flows run over home network.

Fig. 1 shows a performance of real DASH streaming for three clients over a home networks. The name of the DASH player is kept hidden to stress that the paper is not an evaluation of software. As it is clear from the figure the DASH player shows instability in the video bitrates, unfairness between

different clients and the total video bandwidth utilization for the three clients was less than the total available bandwidth.

Software-Defined Networking (SDN) [7] is a network architecture that aims to decouple the control plane from the data plane. The controller has global vision about the network and it makes decisions based on real-time view of the entire network rather than within limited visibility. With this separation [8] we could maintain, manage and optimize home networks in better way. Although, to improve the scalability and avoid the controller to become the bottleneck of the network we could use near-sighted [9] control plane architecture. Where two controllers were used, the first one called the global controller which could be in the cloud and the second one called near-sighted controller which could be within the home gateway and it has less functionalities compare to the global controller. In this paper we proposed a new framework NAVS (Network Assisted Video Streaming) which is built based on the SDN architecture. NAVS aims to improve the QoE by reducing player instability, maximize the fairness between client and increase the video quality. NAVS uses dynamic traffic shaping for the clients to achieve these goals.

The rest of the paper is organized as follows. Related work are presented in Section II. In Section III we explain our proposed framework NAVS. The performance evaluation is presented in section IV. Finally, the conclusion and future work is discussed in Section V.

II. RELATED WORK

HTTP Adaptive Streaming (HAS) is the most popular video delivery technique due to its compatibility with middleboxes. There are many HAS based solutions such as Adobe Systems HTTP Dynamic Streaming, Apple HTTP Live Streaming (HLS), Microsoft Smooth Streaming and DASH. These protocols share common behaviour over Internet networks leads to suboptimal performance. The authors in [10] summarised the main causes of suboptimal behaviour to three reasons: (1) bandwidth estimation depends on the chunks duration (2) disparity between the advertised chunk quality and real encoding (3) the dependence on HTTP/TCP to deliver the video streaming and the on-off pattern leads to inefficient bandwidth utilization.

HAS traffic shaping to improve users QoE in home networks was introduced in [11]. The authors applied traffic shaping in the home gateway based on the desired bitrate for each video flow. It is known when a client requests a bitrate R , it needs a bandwidth $(1 + \alpha)R$. This extra bandwidth is needed due to protocols overhead, queueing and retransmissions. The authors ignore the value of α in their calculation. Although, their technique is not dynamic, the traffic shaping is done in the beginning of session and its not changed when a new video flow starts streaming or video flow leaves the session. Akhshabi et al [12] proposed traffic shaping on the server side to reduce the player oscillations. When instability in the players is detected then the server reduces the player bitrate profile or increases it on the other case. However, this technique solves the instability problem by reducing the

bitrate profile and that leads to reduce the video quality. The main cause of player instability is the competition between video flows and the on-off pattern which should be solved at the home gateway. The authors in [13] compare two traffic shaping techniques, the Heretical Token Bucket (HTB) and the Receiver Tuning Methods (RTM). Further, the authors show that RTM gives better performance compare to HTB. Indeed, the authors shaped the traffic to the bitrate R_n for the video flow n rather than $(1 + \alpha_{R_n})R_n$ and that leads to bad performance for HTB. Notably, OTT video services [14] use TCP rate limiting in the server side to reduce TCP loss rate and the use of RTM could conflict with the server side control loop. Georgopoulos et al [15] proposed an SDN based solution to improve the QoE for video streaming in home networks. The client player delegates the adaptation logic to the controller and by this way they reduce the instability and maximize the fairness between the players. Indeed, changing the client player behaviour is a tough assumption and in this paper we keep both the client and server side without change. In [16] the authors evaluate the performance of traffic shaping for competing video flows over a shared bottleneck links using SDN. They show the individual traffic shaping gives better results than the aggregate traffic shaping. However, the authors shape the traffic for each client to constant value which is impractical because when the number of clients increases then the available bandwidth will be less than the required bandwidth, which leads to poor performance. In NAVS we will use dynamic bandwidth shaping so the traffic shaping for each client will depend on the available resources and number of video flows. Kleinrouweler et al [17] proposed to install HTTP proxy at the gateway. The proxy divides the available bandwidth between the video flows. The proxy rewrites the client requests to force it to reach the selected bitrates. In fact, the authors ignore the extra bandwidth α leads to system instability and as the client HTTP request rewritten then it effects the control loop of the player. In [18] the authors proposed SDN based solution with traffic prioritization to reduce video freezing. This solution need to change the client side to be aware of the video traffic prioritization. In comparison, NAVS keeps the client side without any change to prevent the video freezing. Mansy et al [19] proposed a Video Home Shaper (VHS) to improve video streaming for home networks. VHS allocate bandwidth for different client based on the encoding video rate and screen resolution. In NAVS we used SDN based architecture because we believe in near future we will have more devices in home networks and the best way to manage these devices is through SDN. This vision is consistent with trends of future home networks [20] [21] [22].

Hence, our proposed solution NAVS used SDN based architecture to improve video performance. NAVS keeps the client and the server without change. NAVS tries to improve the QoE and to reduce the instability. Further, improving video streaming in home networks using SDN is the first step to proof the concept of Software Defined Home Networks (SDHN). We will discuss the proposed architecture in more details in the next section.

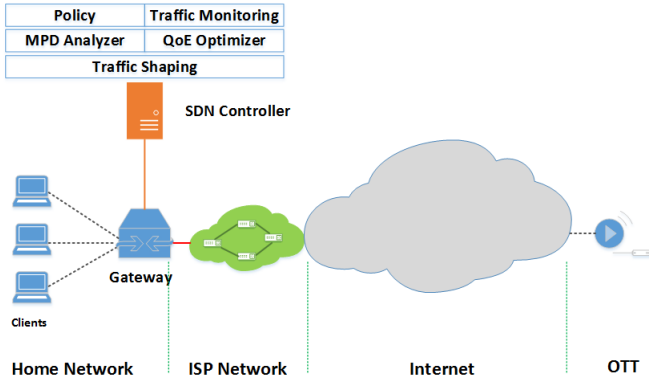


Fig. 2. NAVS architecture for home networks.

III. PROPOSED SOLUTION

In home networks, we have multiple clients connected to ISP network through the gateway. The link between the home network gateway and ISP is often the bottleneck and usually the link speed depends on the selected service price. The bottlenecks in home network [23] [24] could be due to limited bandwidth either in the access or in wireless link. In this paper we focus on solving the first problem. When multiple video flows compete for bandwidth, a poor performance could result as it was shown in Fig. 1. Our proposed framework NAVS (Network Assisted Video Streaming) is based on SDN architecture. NAVS uses dynamic traffic shaping to allocate bandwidth between different clients aiming to increase the QoE. In addition, NAVS used an abstract model for the controller so its location is less important than its function. Fig. 2 shows the NAVS architecture for home network. It presents high level view for NAVS component (controller architecture). The controller consists from five modules and next we will describe each one briefly.

A. Policy Module

Policy module is responsible of the policy that should be applied by the controller within the home network. This policy depends either on service level agreement or on user preferences. We could build this policy either by delegating to the end user the definition of the most appropriate policy using a Graphical User Interface (GUI) [25] or by using some predefined policies [17]. To simplify the problem we chose the second solution and we use a predefined policy. Indeed, fairness between video flows is one of the main objective of this research so the used policy was all video flows should get the same bandwidth.

B. Traffic Monitoring Module

This module is responsible of monitoring all the traffic within the home network and classification of flows. When a new client starts using the home network the first packet sent from the client is forwarded from the gateway to the controller and by this way the controller starts monitoring the new traffic flow. The controller could classify the traffic flows using (1)

DNS classification based on the destination IP address (2) Deep Packet Inspection (DPI) [26]. In this paper we choose the first technique because it is easier to implement and it has lower overhead compare to DPI. When the flow becomes inactive for certain time that means the flow has stopped and it will be removed from the active video flows table.

C. MPD Analyzer Module

When a flow is classified as a video flow by the traffic monitoring module the controller adds it to the list of active video flows. Meanwhile, the DASH protocol runs over HTTP which uses request and response messages. When a new client stream a video from DASH server it sends at the beginning a get messages to request the MDP file. This request is monitored by the controller. Then the server sends a response message which contains a copy from the MDP file. When this response reaches to the gateway a copy is forwarded to the controller. The controller passes the MPD file to the MPD analyzer to start the extraction of metadata and store it in the active video flow info table. The metadata contains the video length, the number of chunks, the available bitrates and the chunks URLs. When the video flow removed from active video flows table, the active video flow info table belonging to it is also removed.

D. QoE Optimizer Module

We have multiple clients stream video using DASH from video server. First the traffic module monitors all the traffic in the network and it classifies the video traffic. Then the MPD is sent as a copy to the controller. Based on the predefined policy the QoE optimizer tries to find the optimal allocation for the bandwidth between the clients. Meanwhile, the predefined policy presses the allocated bandwidth should be the same for all clients. We know if client n wants to stream video at bitrate R_n it needs bandwidth $(1 + \alpha_{R_n})R_n$ so we can formulate an optimization problem for N users as follow:

$$\arg \max \sum_{n=1}^N B_n \quad (1)$$

Subject to:

$$\sum_{n=1}^N B_n \leq BW \quad (2)$$

$$B_i = B_j \quad \forall \text{ two video flows } i \text{ and } j \quad (3)$$

Where: video flow n needs bandwidth $B_n = (1 + \alpha_{R_n})R_n$ to stream at bitrate R_n .

R_n is the bitrate selected by user n , α_{R_n} is the extra bandwidth needed by the flow that stream video with bitrate R_n (this extra bandwidth is needed due to the protocol overhead, queuing and retransmission), B_n is the bandwidth should be allocated to user n and BW is the total bottleneck bandwidth. The objective function of the QoE optimizer is to maximize the bitrate received by each client which represented by equation 1. This objective is subjected to (1) the bandwidth that assigned to all video flows is less than the overall

TABLE I
SIMULATION PARAMETERS

Simulation Parameters	Value
Bottleneck Bandwidth	30 Mbps
End to End RTT	60 ms
MTU	1500bytes
Video Bitrates	190, 260, 380, 750, 1350, 2750 and 5000 Kbps
Player Buffer Size	30 s
Chunk Duration	1 s
Simulation Time	100 s

bandwidth and is represented by equation 2 (2) Any two video flows should get the same bandwidth which is predefined by the policy and is represented by equation 3.

To solve this equation we need to know the value of α_{R_n} that should be assigned to bitrate R_n . In section VI we will explain how we evaluate the value of α_{R_n} but for now we suppose is preloaded at the controller. After solving these equations we could find the value of bandwidth B_n that should be assigned to each client n. This values will be sent to the traffic shaping modules to make sure it is applied at the home gateway.

E. Traffic Shaping Module

This module is informed of the bandwidth that should be allocated to each flow. Indeed, in this paper the value B_n is the same for all video because the policy is to assign the same bandwidth for all video clients but if the policy was changed the bandwidth could change from one video flow to another depending on the chosen policy. NAVS will do dynamic traffic shaping based on the number of video flows and the policy of the traffic shaping for each client will be defined. Further, if a new video flow starts or video flow is stopped the value of the allocated bandwidth for each client is recalculated. Traffic shaping is supported by most SDN switches such as OpenvSwitch [27] and by that NAVS could be implemented in any SDN based home gateway. We will describe how to calculate the value of α for each bitrate and the performance evaluation of NAVS in the next section.

IV. EVALUATION AND RESULTS

We divide this section to three subsections. In the first subsection we describes how to calculate the value of α . In the second subsection the performance metrics used to evaluate NAVS is presented. In the last subsection we evaluate the performance of NAVS.

A. Alpha Calculation

When a client streams a video at bitrate R_n the needed bandwidth is $(1 + \alpha_{R_n})R_n$ and this extra bandwidth is needed due to protocol overhead, queuing and retransmission. We use NS3 [28] network simulation tool to simulate the DASH video streaming over a network. NS3 doesn't include a module for DASH video streaming. However, we have implemented a new

module for NS3 to simulate the DASH video streaming. To make the experiment more realistic we used the same bitrates as used by Youtube [19]. Further, the client streams video with fix bitrate over the whole simulation time and from the average throughput we could calculate the value α .

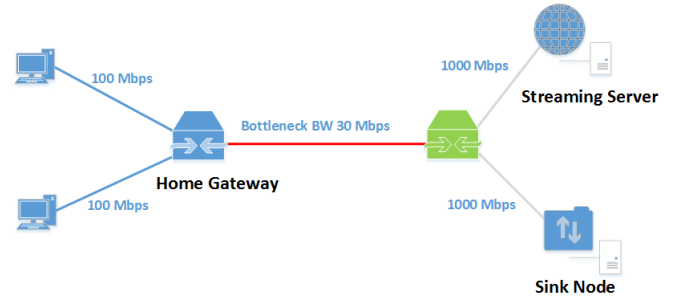


Fig. 3. The network topology used for the simulation.

Fig. 3 shows the network topology that was used in the simulation. We use dumbbell network topology to stress the bottleneck bandwidth which was 30 Mbps. Further, table I shows the simulation parameters used for the simulation. In this part we stream the video from streaming server without adaptation logic so in each run the video is streamed from the server using a fix bitrate. Then we evaluate the average throughput and we calculate the value of α_{R_n} that for a video stream with bitrate R_n .

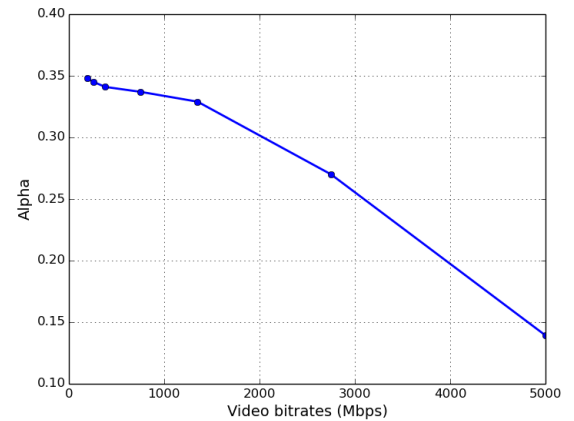


Fig. 4. Alpha vs the video bitrates.

Fig. 4 shows the value of α which is needed for each bitrate. When the video bitrate increases the value of α decreases and that due to (1) the protocol overhead is less compare to the overall streamed video size (2) when the bitrate increases the TCP protocol will converge to the TCP congestion avoidance period faster and that leads to less queuing and packets retransmission. Hence, the OTT video service provider could calculate the value of α with different scenarios and adds it to the MPD file then the controller could extract the value of α for each available video bitrate. Meanwhile, NAVS framework

could start the dynamic traffic shaping for each video stream based on traffic statistics, number of video flows and policy.

B. Performance Metrics

To evaluate NAVS we choose four metrics widely used in the literature to reflect the QoE when multiple DASH video flows share the same bottleneck link.

1) *Fairness*: based on Jain fairness [29] which is widely used in literature to evaluate the fairness between competing video streams [19] [30]. we define fairness between video streams as follow:

$$Fairness = \frac{1}{T} \sum_{t=1}^T \frac{(\sum_{n=1}^N R_{nt})^2}{N \sum_{n=1}^N (R_{nt})^2} \quad (4)$$

Where T is the streaming session length, N is the total number of video streams and R_{nt} is the video bitrate streamed by the video stream n at time t. When the fairness value is 0 that means the streamed bitrates are not fair at all. When the fairness value is 1 that means the streamed bitrates is totally fair.

2) *Instability*: QoE of DASH based video streams are extremely impacted by the bitrate switches [19][30]. We defined instability of the video streams as follow:

$$Instability = \sum_{n=1}^N \frac{S_n}{TC_n} \quad (5)$$

Where S_n is the number of switches of bitrate for video stream n and TC_n is the total number of chunks downloaded for video stream n. Smaller instability reflects better performance and vice versa.

3) *Average Video Quality*: The video quality increases as the bitrate increases [10]. Higher average video quality means better quality that leading to increase the user engagement with OTT video service. We defined the Average Video Quality (AVQ) for video streams as follow:

$$AVQ = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T R_{nt} \quad (6)$$

4) *Video Traffic Utilization*: Video Traffic Utilization (VTU) is a fraction between summation of bitrates used by video flows to the bottleneck bandwidth regardless of any background traffic [19][30].

$$VTU = \frac{1}{T} \sum_{t=1}^T \frac{\sum_{n=1}^N R_{tn}}{BW} \quad (7)$$

Where BW is the bottleneck bandwidth (link capacity). There is a relation between the average video quality and video traffic utilization. When the average video quality increases the video traffic utilization increases and vice versa.

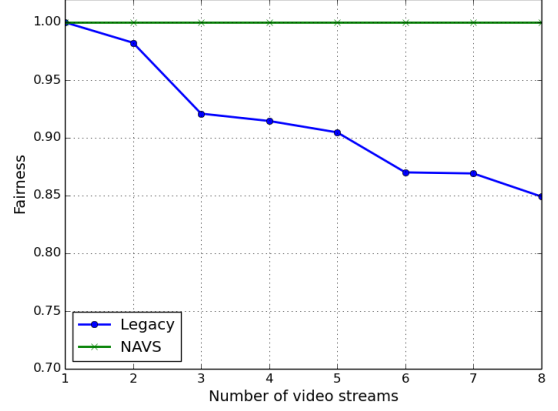


Fig. 5. Fairness between video streams with NAVS and without (labeled 'Legacy').

C. NAVS Evaluation

To evaluate NAVS we used the same topology as in Fig. 3 and we used the same simulation parameters of table I. To make the experiment more realistic we add background traffic so one client is downloading a file from the sink node. We compare NAVS architecture with legacy home network (We mean by legacy network a best effort network without NAVS framework). The adaptation logic in both cases was bandwidth estimation based algorithm. We run the simulation multiple times for different number of concurrent video streams ranging from 1 to 8. We analyze the trace files to evaluate different performance metrics (equation from 4 to 8). Fig. 5 shows the fairness vs the number of video streams. In legacy network when the number of video streams increases the fairness decreases. As a result some video flows stream in high bitrates and others in low bitrates due to wrong bandwidth estimation. The background traffic could lead to poor bandwidth estimation for the video flows. Contrarily, NAVS divides the available bandwidth equally between the video clients and makes like separated channel between the video flows so each client could stream the video without any interruption from other clients. Further, this separation reduces the bitrate switches for all clients due to the accurate bandwidth estimation and that leads to total fairness between the client. Hence, the ON-OFF pattern of DASH will not effect because of the separation between video flows resulting from dynamic traffic shaping. The extra bandwidth (after solving equation 1, 2 and 3) will be allocated to the background traffic.

Fig. 6 shows the instability vs the number of video streams. For the legacy network when the number of video streams increases the instability increases sharply and that due to bad bandwidth estimation resulting from the competition between video stream flows and the on-off pattern of DASH. In the other hand, NAVS shows more stability because the dynamic bandwidth shaping so each client could stream the video without any effect from other clients. The effect of background traffic was eliminated due to the traffic shaping. Reducing

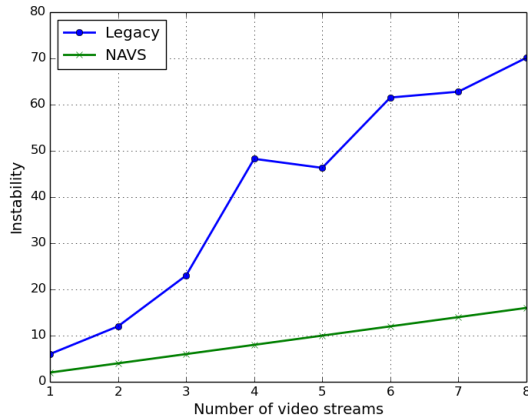


Fig. 6. Instability of video streams with NAVS and without (labeled 'Legacy')

the instability of video streaming will lead to better QoE and that will increase the client engagement with the OTT video service.

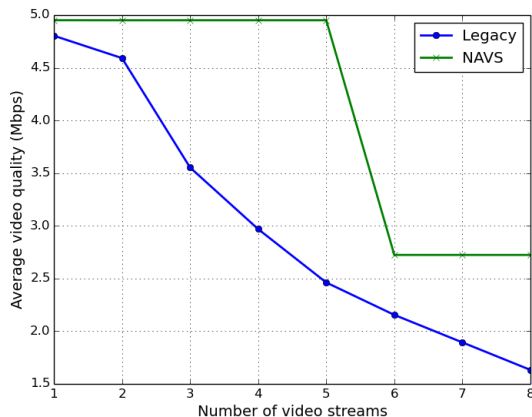


Fig. 7. Average video quality for video streams with NAVS and without (labeled 'Legacy') .

Fig. 7 shows the average video quality vs the number of video streams. For legacy network the average video quality decreases when the number of video streams increases and that due to unfairness and instability. However, the background traffic will compete with video streams and all of that leads to decrease the average video quality. In comparison, NAVS allocates the maximum possible bandwidth for each client to stream at the maximum possible bandwidth (equation from 1 to 3). When the number of video streams is less than or equal 5, NAVS allocates the maximum equal bandwidth for video flows but when the number of video flows increases more the five that will violate equation 2. Hence, NAVS reduces the allocated bandwidth for video streams and that explains the decrease of average video quality when the number video streams becomes greater than 5. Further, NAVS shows higher

average video quality compare to legacy network. The bottleneck bandwidth in our experiment was 30Mbps and when the number of video streams exceeded 5 it reduced the allocated bandwidth for each video stream because it violate equation 2. If we use higher bottleneck bandwidth then NAVS will make the average video quality higher until if the number of video streams exceed 5.

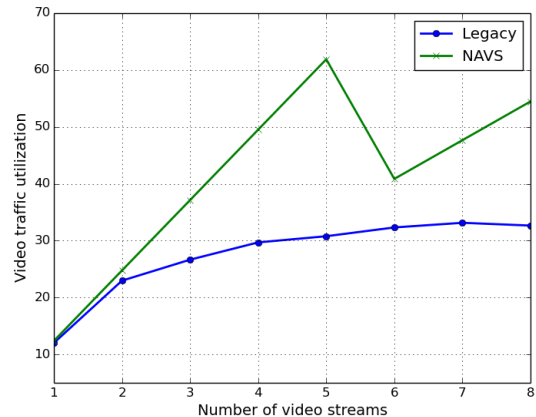


Fig. 8. Video traffic utilization for video streams with NAVS and without (labeled 'Legacy') .

Fig. 8 shows the average video quality vs the number of video streams. In legacy network when the number of video streams increases the video traffic utilization increases slowly until it reaches the saturation and that due to the decreases in average video quality and the traffic video utilization never reach 35%. Instead, NAVS shows better video traffic utilization until the number of video streams reach 5 then the video traffic utilization starts decreasing due to the lowering in average video quality. The zigzag pattern is due the bottleneck bandwidth and if we use higher bandwidth we will not see this zigzag pattern because NAVS will allocate higher bandwidth for each video stream. NAVS shows higher traffic utilization up to 60% which means video streams get higher average video quality.

V. CONCLUSION AND FUTURE WORK

We propose NAVS, a new SDN based framework to improve video streaming in home networks. NAVS allocated bandwidth for different video streams based on the policy, number of video flows, available video bitrates and bottleneck bandwidth. When a client streams a video with bitrate R_n it needs bandwidth $(1 + \alpha_{R_n})R_n$. We evaluated the values of α_{R_n} for Youtube bitrates R_n and we add them into the MPD. NAVS evaluates the most appropriate bandwidth for each client then makes dynamic traffic shaping.

Our work will continue in the following directions. Firstly, we used a simple policy which allocates the same bandwidth for all clients. We will address the problem of using more complex policies and implement them within NAVS. Secondly, we suppose the video quality increases when the bitrate

increases. In fact, other factors such as screen resolution, viewing distance and screen dimension could also effect. We will use more complex video quality function such as used in [19]. Thirdly, in this work we used bandwidth estimation based adaptation logic. We will evaluate the performance of NAVS with different adaptation strategies such as buffer based approach [31]. Fourthly, we will extend NAVS to support wireless home networks. In wireless links, the channel strength depends on the signal to noise ration and that could be another bottleneck in the network. We will address this wireless link bottleneck. Although, we will address the mobile video offloading from mobile operator to home network and vice versa. Finally, NAVS is the first step for Software Defined Home Networking (SDHN) and we will optimize it to be ready for implementation in commodity hardware.

REFERENCES

- [1] CISCO, "Visual networking index: Global ip traffic forecast 2014-2019," May,2015.
- [2] SANDVINE, "Global internet phenomena report 2015," May,2015.
- [3] V. Anthony and S. Iraj, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE Multimedia*, no. 4, pp. 62–67, 2011.
- [4] "http://www.castlabs.com," Sept,2015.
- [5] "https://github.com/dash-industry-forum/dash.js," Sept,2015.
- [6] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when http adaptive streaming players compete for bandwidth?" in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*. ACM, 2012, pp. 9–14.
- [7] O. N. Fundation, "Software-defined networking: The new norm for networks;" *ONF White Paper*, April,2012.
- [8] H. Kim and N. Feamster, "Improving network management with software defined networking," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 114–119, 2013.
- [9] J. Schulz-Zander, N. Sarrar, and S. Schmid, "Towards a scalable and near-sighted control plane architecture for wifi sdns," in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 217–218.
- [10] S. Bae, D. Jang, and K. Park, "Why is http adaptive streaming so hard?" in *Proceedings of the 6th Asia-Pacific Workshop on Systems*. ACM, 2015.
- [11] R. Houdaille and S. Gouache, "Shaping http adaptive streams for a better user experience," in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 1–9.
- [12] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2013, pp. 19–24.
- [13] C. B. Ameer, E. Mory, and B. Cousin, "Evaluation of gateway-based shaping methods for http adaptive streaming."
- [14] M. Ghobadi, Y. Cheng, A. Jain, and M. Mathis, "Trickle: rate limiting youtube video streaming," in *Proceedings of the 2012 USENIX conference on Annual Technical Conference*. USENIX Association, 2012, pp. 17–17.
- [15] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide qoe fairness using openflow-assisted adaptive video streaming," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013, pp. 15–20.
- [16] J. J. Quinlan, A. H. Zahran, K. Ramakrishnan, and C. J. Sreenan, "Delivery of adaptive bit rate video: balancing fairness, efficiency and quality," in *Local and Metropolitan Area Networks (LANMAN), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.
- [17] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar, "Modeling stability and bitrate of network-assisted http adaptive streaming players," in *Teletraffic Congress (ITC 27), 2015 27th International*. IEEE, 2015, pp. 177–184.
- [18] S. Petrangeli, T. Wauters, R. Huyssegems, T. Bostoën, and F. De Turck, "Network-based dynamic prioritization of http adaptive streams to avoid video freezes," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1242–1248.
- [19] A. Mansy, M. Fayed, and M. Ammar, "Network-layer fairness for adaptive video streams."
- [20] "http://goo.gl/eytaep," Sept,2015.
- [21] "https://goo.gl/kkbnvp," Sept,2015.
- [22] "https://goo.gl/nxhh5o," Sept,2015.
- [23] S. Sundaresan, S. Burnett, N. Feamster, and W. De Donato, "Bismark: a testbed for deploying measurements and applications in broadband access networks," in *2014 USENIX Conference on USENIX Annual Technical Conference (USENIX ATC 14)*, 2014, pp. 383–394.
- [24] S. Sundaresan, N. Feamster, and R. Teixeira, "Locating throughput bottlenecks in home networks," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 351–352.
- [25] H. Kumar, H. H. Gharakheili, and V. Sivaraman, "User control of quality of experience in home networks using sdn," in *Advanced Networks and Telecommunications Systems (ANTS), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.
- [26] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song, "Flowqos: Qos for the rest of us," in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 207–208.
- [27] "http://goo.gl/lniek3," Sept,2015.
- [28] "https://www.nsnam.org," Sept,2015.
- [29] R. K. Jain, D.-M. W. Chiù, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," 1984.
- [30] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 389–400.
- [31] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 187–198.