# A Markov Model for Evaluating Resource Sharing Policies for DASH Assisting Network Elements

Jan Willem Kleinrouweler*, Sergio Cabrero*, Rob van der Mei*[†], Pablo Cesar*[‡]

*Centrum Wiskunde & Informatica      [†]VU University      [‡]Delft University of Technology

Amsterdam, The Netherlands      Amsterdam, The Netherlands      Delft, The Netherlands

Email: {j.w.m.kleinrouweler, cabrero, r.d.van.der.mei, p.s.cesar}@cwi.nl

*Abstract*—In this paper, we present a model for evaluating bandwidth sharing policies, that can be applied to networks that handle both video streaming traffic, as well as other traffic. Video streaming is a demanding network application. In crowded networks, resources need to be properly divided in order not to diminish the streaming experience. However, in network deployments with a large number of users, the streaming performance cannot be obtained straightforwardly from a sharing policy. Therefore, we propose a Markov model that is compatible with Dynamic Adaptive Streaming over HTTP (DASH), the major technology for video streaming over the Internet. If DASH is combined with in-network resource management, its performance can be significantly improved. Nevertheless, resource sharing policies need to be configured. This requires evaluation of many different configurations. Real deployments or network simulations demand many system resources and time. In contrast, our model can quickly evaluate many configurations, and for each configuration output the expected video bitrate and number of changes in video bitrate. These two parameters play an important role in the Quality of Experience of the viewer. In this paper, we demonstrate how our model can be used to analyze and optimize resource sharing policies. As such, our model is a useful tool for network administrators and allows them to better provision and configure their networks.

## I. INTRODUCTION

Video streaming over the Internet is a popular network application. However, due to the demand that video streaming puts on a network, its performance and resulting quality of experience (QoE) is largely dependent on how the network is handled. In managed networks, bandwidth sharing policies determine how much bandwidth is available for video streaming, and how bandwidth is available for other traffic. In this paper, we investigate how sharing decisions for these policies affect the streaming performance. We present a performance model that can be used to determine the streaming performance, that is compatible with Dynamic Adaptive Streaming over HTTP (DASH) and network-assisted DASH.

DASH (sometimes referred to as HTTP adaptive streaming) is the major technology for delivering video over the Internet. It has been adopted by major content providers, such as YouTube and Netflix. The client-pull based technology relies on HTTP for transport, and it is known to be firewall friendly and it allows for scalable distribution using content delivery networks (CDNs). Although this technology has advantages over traditional UDP based streaming, it has the drawback that the streaming performance is highly sensitive to other traffic in the network. DASH players have difficulties to maintain a stable high quality stream when a network connection is shared with other DASH players, or when there is (non-video) background traffic. This results in the Quality of Experience (QoE) of the viewer being lowered, measured in terms of video bitrate and number of changes in video quality during playback [1][2].

DASH streaming performance can be significantly improved by using DASH assisting network elements (DANEs). DANEs are in-network elements with knowledge about the network, the current load, and specifically the number of active DASH players. Based on these factors, a DANE partitions the available bandwidth in resources dedicated to DASH streaming and resources for the remaining traffic. Furthermore, it will divide the bandwidth among DASH players, when there are multiple DASH players active at the same time. The DANE will enforce these sharing rules by means of traffic shaping and signaling target bitrates to the players.

In small networking environments (e.g. a home- or small business network) DANEs can provide highly personal network management. Sharing rules are typically ad-hoc and require network administrators to have specific knowledge about the users, their devices, and their applications. Nevertheless, the effects of the sharing rules on the resulting streaming performance are clear. This is different when a large number of users shares a network infrastructure. It is not straightforward how sharing policies propagate considering dynamics in the number of video streams and variations in background traffic.

To determine the streaming performance given a policy, the network environment that is controlled by the DANE has to be evaluated. Real deployments or network simulations with a large number of nodes are unpractical (i.e. they are time consuming or require powerful hardware) and are sometimes even unfeasible. However, to find the optimal policy it is required to quickly evaluate many different configurations. This cannot be offered by actual deployments or network simulations.

The contribution in this paper is twofold. First, we propose a Markov model that can be used to evaluate the streaming performance given a sharing policy. Our model distinguishes DASH flows from other traffic, and allows to specify how bandwidth is shared between these two types of traffic. The result of the model is the mean streaming bitrate and how often this bitrate changes. For these two factors it has been identified that they largely contribute to the Quality of Experience of the viewer [1]. Secondly, we apply our model to demonstrate how decisions on bandwidth sharing affect the video quality, and

show how our model can be used to obtain an optimal resource sharing policy.

It is impossible to provide a definitive answer on how to optimize video quality, since it largely depends on the local configuration. Changing parameters like capacity, demand on the network, and optimization goal may lead to different outcomes. Therefore, we intent our model to be used as an analytical tool, targeted to network administrators that would like to better manage video traffic in their networks. Based on analyses with the model that we presented in this paper, network administrators can optimize resource management while being aware of the resulting video streaming performance. Compared to real deployments and network simulators, applying our model requires less system resources and time. As such, it allows for evaluation of many different configurations prior to deployment, leading to better configured DANEs.

The remaining of this paper is structured as follows. In Section II we discuss related work and provide a background on video streaming using DANEs. Section III characterizes the different traffic flows and identify different sharing policies. In Section IV we formulate the Markov model that can be used to evaluate sharing policies. In Section V we demonstrate how our model can be used to analyze and optimize streaming performance. Section VI concludes this paper.

## II. BACKGROUND

To determine the video streaming performance in a network, given a sharing policy, it is required to perform many evaluations. Because real deployments and simulations are often expensive, we proposed a model that can be used to perform such evaluations [3]. To the best of our knowledge this model is the only model specifically on evaluating DASH streams in networking environments with DANEs, though it shows similarities to the use case from Elayoubi and Roberts to evaluate in-network video transcoding and caching services [4]. However, their evaluation is primarily targeted at cache performance, and does not take changes in video bitrate as an output parameter.

Our previous model allows to specify different groups (e.g. different encoding schemes, devices, or type of user) and accurately outputs mean video bitrate and bitrate changes [3]. However, applicability of this model is limited to deployments with only DASH traffic. As such, it only focusses on resource sharing between DASH players, and not on the impact of background traffic on video quality. In many networks a video streaming service would be deployed in the presence of other traffic in the network. In this paper we aim to close this gap and evaluate how bandwidth sharing between DASH streams and background traffic affects streaming performance. Furthermore, we apply our model to three different sharing policies and analyze the impact on streaming performance.

The goal of a resource sharing policy is to increase the streaming performance of DASH players, and thus to improve the quality of experience of the viewer. The four main factors that contribute to the QoE are: stalling, initial delay, quality and resolution, and bitrate switches (i.e. changes in video quality) [1]. Due to the nature of adaptive streaming, stalling can largely be avoided and the initial delay can be kept short, by changing to low-bitrate video when needed. Furthermore, if DANEs provide Quality of Service for DASH streams, then these events are even more unlikely. The video bitrate and bitrate changes depend on how the DANE handles the current demand, and variations in demand, on the network. This is how the resource sharing policy in the DANE manifests itself and influences the QoE of the viewer. Therefore, in this paper we will use video bitrate and bitrate changes as metrics to measure streaming performance.

To optimize the streaming performance the video bitrate must be maximized. This means that it is important to spend time on higher layer representations when possible [2][5]. Although changing to a higher bitrate representation is often appreciated, if it is followed up by a switch back to a lower bitrate representation shortly after, this effect diminishes. Often switching between different representations, known as bitrate instability, has been identified to negatively impact the QoE [6][7][8]. In the remaining of this section we will give a background on related technologies, for the convenience of the reader.

Dynamic adaptive streaming over HTTP (sometimes referred to as HTTP adaptive streaming) is the dominant technology for video streaming over the Internet. In DASH, the a video file is split up into short segments, typically with a duration between two and ten seconds. Each segment is encoded at multiple bitrates and resolutions. A manifest file describes the representation and the order of the segments. DASH protocols are designed to be client-pull based with HTTP as the transport protocol. This means that the segment files, together with the manifest, are put on an HTTP server. A DASH player first downloads the manifest to obtain full knowledge of the stream. Based on an adaptation algorithm, the player then downloads segments in representations it sees fit. Adaption algorithms can include factors like current network bandwidth, buffer level, device type, screen resolution, and battery level.

In most DASH architectures the adaptation algorithm is built into the player. This has the major advantage that server infrastructures are stateless (i.e. normal HTTP), and thus allow for more scalable distribution compared to server-push technologies. However, it has been shown that current DASH-capable solutions have difficulties selecting a bitrate, and that they suffer instability and unfair resource sharing when there is background traffic, or when multiple DASH players share a bottleneck link [9][10][11]. Instability and unfairness are the result of a mismatch between the bursty ON/OFF download behavior of DASH players and the TCP transport protocol [12].

A plethora of adaptation algorithms have been proposed in the literature. However, solving problems with bandwidth sharing in the players remains difficult due to a lack of knowledge on network usage. Network devices potentially have this knowledge, and in combination with their capabilities of doing in-network traffic control, they can be leveraged to form a better adaptation mechanism. Network devices that have a minimum knowledge about DASH traffic, and assist DASH players to improve their streaming performance, are referred to as DASH assisting network elements [13]. Several implementations have been presented, including traffic shaping at the residential gateway [14], DASH-aware proxy

servers [15][16][3], and Software Defined Networking (SDN) based implementations [17][18]. DANE implementations aim to eliminate the problems introduced by the client-pull mechanism, and provide a bridge between the traditional server-push based streaming and DASH technology. In the Server and Network Assisted DASH (SAND) architecture the communication between clients and DANEs, and between different DANEs in a network, is being standardized [13].

## III. System description

We first shortly discuss the networking environments that we think are interested to apply our models to. Secondly, we characterize the different types of traffic and identify different sharing policies.

### A. Use case

Resource management is typically deployed in networks where bandwidth is scarce. This can occur in cases where the available bandwidth is little, or when a network is shared among a large number of users. In both such cases, an overseeing entity can provide a better division of bandwidth between flows in order to optimize the networking experience. In small deployments, different flows can be addressed separately and DANEs can provide highly personalized traffic management. In this work we are interested in the scenario where a large number of users shares a network connection, and a DANE is deployed to optimize the QoE for a video streaming service. A popular example of such a scenario is a public Wi-Fi network offered in a social event. The event organization, or venue, could enrich the experience by offering video feeds from other parts of the event or with additional content.

### B. Traffic characteristics

In our analysis we are primarily focussed on video streaming performance. Therefore, we distinguish between DASH video traffic and other traffic. We characterize the two types of traffic as follows:

Type 1 flows are for DASH streams that belong to the video streaming service we are trying to optimize. For each individual stream the DANE knows in which bitrates the stream is available. Based on the current demand and the selected sharing policy, the DANE selects one of those available bitrates and reserves a matching amount of bandwidth. The target bitrate is communicated to the DASH player, that will use this bitrate in the requests for the next segments. DASH is client-pull based, and the player is responsible for selecting the bitrate. It selects the bitrate in the HTTP requests for a next video segment. This poses the restriction that the video bitrate can only be changed every $x$ seconds, where $x$ is the duration of a segment. As a result, if a DASH player receives multiple target bitrates in between two segment requests, only the last communicated bitrate will be effective.

Type 2 flows are for the remaining flows that do not belong to the DASH streaming service. Since our primary focus is on video quality, we will refer to Type 2 flows as background traffic. How much bandwidth is reserved depends on the demand on the network. Type 2 flows represent the aggregate of all background traffic, for which the throughput is obtained by monitoring the network in the DANE. Because it takes time to propagate the new bandwidth reservations, and install the new queueing configuration for traffic control, changes in background traffic are processed as averages over longer intervals (e.g. a few seconds). This will also prevent the DANE from responding to aggressively to short peaks in demand. Additionally, shifting to a different bandwidth reservation level occurs in discrete steps. Smaller steps in combination with more frequent updates, will result in a better match to the actual demand. However, such decisions have an effect on quality and stability of DASH stream. In this paper we will, among other factors, evaluate what is the impact of reservation step size and change interval.

### C. Sharing policies

Depending on the network where a DANE is deployed in, a different type of sharing policy can be adopted. In this paper, we specify three types of policies that will be discussed below in detail: DASH priority, background traffic priority, and mixed mode. For each type of policy, we define the admittance of DASH players to the network and the priority of DASH traffic over background traffic. The available bandwidth is divided into three zones: reserved for DASH, reserved for background traffic, and a shared zone. The two reserved zones are put in place to prevent starvation of either type of traffic. These zones should be configured to represent the absolute minimum bandwidth that is made available to DASH and background traffic. The shared zone is divided between DASH players and background traffic depending on the selected policy. We only consider policies where once a DASH player is admitted to the network, it will receive enough resources to finish the stream. Stalling, or forcefully stopping a stream by the DANE, as result of a too high demand on the network has such a strong impact on QoE, that these cases should be avoided.

The different types of policies define how DASH traffic interacts with background traffic. They are defined as follows:

**DASH priority:** In this policy DASH traffic is treated as most important. DASH streaming can take as much of the bandwidth in the shared zone as it requires. This decision has two major effects. First, it means that DASH players are allowed in the network as long as they fit in the reserved zone for DASH plus the shared zone. This means that the bandwidth that is required for all players to stream at the lowest available bitrate cannot exceed the bandwidth that is available in the DASH zone plus that is available in the shared zone. The second effect is on the video bitrate for the active players. In DASH priority mode, A DANE selects the highest bitrate that is possible, and can take up all the bandwidth in the shared zone while doing this. Only when there is bandwidth left, the DANE allocates more resources for background traffic. This policy will be most effective in scenarios where DASH traffic is the primary type of traffic, and network access for other services is only provided for convenience.

**Background priority:** This policy is the opposite of DASH priority, and should be considered in environments where a video streaming service should interfere as little as possible with the "original" traffic. As an implication of the rule that the DASH streams should not be stopped by the DANE once

allowed into the network, the number of DASH flows is limited to how many DASH flows would fit in the zone reserved for DASH. Allowing more DASH flows would risk the possibility that the background demand cannot be satisfied. The DANE only selects higher bitrate than the lowest when the demand from background traffic allows it (i.e. when the demand is low).

**Mixed:** In this mode, the policy for admitting DASH flows into network is taken from DASH priority mode, and combined with the constraint that the bitrate for DASH flows is determined by the background demand. This means that DASH flows are admitted to the network as long as they fit the shared zone, but can only get to higher video bitrates when the background demand is low enough. However, to provide a bit more flexibility, a target bitrate could be specified. If a target bitrate is specified, the highest bitrate that does not exceeds this target bitrate, and would fit in the shared bandwidth zone, will be selected. Higher bitrates can only be selected when the demand for background traffic would allow this. Setting the target bitrate to the highest available bitrate would yield the DASH priority policy. However, setting it to the lowest bitrate does not equal the background priority policy, as the maximum number of DASH flows that is allowed into the network could be higher in mixed mode.

## IV. PERFORMANCE MODEL

In this section we present the model that can be used to determine the streaming performance. For the purpose of traceability we adopt a Markov model.

### A. Markov model

We define a two dimensional Markov process to describe the population of the two traffic types. The Markov process is defined as the vector $(n_d, n_{bg})$, where $n_d$ describes the number of DASH flows, and $n_{bg}$ represents the current demand from background traffic. The state space of the Markov process is denoted by $\mathcal{S}$. We use the notation $n_{d|bg}(\vec{x})$ to refer to the number of DASH flows, or the intensity of background demand, represented by a state $\vec{x}$.

In the horizontal dimension, we model the number of DASH flows that are active in the network. The state space is horizontally bounded by the maximum number of DASH flows that can admitted in the network, without affecting the continuity of the stream. Let $C$ be the capacity of the channel, $Z_{dash}$ the bandwidth zone that is guaranteed to be reserved for DASH flows, $Z_{bg}$ be the bandwidth zone reserved only for background traffic, and $B_{min}$ the lowest available bitrate for DASH videos. The maximum number of DASH flows then becomes $(C - Z_{bg})/B_{min}$ for DASH priority- and mixed mode. For background priority mode, this maximum is $Z_{dash}/B_{min}$.

The vertical dimension represents the demand of background traffic on the network. As defined in Section III-C, the demand changes with discrete sized steps. A transition from $n_{bg}$ to $n_{bg} + 1$ represents increasing the demand of background traffic by one step. In our analysis we will make a distinction between background demand, and how much bandwidth is actually reserved for background traffic. The Markov process describes the background demand, and is

vertically bounded by the maximum number of background demand steps: $C/B_{bg}$, where $B_{bg}$ is the size of a step. Note that the background demand could overlap with $Z_{bg}$, to allow for the channel to be fully utilized when there are no (or a low number of) DASH players.

The vector $(n_d, n_{bg})$ is a Markov process with the following transition rates:

- $n_d$ increases to $n_d + 1$ at rate $\lambda_d$,
- $n_d$ decreases to $n_d - 1$ at rate $n_d/\beta_d$,
- $n_{bg}$ increases to $n_{bg} + 1$ at rate $\lambda_{bg}$,
- $n_{bg}$ decreases to $n_{bg} - 1$ at rate $n_{bg}/\beta_{bg}$.

The above listed transition rates for states $\vec{x} \in \mathcal{S}$ define transition matrix $Q$. In this paper we make the simplifying assumption that both DASH flows and background demand are described by an Erlang process. This allows us to analytically obtain the steady state probabilities $\pi(n_d, n_{bg})$ by using the Erlang multi-rate loss formula, as follows:

$$\pi(n_d, n_{bg}) = \frac{1}{G} \cdot \frac{(\lambda_d \beta_d)^{n_d}}{n_d!} \cdot \frac{(\lambda_{bg} \beta_{bg})^{n_{bg}}}{n_{bg}!}, \qquad (1)$$

where

$$G = \sum_{\vec{x} \in \mathcal{S}} \frac{(\lambda_d \beta_d)^{n_d(\vec{x})}}{n_d(\vec{x})!} \cdot \frac{(\lambda_{bg} \beta_{bg})^{n_{bg}(\vec{x})}}{n_{bg}(\vec{x})!}. \qquad (2)$$

For DASH flows we have shown in previous work that this would be appropriate [3]. However, different networking environments could yield different background traffic patterns, that possibly require a Markov process with different transition rates for transitions $n_{bg} \pm 1$. Nevertheless, it would still be possible to numerically solve $\pi Q = 0$ in short time, as the size of $\mathcal{S}$ is not too large. Furthermore, the methods to obtain the streaming performance that we will describe below, still apply.

### B. Streaming performance

The bitrate of the DASH streams, and how often this bitrate changes, depends strongly on how a DANE selects bitrates for the streams. Depending on the number of DASH flows and the background demand, a DANE will select the highest possible bitrate from the set of available bitrate, that satisfies the constraints put there by the sharing policy. We determine the bitrate for each state in $\mathcal{S}$, and determine the expected bitrate by averaging over the states based on the steady state probabilities.

If the expected number of players is defined as:

$$\mathbb{E}[N_d] = \sum_{\vec{x} \in \mathcal{S}} \pi(\vec{x}) n_d(\vec{x}), \qquad (3)$$

then expected bitrate of the DASH players becomes:

$$\mathbb{E}[B_d] = \sum_{\vec{x} \in \mathcal{S}} \pi(\vec{x}) n_d(\vec{x}) q_d(\vec{x}), \qquad (4)$$

where $q_d(\vec{x})$ is the bitrate that is selected by the DANE for DASH players in state $\vec{x}$. In practice, the encoding scheme for DASH videos typically have bitrate steps that are not of the same size. They are chosen such that every next step in

bitrate will give a similar improvement in quality. Equation 4 can straightforwardly be modified to express mean quality level, by letting $q_d(\vec{x})$ define quality level instead of bitrate.

The number of bitrate switches closely relates to how often the Markov process transitions between states that have a different bitrate selected for DASH players. Changing the bitrate of a DASH stream is technically limited be the size of DASH segments. Let $t_{seg}$ be the size of the segments in seconds, then DASH streams can only change bitrate every $t_{seg}$ seconds. We include this behavior in our analysis by only observing the Markov process every $t_{seg}$ seconds. Via uniformization of the Markov chain we can obtain the probabilities that the process transitions from state $\vec{x}$ to state $\vec{y}$ in $t_{seg}$ seconds, $P_{\vec{x},\vec{y}}$. The expected number of bitrate switches per minute then becomes:

$$\mathbb{E}[Q_d] = \frac{60}{t_{seg}\mathbb{E}[N_d]} \sum_{\vec{x},\vec{y} \in \mathcal{S}} \pi(\vec{x}) P_{\vec{x},\vec{y}} s(\vec{x} \rightarrow \vec{y}), \quad (5)$$

where $s(\vec{x} \rightarrow \vec{y})$ is the number of DASH stream that make a bitrate switch when transitioning from $\vec{x}$ to $\vec{y}$. DASH streams only switch bitrate when the bitrate in $\vec{x}$ is different from the selected bitrate in $\vec{y}$. Furthermore, only streams that are active in both states have to make a switch. The number of streams that make a switch on the transition $\vec{x} \rightarrow \vec{y}$ then becomes:

$$s(\vec{x} \rightarrow \vec{y}) = \min(n_d(\vec{x}), n_d(\vec{y})) \cdot \min(1, |q_d(\vec{x}) - q_d(\vec{y})|). \quad (6)$$

### C. Background demand

In the Markov process, we model the demand of background traffic on the network. However, in some conditions the actual bandwidth that is reserved for background traffic cannot match the demand, because the sharing policy prescribes a higher priority for DASH traffic. In those cases, the actually reserved bandwidth, denoted as steps by $b_{bg}$, will be lower than $n_{bg}$. Similar to the selected bitrates for DASH, $b_{bg}$ has to be computed for each state in $\mathcal{S}$. The bandwidth that is assigned to background traffic for state $\vec{x}$, given $B_{bg}$ as the background demand step size, is defined as:

$$b_{bg}(\vec{x}) = \frac{C - n_d(\vec{x})q_d(\vec{x})}{B_{bg}}. \quad (7)$$

This means that background traffic will get all the remaining bandwidth assigned. This potential over-reservation of bandwidth will allow for faster transfers. Given the reserved bandwidth in each state, we can obtain the expected bandwidth that is assigned to background traffic, by averaging over all states weighted by the steady state distribution:

$$\mathbb{E}[B_{bg}] = \sum_{\vec{x} \in \mathcal{S}} \pi(\vec{x}) b_{bg}(\vec{x}) B_{bg}. \quad (8)$$

In states where the demand exceeds the actual reservation, $n_{bg}(\vec{x}) > b_{bg}(\vec{x})$, there is under-reservation. Under-reservation could lead to traffic being blocked in the network, and should be kept as low as possible. The probability that a state is encountered where there is under-reservation, can be found by:

$$\mathbb{E}[U_{bg}] = \sum_{\vec{x} \in \mathcal{T}} \pi(\vec{x}) \quad (9)$$

where $\mathcal{T} = \{\vec{y} \mid \vec{y} \in \mathcal{S} \wedge b_{bg}(\vec{y}) < n_{bg}(\vec{y})\}$, the subset of all states that have under-reservation.

## V. RESULTS

In this section we perform a model based evaluation where we demonstrate how the different parameters in our model affect the streaming performance and the resource allocation for background traffic. Further, we perform a weighted sum optimization to construct the final sharing policy. We want to stress that the analysis below contains only a few example scenarios. Different environments require different input parameters, and may lead to different results. For this reason, we propose our model as an analytical tool, and allow it to be adapted to the environment.

### A. Scenario

We consider the scenario of a single public Wi-Fi hotspot that is used by a large number of users at the same time. Users can access the Internet for any purpose, but a DASH based video-on-demand service is offered. A DANE is deployed to manage network resource sharing. The capacity of the network will be $C = 26000$ kbit/s. This fairly well represents the maximum throughput of a 54 mbit/s Wi-Fi network, that is about half the theoretical throughput. The background demand is configured as $\lambda_{bg} = 0.1$ and $\beta_{bg} = 50$, and will result in a mean demand of 10 mbit/s, while showing reasonable variability. Furthermore, such a demand in background traffic will leave enough bandwidth to deploy a video streaming service. The two zones that are reserved for either DASH- or background traffic are set to set to $Z_{dash} = Z_{bg} = 6000$ kbit/s, to guarantee bandwidth for either type of traffic. The mean video duration is set to 140 seconds, and is available in the following bitrates: $\{300, 400, 500, 750, 1000, 1500, 2000, 3000, 4000, 6000\}$. A segment size of 4.0 seconds is used.

### B. Sharing policies

The different policies defines how many DASH players are allowed into the network, and how much of the shared bandwidth can be reserved for DASH players. In Figure 1 we compare the DASH priority policy with the background priority policy, in terms of the expected video quality. The available video bitrates are mapped to quality levels, 0 representing the lowest bitrate, 9 representing the highest.

Figure 1 shows that for a lower DASH demand, the DASH priority policy provides an increase of about one quality level. However, for $\lambda_d \geq 0.2$ the video quality will be much higher in favor of background priority mode, where the difference is at most four quality levels. This might seem counterintuitive at first, but is actually an effect of the limited admission of DASH players into the network for the background priority policy. Background priority keeps the number of DASH flows in the network low. This means, that when is a low demand from background traffic, the DASH streams are switching to higher bitrates, instead of allowing more flows into the network. The difference in the expected number of DASH players is illustrated in Figure 2.

Figure 2 shows that for $\lambda_d > 0.1$ the DANE is denying DASH players service, and thus keeps the number of players in the network limited. Although we did not work it out in detail for the sake of brevity, the video quality of DASH streams in background priority mode could also go down to the lowest

quality level. This happens when the demand of background traffic would be higher. In those cases, the probability that background traffic leaves bandwidth in the shared zone for DASH becomes very small.

In mixed mode, the quality of DASH streams can be expected to be in between that of DASH priority mode and background priority mode, as long as the quality of DASH flows in DASH priority exceeds background priority. For a higher DASH demand, the target bitrate is not likely to be reached due to a too large number of DASH flows in the network. In these cases, the video bitrate will be similar to the one in DASH priority policy.

Consider a moderate DASH demand with $\lambda_d = 0.08$ (in the remaining of this paper we will use this arrival rate for DASH flows). By increasing the target bitrate in mixed mode, the bitrate of the DASH streams can be increased. The set of target bitrates is equal to the available bitrates (i.e. it would not make sense to have other targets, because a target in between two available bitrates will result in the selecting the lowest of the two). Figure 3 displays the increase in video quality level when increasing the video bitrate. Interestingly, this increase is only little.

The reason that the increase in quality level is only little, is because every step to a higher bitrate is done by all players. Therefore, for the extra demand that is put on the network, the step in bitrate has to multiplied by the number of DASH players. Since the configuration in this example represents a moderately loaded network, there is not that much bandwidth left for increasing the video bitrate. Increasing the target bitrate will thus have a bigger effect in the states where the number of DASH players is lower, but when the load on the network is higher DASH players have to switch to lower bitrates anyway.

### C. DASH stream stability

Having only a marginal increase in bitrate when increasing the target bitrate, as displayed in Figure 3, does not mean that setting the target bitrate is of little importance. Video bitrate is only one of the factors that contribute to the quality of experience of the viewer. Another important factor is the number of changes, and the size of these changes, during
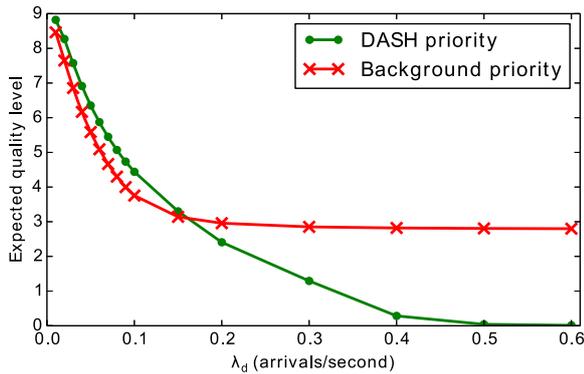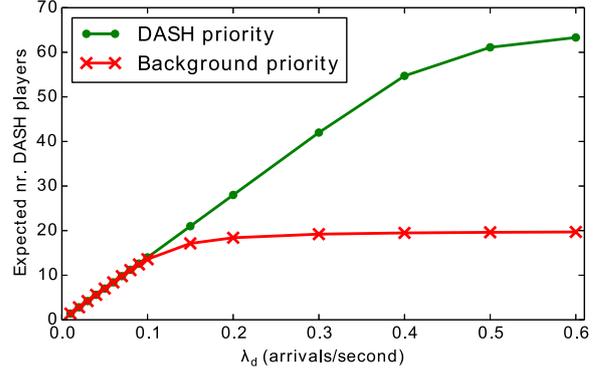


Fig. 2. Effect of DASH demand on the number of DASH flows. A comparison of the DASH priority- versus the background priority policy.
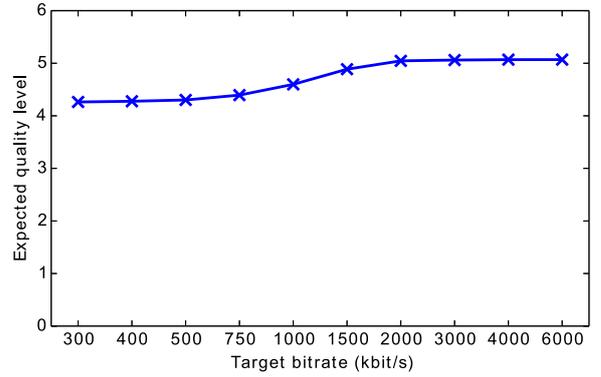


Fig. 3. Effect of target bitrate (mixed mode) on mean video quality

playout of a video stream. Figure 4 shows that the number of quality level changes (per minute) can be reduced by more than a half, when selecting the target bitrate of 1500 kbit/s.
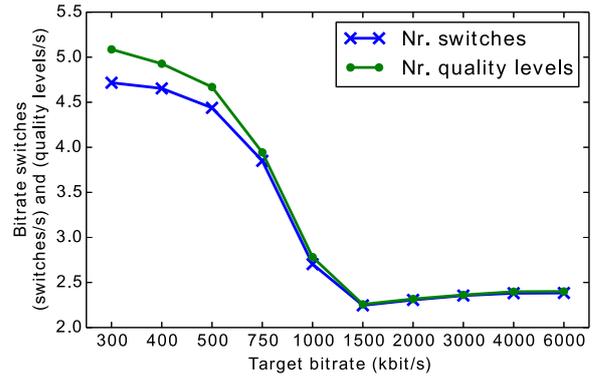


Fig. 4. Effect of target bitrate on number, and size, of quality switches.

At target bitrate 1500 kbit/s, the mean bitrate of DASH flows is close to 1500 kbit/s as well. Therefore, it is more likely



Fig. 1. Effect of DASH demand on video quality level. A comparison of the DASH priority- versus the background priority policy.

that DASH flows have to make smaller bitrate switches, or no switches at all, compared to the other target bitrates. Figure 4 also shows a difference in the the number of bitrate switches, and the number quality levels that is switched in between, for target bitrates up until 1500 kbit/s. This means that at least for some of the bitrate switches, the size of the switch was bigger than one quality level. We expect that the difference between the number of switches and the number quality levels becomes smaller, when increasing the target bitrate, as an effect of the sensitivity of to changes in background demand. For the lower target bitrates, background traffic has higher priority over DASH traffic compared to the higher target bitrates. This difference in sensitivity to background traffic is shown in Figure 5. This figure shows the number of bitrate switches for target bitrates $\{300, 1500, 6000\}$, while increasing the speed of variations in background demand.
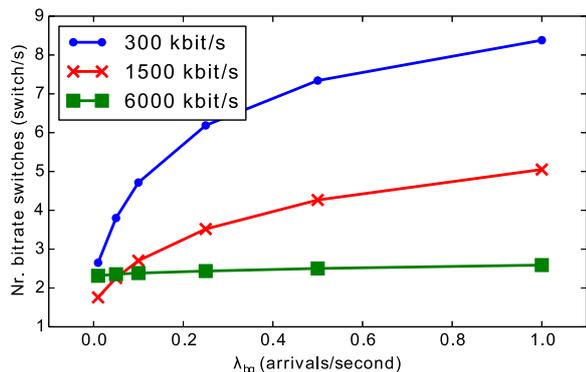


Fig. 5. Effect of background demand instability on DASH stability. A comparison of target bitrates (mixed mode) 300 kbit/s, 1500 kbit/s, and 6000 kbit/s.

For each point, $\lambda_{bg}$ and $\beta_{bg}$ are chosen such that they result in a background demand of 10 mbit/s (i.e. $\beta_{bg} = 5/\lambda_{bg}$). This figure very well shows that lower target bitrates are much more effected by instability in the background demand. It also shows that at the highest target bitrate shows almost no increase in bitrate switches. This is to be expected, as selecting the highest target bitrate equals the DASH priority policy, for which it was the goal that background demand has the as little effect on DASH as possible. Based on this analysis, we can conclude that if video bitrate stability is an important, selecting higher target bitrates would better. However, selecting a high target bitrate might negatively impact the overall performance of a DANE. For example, increasing the video bitrate lowers the available bandwidth for background traffic, which at some point could be problematic. We will address these conflicting objectives in the next section.

### D. Optimizing performance

A typical approach to improve video quality is to increase the bitrate. However, if we consider the same configuration as above, then the increase in bitrate that can be accomplished by increasing the target bitrate is less than one quality level (from 4.26 at for target bitrate 300 kbit/s, to 5.07 for target 6000 kbit/s, a difference of only 0.8 quality levels). Therefore,

in this scenario it might not be interesting to increase the target bitrate for the goal of a higher video bitrate. However, as we showed in Figure 4, the number and size of the bitrate switches can be greatly reduced. While increasing the target bitrate, one must be careful though not to have a too large background traffic under-reservation. Not reserving enough bandwidth for background traffic may have the negative side-effect that some of the background traffic has to blocked. In Figure 6 we plot the probability that background traffic demand the actually reserved bandwidth for background traffic. The figure shows a steep curve from target bitrates higher than 750 kbit/s.
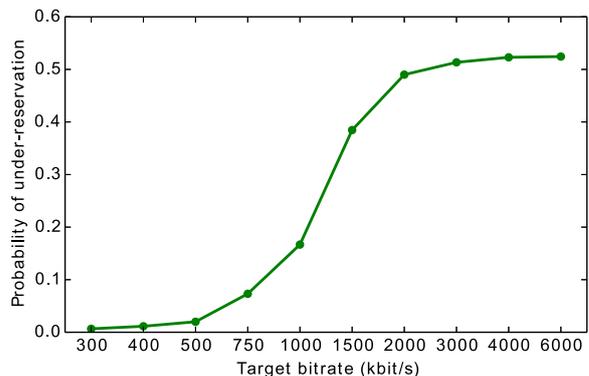


Fig. 6. Effect of target bitrate (mixed mode) on under-reservation of bandwidth for background traffic.

From the examples above, we can distill that there are three objectives for which we can optimize: video bitrate, number of bitrate switches, and background traffic under-reservation. However, optimizing for one objective could potentially conflict with another objective. For example, increasing the video bitrate also increases bandwidth under-reservation, that should be kept low. Furthermore, increasing the video bitrate is only beneficial for target bitrates up to 1500 kbit/s. Therefore, optimizing the sharing policy becomes a trade-off between bitrate, bitrate switches, and background traffic under-reservation. We employ the weighted sum optimization method to find the optimal target bitrate given these three objectives.

We define three optimization objective functions, $Q_b(t), Q_s(t), Q_u(t)$, that define how well each target bitrate scores in terms of video bitrate, quality switches, and background demand under-reservation. Each function $Q_x(t)$, is a linear projection of the best possible value (in this scenario) to 0.0, to the worst possible value to 1.0. Given three weights for each optimization objective, $w_b = 0.25, w_s = 0.25, w_u = 0.5$, we can formulate the optimization function as follows:

$$\min \quad f(t) = w_b Q_b(t) \cdot w_s Q_s(t) \cdot w_u Q_u(t)$$
$$\text{subject to} \quad t \in \mathcal{B} \tag{10}$$

where $\mathcal{B} = \{300, 400, 500, 750, 1000, 1500, 2000, 3000, 4000, 6000\}$, the set of target bitrates. Note that the weights in this example are chosen arbitrarily, as we are not network administrators and have no particular goals for a network
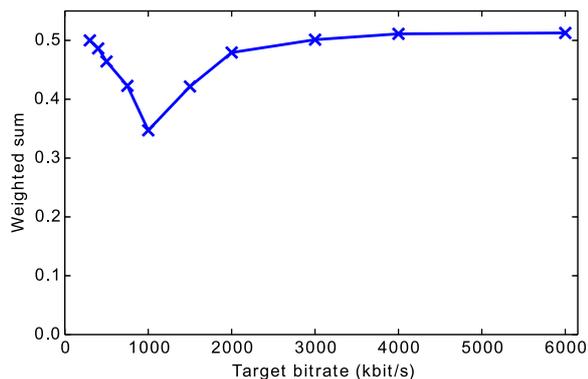
Fig. 7. Weighted sum optimization function including video bitrate, bitrate switches, under-reservation probability of background traffic (lower is better).

environment. Therefore, we selected weights such that DASH traffic has the same weight as background traffic, and video bitrate has the same weight as bitrate switches. The result of the optimization function, $f(t)$ is depicted in Figure 7.

Figure 7 shows that, given the optimization function $f(t)$ and the goal to find the minimum, the optimal target bitrate is 1000kbit/s. By selecting this target bitrate, we have reduced the number of quality switches by 55%, while keeping the video bitrate more or less the same, and the probability of background demand under-reservation below 17%.

## VI. CONCLUSION

Video streaming is a popular networking application. However, it requires relatively high bandwidth for the duration of the stream, and therefore it can put a significant demand on the network. In busy networks, that are used by many users simultaneously, it is important that network resources are properly divided among the services, to maintain a satisfying experience. In this paper we have proposed a performance model to evaluate resource sharing policies, that can be applied to network connections that need to handle a high demand, containing both video streaming and other traffic. Our model is consistent with DASH, the technology that nowadays dominates the video streaming market, and DANEs that are the executors of the resource sharing policies. As such, our model provides the streaming bitrate, and the number and size of switches in bitrate over time, as a measure of quality of experience. Furthermore, it allows to estimate the resources that are reserved for DASH and for background traffic, as well as provide an indication of how likely it is that not enough resources are allocated.

With the large interest of research in managing network resources including DASH, and also due to the standardization efforts of the Server and Network Assisted DASH (SAND), it will be likely that the number of DANE deployments will grow in the future. Our model will then be a useful tool for network administrators, that want to better configure their networks. Decisions on resource sharing are not necessarily straightforward, and thus require tools for analysis. By making an analysis with our model similar to the one we performed in

this paper, it can be understood what are the implications of sharing decisions on streaming performance or on non-video services. Compared to real deployments or network simulators, our model is more accessible as it allows for a quick evaluation of many configurations.

In previous work, we validated an early version of our model for evaluating DASH streaming performance, using our streaming testbed including real DASH streams [3]. That model showed high accuracy, but was limited to networks with only DASH traffic. In this paper we build on top of that fundament, but the model presented in this paper accounts for background traffic as well. However, as the concept of how to analyze the Markov process to obtain the streaming performance could be transferred to the current model, we expect that this model will yield high accuracy as well.

Nevertheless, future efforts will be on providing an evaluation of our model that compares it to real deployments. Furthermore, we will will study what background traffic patterns are common in, for example, public Wi-Fi hotspots. By analyzing traces from such networks, we will be able to reconfigure our model, and apply it to optimize resource sharing in these networks. Moreover, computing the streaming performance with our model is a matter of seconds on modest PC hardware. This potentially allows the model to be used online, and select the optimal resource sharing given the current demand. The next steps with our model are geared towards this direction.

## REFERENCES

[1] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2014. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6913491

[2] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, pp. 320–332, 2015.

[3] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar, "Modeling Stability and Bitrate of Network-Assisted HTTP Adaptive Streaming Players," in *27th International Teletraffic Congress (ITC 27)*, Ghent, Belgium, Sep. 2015.

[4] S.-E. Elayoubi and J. Roberts, "Performance Evaluation of Video Transcoding and Caching Solutions in Mobile Networks." *International Teletraffic Congress*, pp. 55–63, 2015. [Online]. Available: http://dx.doi.org/10.1109/ITC.2015.14

[5] M. Seufert, T. Hosfeld, and C. Sieber, "Impact of intermediate layer on quality of experience of HTTP adaptive streaming," in *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE, 2015, pp. 256–260. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7367367

[6] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *International Journal of Human-Computer Studies*, vol. 64, no. 8, pp. 637–647, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581905002028

[7] F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica, and H. Zhang, "Understanding the Impact of Video Quality on User Engagement," *Commun. ACM*, vol. 56, no. 3, pp. 91–99, Mar. 2013. [Online]. Available: http://doi.acm.org/10.1145/2428556.2428577

[8] D. C. Robinson, Y. Jutras, and V. Craciun, "Subjective Video Quality Assessment of HTTP Adaptive Streaming Technologies," *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 5–23, 2012. [Online]. Available: http://dx.doi.org/10.1002/bltj.20531

[9] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *IMC '12: Proceedings of the 2012 ACM conference on Internet measurement conference*. New York, New York, USA: ACM Request Permissions, Nov. 2012, pp. 225–238. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2398776.2398800

[10] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An Experimental Evaluation of Rate-adaptation Algorithms in Adaptive Streaming over HTTP," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*. New York, NY, USA: ACM, 2011, pp. 157–168. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943574

[11] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" in *NOSSDAV '12: Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*. New York, New York, USA: ACM Request Permissions, Jun. 2012, pp. 9–14. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2229087.2229092

[12] J. Esteban, S. A. Benno, A. Beck, Y. Guo, V. Hilt, and I. Rimac, "Interactions Between HTTP Adaptive Streaming and TCP," in *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*. New York, NY, USA: ACM, 2012, pp. 21–26. [Online]. Available: http://doi.acm.org/10.1145/2229087.2229094

[13] E. Thomas, M. O. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG dash performance via server and network assistance," in *The Best of IET and IBC*. Institution of Engineering and Technology, 2015, pp. 48–53. [Online]. Available: http://digital-library.theiet.org/content/conferences/10.1049/ibc.2015.0014

[14] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," in *MMSys '12: Proceedings of the 3rd Multimedia Systems Conference*. New York, New York, USA: ACM Request Permissions, Feb. 2012, pp. 1–9. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2155555.2155557

[15] N. Bouten, J. Famaey, S. Latré, R. Huysegems, B. D. Vleeschauwer, W. V. Leekwijck, and F. D. Turck, "QoE optimization through in-network quality adaptation for HTTP adaptive streaming," in *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm)*. IEEE, 2012, pp. 336–342.

[16] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, pp. 28:1–28:24, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2818361

[17] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *FhMN '13: Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. New York, New York, USA: ACM Request Permissions, Aug. 2013, pp. 15–20. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2491172.2491181

[18] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An SDN architecture with DASH assisting network elements," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: ACM, 2016, pp. 4:1–4:10. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910599