A Scalable User Fairness Model for Adaptive Video Streaming over SDN-Assisted Future Networks

Mu Mu*, Matthew Broadbent†, Arsham Farshad†, Nicholas Hart†, David Hutchison†, Qiang Ni†, Nicholas Race†

* The University of Northampton, Northampton, NN2 6JD, United Kingdom

mu.mu@northampton.ac.uk

† Lancaster University, Lancaster, LA1 4WA, United Kingdom

f.lastname@lancaster.ac.uk

Abstract

The growing demand for online distribution of high quality and high throughput content is dominating today's Internet infrastructure. This includes both production and user-generated media. Among the myriad of media distribution mechanisms, HTTP adaptive streaming (HAS) is becoming a popular choice for multi-screen and multi-bitrate media services over heterogeneous networks. HAS applications often compete for network resources without any coordination between each other. This leads to Quality of Experience (QoE) fluctuations on delivered content, and unfairness between end users. Meanwhile, new network protocols, technologies and architectures, such as Software Defined Networking (SDN), are being developed for the future Internet. The programmability, flexibility and openness of these emerging developments can greatly assist the distribution of video over the Internet. This is driven by the increasing consumer demands and QoE requirements. This paper introduces a novel user-level fairness model *UFair* and its hierarchical variant *UFair^{HA}*, which orchestrate HAS media streams using emerging network architectures and incorporate three fairness metrics (video quality, switching impact and cost efficiency) to achieve user-level fairness in video distribution. The UFair$^{\mathrm{HA}}$ has also been implemented in a purpose-built SDN testbed using open technologies including OpenFlow. Experimental results demonstrate the performance and feasibility of our design for video distribution over future networks.

Index Terms

Hierarchical resource allocation, adaptive media streaming, software defined networking, QoE utility fairness, network orchestration, human factor

I. INTRODUCTION

Driven by an increase in the demand for bandwidth in the Internet, recent advances in media technologies, and the growing popularity of smart phones and tablet computers, video streaming over the Internet has quickly risen to become a mainstream "killer" application over the past two decades [20]. Globally, IP video traffic is estimated to be 79% of all consumer Internet traffic in 2018, up from 66% in 2013, with nearly a million minutes of video content crossing the network every second [6]. HTTP adaptive streaming (HAS) protocols, especially the MPEG Dynamic Adaptive Streaming over HTTP (DASH), are becoming a popular vehicle for online video delivery, thanks to their unique adaptation feature that allows dynamic selection of quality representations in the face of network fluctuations. Despite its advantages and features, the increasing popularity of HAS deployment has also led to a number of challenges related to *rate adaptation components*, *rate adaptation strategies* and *quality of user experience* [20], which have also been highlighted in a recent J-SAC Special Issue on Adaptive Media Streaming [38].

Importantly, most HAS protocols adopt TCP as the transport layer protocol. This greatly mitigates the impact of network impairments to the delivered video quality. TCP aims to increase bandwidth utilization whilst avoiding congestion, which allows the application to fully exploit the available network resources. The adaptation between representations is often managed at the client side and is not specified by the DASH standard. Recent years have seen an increasing number of single-stream HAS optimization algorithms [14], [37], [22], [31] with the main objective being to maximize the quality of user experience using bandwidth estimation, client-side buffer management, and QoE measurement. However such optimization algorithms work on individual user clients, without any coordination with other devices in the same network. This leads to QoE fluctuations and unfairness when multiple HAS streams on heterogeneous user devices compete for network resources. Conventional fairness models, such as proportional fairness [17], are not suitable for HAS applications where video streams with varying characteristics exhibit distinctive utilities. Additionally, crucial HAS QoE impact factors such as the representation switching impact are not incorporated in current HAS QoE modeling.

One solution to address user-level fairness is to orchestrate the consumption of network resources using user-level fairness models. A number of recent work studied the support of adaptive video streaming in emerging networking technologies, protocols and Internet architectures such as Information-Centric Networking (ICN) and Software-Defined Networking (SDN) [39], [19], [10]. A network-wide orchestration for QoE fairness can be managed at the application level, the network level or a hybrid of both. For instance, Server and Network Assisted DASH (SAND), which is being actively developed within the MPEG community [1], is a control plane framework that allows user clients and content providers to coordinate content delivery. By pairing emerging concepts and technologies, namely SDN and Network-Function Virtualization (NFV), we can enable network-wide programmability, which when combined with flexible network services, allows us to realize functions such as the QoE-aware resource allocation demonstrated herein. This vision is also shared by a number of related works on network function control and QoS provisioning using deep packet inspection [9], [3].

This paper introduces a novel user-level fair network resource allocation mechanism called *UFair* to orchestrate adaptive video streaming using emerging network architectures, such as SDN. It enables the fair sharing of network resources; fairness, in this case, respects the characteristics of media distribution in the future Internet, and human perception of adaptive media. UFair, agnostic to underlying orchestration frameworks, incorporates three tailored fairness metrics based on *video quality*, *switching impact* and *cost efficiency* and uses context-aware resource allocation to maintain fair use experience over media applications. We carried out a number of experiments to verify the performance of UFair with different degrees of network fluctuations and various numbers of competing user clients.

With the aim to support complex network topologies, we also designed a hierarchical fair resource allocation model UFair$^{\mathrm{HA}}$, which slices a network into a hierarchy of zones (based on physical or virtual topology). The model derives aggregated utility for each zone so that the resource provisioning can be conducted between zones, followed by the allocation within each zone. Verified by the quantitative complexity evaluation, such a design greatly reduces UFair's operational complexity whilst maintaining its overall performance. UFair$^{\mathrm{HA}}$ is also implemented in a SDN testbed using OpenFlow and OpenStack. Experimental results demonstrate the feasibility of adopting user and media-level contextual information via the UFair$^{\mathrm{HA}}$ for the improved content distribution with respect to user-level fairness. The remainder of the paper is organized as follows: Section II explores background and related work in the field of network management for HAS streams. The design of the UFair is detailed in Section III. Section IV introduces the experiment set-up as well as the results. Discussions and conclusions are given in Section V.

II. Background and Related work

In order to support a myriad of user devices over heterogenous networks, HAS streams are designed to encapsulate content in multiple dedicated formats. Using MPEG-DASH as an example, media content is encoded in multiple *representations*, each of which is a version of the same content prepared using different encoding specifications (such as bitrate and frame resolution), and tailored for specific scenarios. A number of associated representations are then gathered together to form an *Adaptation Set*. Each representation is served with time-coded chunks, addressable using URLs and retrievable via HTTP. By exploiting the switching points between chunks, “seamless” adaptation is possible. Upon a playback request, the user client receives a *Media Presentation Description (MPD)*, a manifest file which describes all of the resources and structure information required to stream that video. A DASH client often starts from a representation that matches its screen resolution and at modest bitrates. Once the client detects an increase in available bandwidth, it can switch to a higher bitrate.

Most HAS protocols adopt TCP as the transport mechanism. TCP aims to increase bandwidth utilization whilst avoiding congestion. This enables the HAS application to fully exploit the available network resources and attempt to maximize the quality of a video stream. However, this presents two challenges in delivering a good quality of user experience, especially in a shared network environment with heterogeneous user devices. The first challenge is the network fluctuation caused by the packet delivery scheme. A DASH client may continuously inflate its receiver window during TCP ON periods. This inadvertently forces the sender to burst as much traffic as possible, until either enough video chunks are buffered at the client (which then switches to OFF mode), or until the sender incurs TCP packet loss. This behavior causes extremely bursty traffic and results in TCP inefficiency, as connections are repeatedly restarted between ON/OFF periods, resulting in unstable video playback quality [2], [13]. The second challenge is the gap between network-level fairness and user-level fairness when network resources are shared between multiple user applications. Conventionally, the goal of resource allocation is to maximize the aggregate utility of all applications in the network subject to the capacity constraints of the network [17], [26]. One known implementation based on this theory is Kelly's *proportional fairness* [17]. Mo et al. and Marbach also investigated a fairness criterion called max-min fairness using a family of utility functions to approximate an arbitrarily close max-min fair allocation [30], [28], [41]. Proportional fairness performs well when all user applications follow the same utility. When users have different QoS needs, proportional fairness favors users which require lower rates to achieve high utility [26]. On the contrary, resource provisioning to high-demand users/applications only contributes to a small increase in the aggregate utilities. Therefore, maximizing the combined utility does not necessary lead to fairness between users. This approach is efficient for congestion control and is fair in terms of bandwidth allocation only when all the sources attain the same (strictly concave and increasing) utility functions. It is impractical to assume that all network applications have the same QoS requirements for bandwidth and their QoS utilities satisfy the strict concavity condition [41]. The concept of application-specific utilities was hence suggested as part of the *utility max-min* and *utility proportional fairness* work [4], [41]. Meanwhile, research in the field of video quality assessment shows that there is no linear correlation (video quality utility) between the bitrate of a video stream and its perceptual quality [5]. This suggests that a constant increase in bitrate could lead to either a significant gain in resultant video quality or a very limited quality improvement that is barely noticeable by the end user. In order to optimize the efficiency of network resource allocation whilst maintaining a satisfactory level of user experience, it is essential to incorporate content-level and user-level utility of a media stream. Furthermore, most of the current work on utility-based fairness makes the assumption that network links have fixed capacity for user applications. In practice, available network resource for media streaming can change over time due to fluctuating link capacity in wireless networks [40] or influence of other traffic. There is also currently a lack of integration of context related to the adaptive media. For instance, it is inefficient to provision resources to a user application beyond the range of its representations defined by the content provider.

Recent years have seen significant work put forward to improve the QoE of adaptive video distribution. One solution is to have some cross-layer interaction between TCP and HTTP in order to provide the streaming application with better metrics and to allow TCP to reach steady-state [13]. This would indeed improve TCP performance, but would not control the ON/OFF nature of DASH-style applications. Furthermore, it would not attain network-wide fairness across all devices. Tian and Liu [37] use throughput-prediction algorithms to reduce video rate fluctuations. Mansy et al. [27] have shown that DASH's bursty nature leads to excessive queuing in the network (a phenomenon commonly referred to as bufferbloat [11]), and they proposed adjusting DASH's buffering behavior to keep the size of the client's receiver window low. FESTIVE [16] attempts to improve fairness, stability and efficiency using a DASH player with a stateful, delayed-bitrate update mechanism. Huang et al. introduce a buffer-based approach to rate adaptation to reduce the rebuffer rate in online HAS streaming [14]. A client-side rate-adaptation algorithm for HAS is also introduced in [22]. Mok et al. integrated an available bandwidth measurement into the video data probes for improved rate selection [31]. Georgopoulos et al. incorporated a video quality utility as part of work towards network-wide fairness [10]. Recent work also covered the support of adaptive video using emerging networking technologies, protocols and architectures such as HTTP/2.0, Information-Centric Networking (ICN) and Content-Centric Networking (CCN) [39], [19].

Most of the aforementioned work focuses on optimizing the network efficiency or the quality of user experience for individual media streams. User applications that adopt more intelligent and aggressive packet forwarding schemes will likely achieve higher QoE against other HAS applications in the same network. Without coordination between HAS clients, we may see additional network resources being allocated to video streams that have already exceeded user expectation, while the same resource could be better used to significantly improve the QoE on other media streams. This would also lead to severe fluctuations in multi-stream environment observed by Li et al. [22]. Furthermore, the overall user experience of a HAS stream can be greatly affected by the switching between representations. This is often overlooked by related work on single-stream and multi-stream quality optimization. Overall, there is currently a lack of research addressing the user-level fairness of network resource provisioning in a multi-HAS-stream environment. With the increasing number of high fidelity, high resolution and high throughput HAS streams delivered in future Internet, quality assurance via over-provisioning becomes less feasible. Instead, it is necessary to orchestrate the network resource consumption through a better understanding of the user-level requirements of user applications, especially the resource-intensive HAS applications. However, most of the new network architectures such as ICN and SDN are not designed specifically for adaptive video streaming. It is essential to develop a resource allocation model that integrates emerging network and application-level functions to ensure the QoE and user level fairness of media streaming in future networks.

III. User-level fairness model

The ultimate goal of the user-level fairness UFair model is to orchestrate network resource allocation between HAS streams to mitigate QoE fluctuations and improve the overall QoE fairness. The underlying principle of the UFair model is depicted in Figure 1.

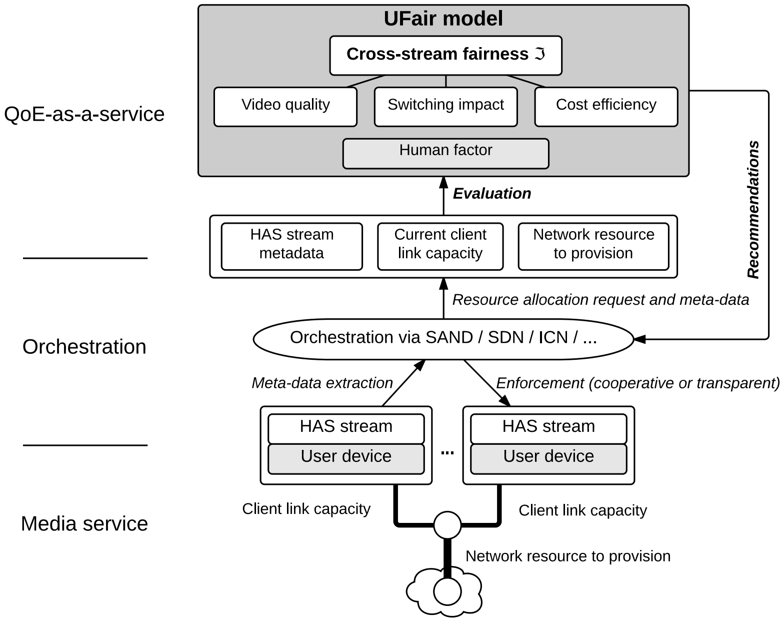


Fig. 1. User-level fairness model

UFair addresses the user-level fairness by incorporating crucial QoE metrics that are directly determined by the provisioning of network resources. We define three novel base metrics; video quality, switching impact and cost efficiency, with user-level QoE mapping, to capture the user experience over adaptive media. Using these base metrics, UFair defines the cross-stream fairness $\Im$ based on the discrepancy (relative standard deviation) between the QoE measurements on related media streams, and orchestrates the allocation of resources to maximize the overall fairness. The assessment of QoE metrics requires: 1) input parameters including context information regarding HAS streams, such as current playback bitrate, resolution, etc., 2) current network capacity at user devices, and 3) total bandwidth to share between multiple devices. Both the network capacity and the total bandwidth are dynamic and can be affected by the change of link capacity (in wireless networks) or background traffic. The input parameters can be derived using a network-level or application-level QoE orchestration framework.

Currently there are two such frameworks being widely investigated. 1) The MPEG community proposed the Server and Network Assisted DASH (SAND) architecture [1]. SAND is a control plane for video delivery that obtains QoE metrics from the users (clients) and returns network-based measurements. The third-party measurement server in SAND, known as DANE (DASH-assisting network element), provides measurement information to the different parties in the delivery chain including CDNs, ISPs and content providers. This enables user clients and content providers to coordinate using the same framework and provide the necessary information for the UFair model. 2) Recent advances in networking, such as SDN and NFV, enable network-wide flexibility and programmability. This facilitates comprehensive network and service functions to be deployed easily and in an on-demand fashion. Leveraging such emerging networking architectures, functions to conduct network- and application-specific measurements can be implemented and maintained within the network and operated transparently without client intervention. Our recent work on an SDN-assisted QoE framework can be found in [7]. With UFair model determines the optimal user-level resource allocation strategy, recommendations can be enforced by network management functions.

Using conventional network equipment and designs, it is impractical to tailor the allocation of network resources between a large number of concurrent media flows. SDN, especially when complimented by NFV, allows centralized control and management application such as load balancing, monitoring, and traffic analysis to be realized with SDN controller operates as part of a service chain [29]. Furthermore, while SDN technologies are rapidly adopted in data centres, there has been a growing number of designs and experimentations of SDN infrastructure architectures [43] and home network SDN deployment [18]. The remainder of the section discusses the incorporation of fairness metrics. Based on this design, a scalable UFair model using hierarchical design is introduced in Section IV.

A. *Video quality fairness*

In order to fairly share network resources with respect to the user experience, it is crucial to understand and model the impact factors related to the delivered video quality. A HAS application chooses an optimal resolution for the playback device and dynamically selects a representation (of a certain bitrate) from the adaptation set according to the available bandwidth (or buffer occupancy). Based on the assumption that the encoding scheme does not vary significantly between representations, a higher encoding bit-rate results in less compression loss and therefore yields higher video quality pertaining to picture fidelity. A video-quality utility (*VQ*) function (a type of rate-distortion function) is often employed to capture the non-linear relationship between bitrate and video quality. It captures the notion of the law of diminishing returns [34] – a certain addition of resources to what one already has increased the total worth, but it contributes less and less to the increase if one has more of the resource already.

To model the video-quality utility function, we selected a reference source video file of an animated film called "Big Buck Bunny". These rushes are widely used by researchers in the area of adaptive content distribution. Our previous work also investigated the influence of content characteristics by employing multiple types/genres of rushes [32]. Such complex modelling is considered as extended feature of the QFair model for future study. We acquired the uncompressed YUV video files in 360p, 720p and 1080p resolution and the FLAC audio file. Then, we encode the source files using the H.264/AAC audio-visual codec. The stated resolutions are selected to represent typical use case scenarios. Nowadays, modern devices could be equipped with even higher resolution screens and corresponding computing resources, but we believe that the three selected resolutions encompass a significant number of use cases. We generated 22 test video sequences with various predefined bitrates for each resolution, with respect to practice. We employed Structural Similarity (SSIM) index [42], an objective quality assessment model, to quantify the quality difference between each encoded test sequence and its uncompressed version.

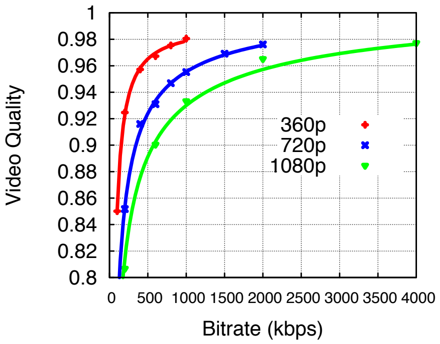


Fig. 2. Scatter plot and utility curves of bitrate utilities

Figure 2 shows the scatter plot and the fitted utility curves of the HAS video bitrate utility in the three named resolutions. The figure reflects the common understanding that more resources are required to deliver video of the same quality on higher resolutions. The utility plot quantifies such a relationship between video quality and bitrate.

$$Q_{res} = ar^b + c \tag{1}$$

$$Q_{720p} = -4.85r^{-0.647} + 1.011 \tag{2}$$

Equation 1 is the generic QoE utility function. r denotes the video bitrate and the Q is the video quality. A Q of 1 is the maximum possible video quality (when no compression or lossless compression is applied to the content). a, b, and c are the coefficients that instantiate the utility function for certain video resolutions. For instance, Equation 2 is an instance of utility function for 720p videos. The utility function is of low complexity (suitable for real-time quality assessment) and yet offers high performance. Equation 2 shows significant correlation (R^2 of 0.9983 and $RMSE$ of 0.002923) to the observed experimental results in our previous work [10]. Table I gives details of the utility functions and corresponding coefficients.

TABLE I

MODEL AND COEFFICIENTS FOR VIDEO QUALITY UTILITY FUNCTION

General Model	**Two-term Power Series Model** $f(r) = ar^b + c$			**Goodness of Fit**	
For	a	b	c	Adjusted R^2	RMSE
1080p	-3.035	-0.5061	1.022	0.9959	0.006011
720p	-4.85	-0.647	1.011	0.9983	0.002923
360p	-17.53	-1.048	0.9912	0.9982	0.002097

In practice, content providers offer a finite set of representations in an adaptation set and the highest quality offered for the content (Q_{res}^{TOP}) is defined in a content manifest such as the MPD by the best representation ($r = r^{TOP}$) in the adaptation set. By adopting such a companion content descriptor provided by the content provider, we prevent the over-provisioning of resources beyond the capacity of HAS streams and therefore avoid HAS streams with higher top representations being penalized. To this end, we are able to further improve the video quality fairness measurement by incorporating both the modeling of user perception and the nature of media applications. We rescale $\mathcal{U}_{res}(r)$ so that the video quality reaches its maximum value 1 when the best representation is active (i.e., $Q_{res}^{TOP} = {\mathcal{U}'}_{res}(r^{TOP}) = 1$). Hence the adjusted video quality utility function is:

$${\mathcal{U}'}_{res}(r) = \frac{\mathcal{U}_{res}(r)}{\mathcal{U}_{res}(r^{TOP})} \tag{3}$$

In practice, the maximum feasible quality of a stream is also limited by the network capacity at the user device. The network capacity determines the highest bitrate feasible (r^{MAX}) for the corresponding media stream. For instance, a user may have only 2Mbit/s network capacity using Wi-Fi networks in her garden, though she subscribes to 50Mbit/s broadband network over DSL. Hence, it is unnecessary to provision more than 2Mbit/s of network resource on the shared access network for this user. The network capacity is often determined by the link capacity and any background traffic on the same link. The video quality utility is therefore adjusted to reflect the network resource constraint at a user device:

$$\mathcal{U}'_{res}(r) = \frac{\mathcal{U}_{res}(r)}{\mathcal{U}_{res}(r^{MAX})}, \; if \; r^{MAX} \; < \; r^{TOP} \tag{4}$$

Using the tailored video quality utility functions, we can then divide network resources between HAS media streams while minimizing any discrepancy between the delivered video quality on all HAS streams, hence ultimately achieving the video quality fairness.

$$\begin{gathered} \mathcal{U}'_{res_1}(r_1) = \mathcal{U}'_{res_2}(r_2) = ... = \mathcal{U}'_{res_N}(r_N), \\ \text{with } r_1 + r_2 + ... + r_N = B \end{gathered} \tag{5}$$

Overall, the optimal video quality fairness between HAS media streams can be achieved mathematically using Equation 5 as influenced by the available bandwidth and HAS media information such as the adaptation set. The fairness between media streams is measured using Relative Standard Deviation (RSD) (Equation 7), obtained by multiplying the standard deviation s by 100 and dividing this product by the mean $\bar{Q}$. RSD captures not only the deviation but also the scale of the video quality difference. A small RSD is a result of less difference between video quality perceived over the related HAS streams, which suggests better fairness at the user level. Note that the representation of fairness in this paper is different from the conventional fairness indices such as Jains Index [15] and α-fairness [30]. The fairness is maximized when $\Im$ reaches 0 which reflects "zero difference" between a user-level metric measured on all media streams. A higher $\Im$ denotes more differences, hence worse in fairness. Such a fairness index allows multiple fairness indices to be easily re-scaled and combined for advanced user-level evaluation.

$$s_{VQ} = \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(Q_j - \bar{Q})^2} \tag{6}$$

$$\Im^{VQ} = s_{VQ-RSD} = 100 \times \frac{s_{VQ}}{\bar{Q}} \tag{7}$$

B. Switching impact fairness

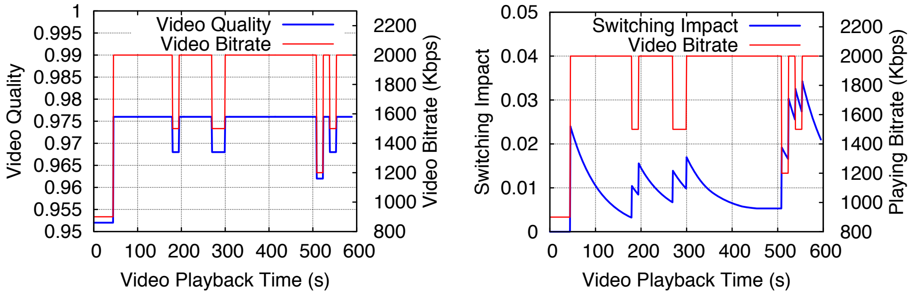
HAS media streams switch between representations as the means to adapt to the available network resource. Quality switching can be triggered to increase the bitrate for the improved video quality or to reduce the bitrate for avoiding any buffer underrun and playback stalling. However, the switching process itself may cause disturbance to the end user. The impact of quality switches (denoted as *SI*) is influenced by the *amplitude* and the *distribution* of switching events [8]. The amplitude is determined by the perception of video quality changes between representations. We define such quality change as $\Delta_{VQ} = |Q - Q'|$ with Q' as the projected video quality after the representation switch. A higher change of video quality leads to more severe perceptual impact at the time of switch. In a related work, Liu et al. observed that the impact caused by "increasing switch" is much smaller than "decreasing switch" of the same scale [24]. The modeling of this advanced feature requires further subjective experiments, which will be carried out in our future work. A crucial aspect when modeling the HAS switching impact is the *forgiveness effect* (first studied and modeled by Seferidis et al. [35] and Hands [12]), which captures the psychological observations that, when followed by intact content, the impact of quality distortion degrades over time. One of the key findings from the user ratings is that the impact of quality distortion is reduced to nearly 70% after 20 seconds. We incorporate the *forgiveness effect* (Equation 8) based on a generalized model introduced by Liu et al [25]. Equation 8 is a function of intensity of quality changes (Δ_{VQ}) and the duration of time since a switching event ($t - t_i$).

$$\begin{gathered} SI_i(t) = (\Delta_{VQ})\mathrm{e}^{-0.015(t-t_i)}, \\ t_i \text{ is the time of the quality switch } i \end{gathered} \tag{8}$$

Using Equation 8, the initial switching impact will eventually diminish to a negligible value when $t - t_i$ is sufficiently large. We consider the QoE as the overall acceptability of a video session as perceived by human user. Therefore, we define 10% of the initial switch impact as a residual influence that lasts for the user's entire viewing session. This means that the residual impact from multiple switching events will accumulate till the end of the viewing session. The impact function is updated as:

$$\begin{gathered} SI_i(t) = max((\Delta_{VQ})\mathrm{e}^{-0.015(t-t_i)}, 0.1\Delta_{VQ}), \\ t_i \text{ is the time of the quality switch } i \end{gathered} \tag{9}$$

Figure 3(a) shows the video quality (VQ) for playing the test video with options to switch between different video bitrates in 720p video resolution. The figure demonstrates the non-linear mapping between the video bit-rate and the video quality. For instance, a switch between two very high bitrates shows less



(a) Video quality

(b) Switching impact

Fig. 3. Impact of HAS adaptation

impact on the video quality compared with the same amount of bitrate change between representations of lower bitrates. Such QoE measurements are valuable for both single-stream quality optimization and QoE fairness between media streams. Switching impact accounts for the frequency and distribution of changes over the playing time. As demonstrated in Figure 3(b), high switching impact can be caused by high video quality variation or small, but temporally close, changes.

With the help of the switching impact measurement, we can evaluate a resource allocation solution or compare different solutions using the following switching impact fairness function based on RSD:

$$s_{SI} = \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(SI_j - \bar{SI})^2} \tag{10}$$

$$\Im^{SI} = s_{SI-RSD} = 100 \times \frac{s_{SI}}{\bar{SI}} \tag{11}$$

During network fluctuations, representation switching is inevitable on one or multiple HAS streams. Switching impact fairness captures the impact of switches throughout the entire life-cycle of a HAS stream, and balances such impact between related HAS streams. As an example, a relatively high RSD measure on switching impact suggests that one or more HAS streams have experienced more frequent or severe quality adaptations. Using the SI fairness metric, the UFair model can mitigate such imbalance and potentially protect the playback bitrate of certain HAS streams from further variations.

C. *Cost efficiency fairness*

Content consumers, especially those having invested in high-definition TV and broadband internet connections, expect video to be delivered at the highest possible quality. Distributing high throughput video content is not possible without a high degree of guaranteed network bandwidth. Such requirements place great challenges on network operators, particularly during prime time when a large amount of concurrent video streams must be supported by shared network resources. High throughput video traffic can also overwhelm "vulnerable" segments of delivery networks and deteriorate packet delivery of other applications. It is in network operators' interests to assure user satisfaction on HAS video streams whilst moderating the utilization of network resources. We define cost efficiency *CT* as a metric to capture the notion of fairness between content consumers and network operators. CT quantifies *the required (or consumed) bandwidth per unit of total targeted (or delivered) video quality*. A high CT denotes low cost efficiency as it requires more bandwidth to deliver a unit of video quality. Given the bitrate of selected representations of related video streams and their adjusted utility functions $\mathcal{U}'$, CT can be quantified using Equation 12. Unlike video quality fairness and switching impact fairness, the cost-efficiency fairness is evaluated based on all related HAS media streams as a whole over the measured network segment.

$$\Im^{CT} = \frac{\sum\limits_{i}^{N} r_i}{\sum\limits_{i}^{N} U'_{res_i}(r_i)} \tag{12}$$

It is also possible to determine the most (theoretically) cost-effective bandwidth-provisioning solution(s) using Lagrange multipliers to find the minimal value(s) of Equation 12 subject to the constraint $\sum\limits_{i}^{N} r_i \leq B$. However, a fairness model built entirely based on CT would most likely favor bitrates towards the lower end of the chart due to the nature of utility curves (Figure 2). Therefore, CT fairness should be in principle exploited in balance with at least a complementary metric such as video quality.

D. Fairness-aware resource allocation

Using video-quality fairness, switching-impact fairness, and cost-efficiency fairness as the user-level metrics, a QoE service can provision bandwidth fairly with respect to the user perception of video content, and cost efficiency of the network to deliver good user experience. Incorporating the fairness metrics in production networks either as a network service or as a QoE middleware function poses a number of challenges. The adaptation sets of HAS streams comprise of discrete and finite representations, hence the optimal solution to share available bandwidth between media streams cannot be derived directly from any continuous utility functions. Ultimately a decision is made from the many combinations of representations for each media streams. For the case that N HAS streams, each comes with M representations, are present for bandwidth sharing, a total number of N^M combinations are available for evaluation against each fairness metric. Every new stream joining the network would increase the number of combinations by M-fold. When multiple QoE metrics are incorporated, the complexity of the fairness model will also increase accordingly. For stateless metrics such as VQ and CT, which do not depend on media streams' historical session status, it is possible to employ techniques such as dynamic programming to improve runtime performance [21]. Stateful metrics like SI require historical information related to quality switches of the video session, hence it is more difficult to reduce their runtime complexity.

In order to improve the feasibility of the UFair model for live resource allocation optimization, we developed an optimization method of three internal stages. At the first stage, the framework uses the continuous VQ utility functions to derive the optimal sharing of bandwidth which aims at an identical degree of video quality on all HAS streams. The process also maximizes the utilization of total available bandwidth to share. This is done by solving the following equation of adjusted utility functions:

$$\mathcal{U'}_{res_1}(r_1) = \mathcal{U'}_{res_2}(r_2) = ... = \mathcal{U'}_{res_N}(r_N),\ with\, r_1 + r_2 + ... + r_N = BW \tag{13}$$

Because VQ utility functions are monotonically increasing, Equation 13 gives at most one set of results: $\hat{R} = [\hat{r_1}, \hat{r_2}, ..., \hat{r_N}]$. The second stage takes the optimal solution given by the continuous utility functions as the starting points, and conducts a bi-directional search of the nearest representations of every optimal bitrate in $\hat{R}$ as defined in MPD. The search returns one or two playback rates for each $\hat{r}$:$[[r_1^l, r_1^h], [r_2^l, r_2^h], ..., [r_N^l, r_N^h]]$. $[r_i^l, r_i^h]$ are the bitrates of representations that best approximate the optimal rate of $\hat{r_i}$, with $r_i^l \leq \hat{r_i}$ and $r_i^h \geq \hat{r_i}$. In the cases that $\hat{r_i}$ is higher than the highest representation or lower than the lowest one, only r_i^l or r_i^h will be available. The searching stage serves as a screening process that greatly reduces the complexity of resource allocation between N streams with M levels of representations from M^N (exhaustive search) to a much more manageable candidate list C of a maximum of 2^N items.

The last stage of the optimization process evaluates the candidate list C using three fairness metrics and identifies the candidate that achieves the best balance between the three. The process begins by calculating video quality fairness $\Im_c^{VQ}$, switching impact fairness $\Im_c^{SI}$, and cost fairness $\Im_c^{CT}$ of all c in candidate list C. We then continue with a pooling process by combining all three measurements and deriving an overall rating for each c. In order to aggregate fairness metrics in different scales, we rescale the fairness measurements using the maximum observed value as the rescaling factor. For instance:

$$\ddot{\Im}_c^{VQ} = \frac{\Im_c^{VQ}}{max(\Im_C^{VQ})} \tag{14}$$

The re-scaled fairness measurement $\ddot{\Im}_c$ has the data range of $[0, 1]$. $\ddot{\Im}_c = 1$ represents the worst solution from all candidates with respect to a given fairness measurement. Any value between 0 and 1 quantifies relative fairness level with in the candidate list. Using the rescaled fairness measurements, we then combine the three fairness measurements using the weighted-sum method:

$$\begin{gathered}\ddot{\Im}_c^{combined} = w_c^{VQ} * \ddot{\Im}_c^{VQ} + w_c^{SI} * \ddot{\Im}_c^{SI} + w_c^{CT} * \ddot{\Im}_c^{CT}, \\ with\ w_c^{VQ} + w_c^{SI} + w_c^{CT} = 1\end{gathered} \tag{15}$$

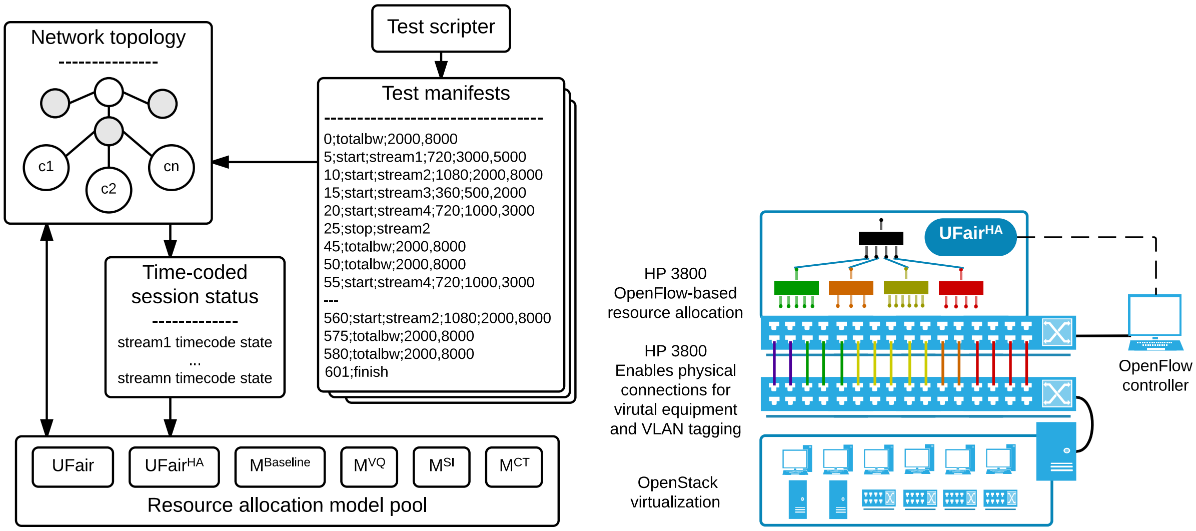
w_c is the weight coefficient for each fairness metric and it defines how fairness of video quality, switching impact and cost is balanced. We use an equal balance between three fairness metrics (i.e., $w_c^{VQ} = w_c^{SI} = w_c^{CT}$) to investigate the impact of each fairness metric to the overall resource allocation solution. In practice, a QoE management framework may adopt a different balance between the metrics with respect to its management scenario. The candidate solution c which exhibits the minimum value of combined $\ddot{\Im}_c^{combined}$ value is considered to be the best option to achieve the overall user-level fairness.

IV. Experiments

A. Discussion on model realization and testbed setup

In Section III, we explored the the potential exploitation of the UFair model in the content-distribution eco-system: the SAND architecture and the in-network QoE orchestration architecture using SDN technologies. Each approach has its advantages and disadvantages. Using a SAND-based solution, an open platform is made available between content providers, network operators and end-hosts; the combination of which enables accurate QoE measurement. Managing HAS media sessions is also more convenient with co-operative user clients. However, this approach requires the participation of a number if distinct parties and it is not yet clear whether such a design will be feasible across multiple domains and providers. On the other hand, the SDN/NFV-based solution is designed to work transparently as a network service, so it does not rely on co-operation between user devices and service operators. Extracting media information, deploying a resource-allocation function, monitoring available bandwidth, and the control of network traffic are non-trivial tasks. In a recent work, we introduced an in-network QoE measurement framework that provides QoE monitoring for HAS streams, and made an initial deployment in a large-scale SDN testbed [7]. The framework leverages control plane flexibility, enabled through the use of SDN technologies, to streamline non-intrusive quality monitoring and to offer a closed control loop for QoE-aware service management. Using the packet-inspection function offered by the framework, we extract media information such as the MPD, current bitrate, and the resolution of HAS streams which are prerequisites for QoE-aware fair resource allocation.

In order to assess the effectiveness of the UFair model under different network conditions, we developed a purpose-built evaluation testbed as shown in Figure 4(a). Using test profiles (which comprise of number of clients, frequency of network fluctuations, client link capacities, etc.), the test scripter function generates randomized network events for test manifests. A testbed function parses any given test manifest and generates client arrival/departure and network fluctuation accordingly. The resource-allocation model pool encapsulates a number of APIs which allow the testbed to specify network status and metadata of HAS streams, and acquire solutions to optimize resource allocation between all relevant media streams. Session logs, which capture time-coded HAS representation changes for all media streams, are also maintained for stateful metrics such as SI fairness. In order to study the characteristics of each fairness metric in achieving user-level fairness, we employed three additional fairness models (M^{VQ}, M^{SI}, M^{CT}), each exclusively uses one of the three fairness metrics ($\Im^{VQ}$, $\Im^{SI}$, and $\Im^{CT}$) to direct resource allocation. We also incorporate a baseline model ($\mathrm{M}^{\mathrm{baseline}}$), which resembles how network resources are provisioned through TCP without the help of an overarching orchestration framework. For a given experiment, the testbed initiates one independent thread for each model in the resource allocation model pool. This allows us to comparatively study the results from each model under identical test conditions.


Network topology
----
c1
c2
cn
Test scripter
Test manifests

Time-coded session status
----
stream1 timecode state
...
streamn timecode state
UFair
$\mathrm{UFair}^{\mathrm{HA}}$
$\mathrm{M}^{\mathrm{Baseline}}$
M^{VQ}
M^{SI}
M^{CT}
Resource allocation model pool
HP 3800 OpenFlow-based resource allocation
HP 3800 Enables physical connections for virutal equipment and VLAN tagging
OpenStack virtualization
$\mathrm{UFair}^{\mathrm{HA}}$
OpenFlow controller

```
0;totalbw;2000,8000
5;start;stream1;720;3000,5000
10;start;stream2;1080;2000,8000
15;start;stream3;360;500,2000
20;start;stream4;720;1000,3000
25;stop;stream2
45;totalbw;2000,8000
50;totalbw;2000,8000
55;start;stream4;720;1000,3000
---
560;start;stream2;1080;2000,8000
575;totalbw;2000,8000
580;totalbw;2000,8000
601;finish
```

(a) Framework of evaluation testbed

(b) OpenFlow-based experimentation

Fig. 4. Testbed setup

The testbed is also realized in a purpose-built experimental environment. This consists of an OpenStack deployment connected to a number of OpenFlow-enabled hardware switches. OpenStack is an open-source cloud computing software platform, which enables us to create, rebuild and destroy a variety of networks and virtual machine instances almost instantaneously; an important facet when our experiments require a large number of user clients and switches. This setup reduces the number of physical components to just one high-specification server and 2 Ethernet switches. Despite this simplicity, all network-based measurements are made using physical links, guaranteeing realism. Each virtual machine instance will serve as either a client or server in our experiment. In the case of the client, we use Scootplayer[1], a highly configurable MPEG-DASH compliant player with support for accurate logging.

On the networking side of the testbed setup, we choose OpenFlow-ready switches. OpenFlow is a communications protocol that facilitates control of the forwarding plane in a switch; a key requirement in realizing a Software Defined Network. In particular, the HP E3800s we deployed support version 1.3 of OpenFlow; a requirement for our experimentation. More specifically, we desired the inclusion of *meter tables*: a construct used to define a number *meter entries*, each of which is a per-flow rate-limiter. This enables complex QoS operations to be realized in-network (provided by the outcome of UFair).

The resource allocation function requires metrics to calculate allocations in an accurate and timely fashion. These measurements are also gathered using the OpenFlow protocol, which enables both fine-grained (flow-based) and coarse-grained (port-based) *counters*, the results of which are then aggregated over time at the OpenFlow controller[2]. The controller can then offer a delta to the resource allocation function by means of a simple API call.

Importantly, the switch also allows multiple logical OpenFlow instances to run simultaneously on a single underlying hardware switch. Each instance presents itself to the OpenFlow controller with a unique *datapath ID*, and consists of a subset of the available physical ports. In order to map the network interface of each VM to a physical port on the hardware switch, we configure multiple virtual networks within OpenStack. Each of these networks is configured with a different Ethernet VLAN (802.1Q) which is then exposed to the physical world over a single VLAN trunk connected to an Ethernet switch. On this intermediary switch, each VLAN is configured to a single port (i.e. a port based VLAN). In this way, a basic 48-port switch can be used as 48 physical ports for an OpenStack hypervisor, while the actual connection between OpenStack and switch is just a single multi-gigabit interface.

Overall, the combination of features, programmability, and openness provided by OpenFlow greatly assist us to fully realize the fair orchestration in real-world networks. Although this may have been technically possible beforehand, creating a solution around vendor-specific interfaces has limited applicability; with OpenFlow, we can create a generic solution that should work across a multitude of vendors and within a variety of scenarios. This is particularly important when scalability and interoperability are core requirements, as is the case with this work. This use case also demonstrates the benefits of cross-layer cooperation between emerging network architectures and novel media technologies in future Internet.

[1] http://github.com/broadbent/scootplayer

[2] http://osrg.github.io/ryu/

B. Non-equal resource sharing

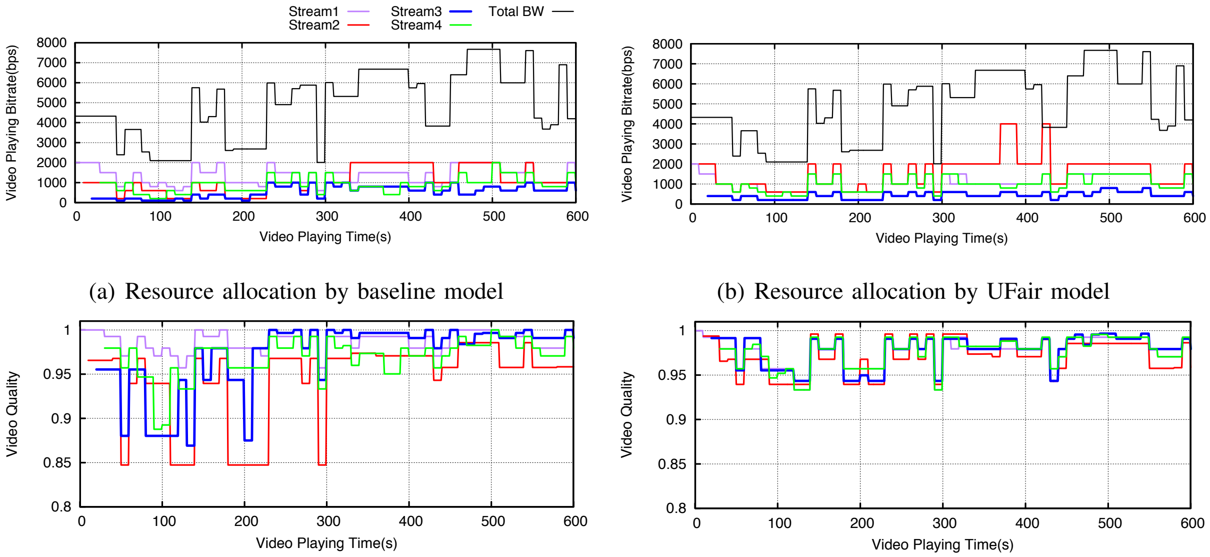
For the first test, we compare how the baseline model and UFair provision network resource differently in a dynamic environment. The test is specified using a generic tree structure. A number of clients (c_1,...,c_n) connect to an aggregation node via corresponding links (l_1,...,l_n) and share the resource of an access network. One HAS stream is delivered to each of the clients. The first test is defined with the duration of 10 minutes and four clients (streams). The total available bandwidth accessible to video streams on the access network randomly fluctuates between 2Mb/s to 8 Mb/s as influenced by background traffic. The resolution of video streams delivered to the clients is randomly selected between 360p, 720p, or 1080p. The bitrates for each resolution are given in Table II. The available bandwidth of each client link also changes randomly between 500kb/s to 8Mb/s during the course of the test to reflect changes of link capacity (in wireless networks) or background traffic. The total available access network bandwidth is shared between media streams with respect to the resource available on client links.

TABLE II

SET BITRATES FOR THREE VIDEO RESOLUTIONS

Resolution	**Video Bitrate (kbps)**
1080p	100, 200, 600, 1000, 2000, 4000, 6000, 8000
720p	100, 200, 400, 600, 800, 1000, 1500, 2000
360p	100, 200, 400, 600, 800, 1000

Figure 5(a) and Figure 5(b) give details of how network resource is shared between four video streams in the first 65 seconds of the test as instructed by the two different models. The resultant video quality of each video stream is given in Figure 5(c) and Figure 5(d) for the baseline model and the UFair model respectively. The results clearly demonstrate the significant differences of the network provisioning strategy adopted by the user-level model compared with the conventional TCP-based network-level baseline model. The baseline model allows video streams with more intensive requests at the transport layer to acquire more resources, leading to some video streams being heavily penalized (Figure 5(a)). Using the first 20 seconds of the test as an example, stream2, stream3 and stream4 all suffered from low video quality and severe quality fluctuation while the quality of stream1 remains high through the entire test (Figure 5(c)). This example demonstrates the gap between network-level and user-level fairness. Using the bespoke UFair model, which takes advantage of three fairness metrics, the network-management element in the testbed is able to schedule the resource according to the QoE requirements and link status of every HAS stream (Figure 5(b)). As a result, network resources are dynamically provisioned in a way that similar video quality is maintained on all related media streams for the entire course of the experiment (Figure 5(d)). Furthermore, the UFair model also avoided any severe video quality fluctuation thanks to its incorporation of the switching-impact fairness.



(a) Resource allocation by baseline model

(b) Resource allocation by UFair model

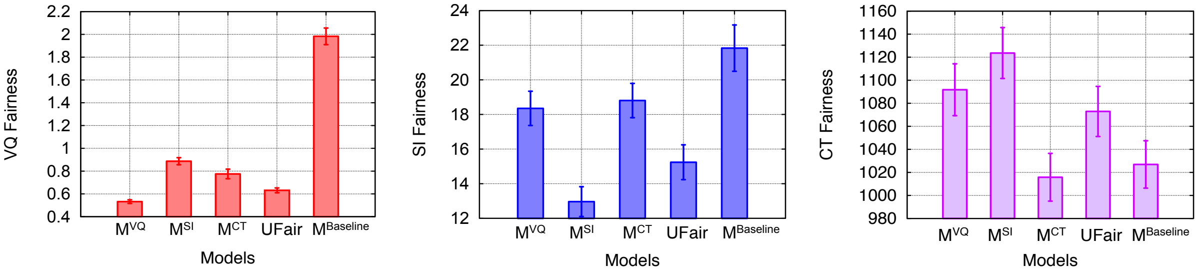
(c) Video quality delivered by baseline model

(d) Video quality delivered by UFair model

Fig. 5. Resource allocation and resultant video quality of UFair and baseline model

In order to further investigate the performance of the UFair fairness model and specifically how each individual fairness metric contributes to the user-level fairness, we defined a test manifest similar to the first test and enabled all five fairness models (UFair, $\mathrm{M}^{\mathrm{baseline}}$, M^{VQ}, M^{SI}, M^{CT}). Test manifests are defined with respect to a test scenario such as “busy wireless home network with a DSL broadband connection”. It specifies the number of HAS streams, and the overall frequency at which the total shared bandwidth and the network capacity at user devices fluctuate. The exact timing and scale of the dynamics are purposely randomized and will only be instantiated at run time. Therefore every iteration of the test will generate a unique test configuration of a predefined scenario and hence test results. This also allows us to evaluate the consistency of the model. Exploiting such a feature of the testbed, we repeated the test 50 times. Figure 6(a) compares how five models perform in terms of video quality fairness. It reflects our previous observations in Figure 5 that the UFair model significantly outperforms the baseline model (a lower value in fairness metric denotes better fairness). Between the M^{VQ}, M^{SI}, and M^{CT} models, M^{VQ} (whose objective is to maximize the video-quality fairness exclusively without considering other fairness metrics) yields the best results, unsurprisingly. The M^{SI} and M^{CT} models compromise on video-quality fairness to balance switching impact or cost fairness but still greatly outperform the baseline model.

The evaluation based on switching impact fairness is shown in Figure 6(b). Similar to the conclusions in Figure 6(a), all user-level fairness models achieve better performance than the baseline model, while we can maximize the switching impact fairness using the M^{SI} model. The results delivered by the M^{VQ} and M^{CT} models are between M^{SI} and baseline. The results on cost-efficiency fairness are slightly different (Figure 6(c)) to the other two. In this case, the baseline model exhibits marginally better performance in delivering cost-efficiency fairness compared with M^{VQ}, M^{SI} and UFair, and is only beaten by the M^{CT} model. This is due to the fact that gaining a unit of video quality is easier when the bitrate of video is low according to the video quality utility function which resembles the law of diminishing returns. With more streams in the lower-bitrate and lower-quality ranges, the baseline model can be more cost effective in terms of consumed bitrate per unit of delivered quality, though the delivered video quality is still much lower than UFair and other models as demonstrated in Figure 5(c).



(a) Video quality fairness

(b) Switching impact fairness

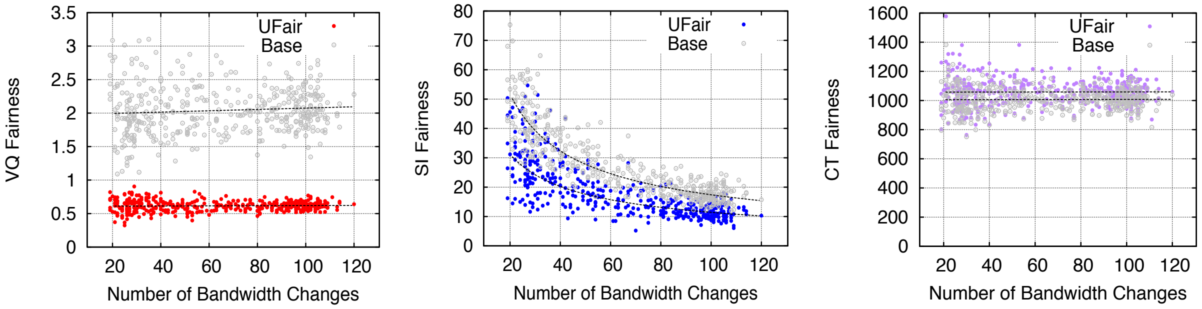
(c) Cost efficiency fairness

Fig. 6. Fairness measurements of resource allocation models

Overall, video-quality fairness, switching-impact fairness and cost-efficiency fairness all exhibit their distinctive benefits to the overall user-level fairness. A model achieving the best on one fairness metric usually shows sub-optimal performance on the other two fairness metrics. Experimental results suggest that the integration of multiple user-level metrics helps introducing a fair share of network resources.

C. Evaluate the impact of network fluctuations

In practice, a shared network can be very quiet or extremely busy. To study the consistency of the UFair model in face of network fluctuations of various degrees, we specified new test manifests by manipulating the probability of a bandwidth change using the test scripter. We generated a total of 400 10-minute-long tests with the number of bandwidth changes varying from around 20 (one change to the shared bandwidth every 30 seconds) to 120 (one change to the shared bandwidth every 5 seconds). Every change of the shared bandwidth leads to a reallocation process instructed by the resource-allocation function. Again, we use both the UFair and baseline models for a comparative analysis. The increasing number of quality fluctuations is believed to have an impact on the UFair model especially through its stateful SI metric, where every change of video quality is accounted for in the user experience.



(a) Video quality fairness

(b) Switching impact fairness

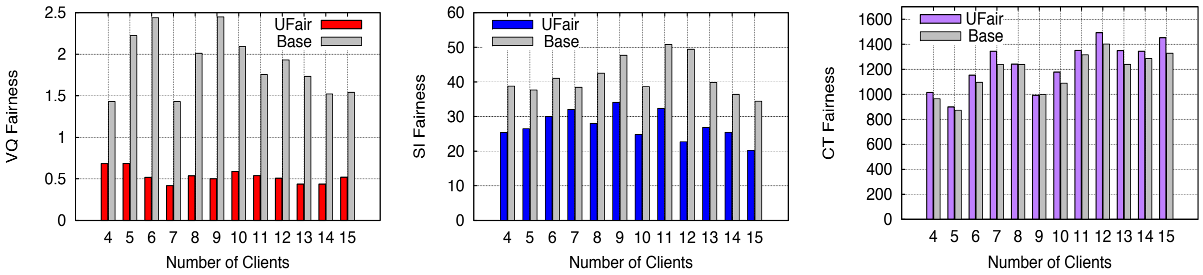
(c) Cost efficiency fairness

Fig. 7. Performance of UFair model influenced by the number of bandwidth fluctuations

Figure 7 compares how the UFair and baseline models deliver user experience on media streams using the three fairness measurements for networks of different characteristics. Each point on the figure projects the mean value of the corresponding fairness measure of the entire test. The UFair model achieves its design objectives of delivering a good level of video quality fairness and switching impact fairness whilst maintaining the cost efficiency compared with the baseline model. The SI fairness measurements are more scattered between tests of fewer bandwidth changes than the tests of frequent bandwidth changes (Figure 7(b)). This is due to the fact that switching impact is a stateful metric that recognizes the dependency between consecutive changes of playback bitrate. Hence, tests with larger numbers of quality switches are more likely to statistically capture the performance of the a model on SI fairness. Furthermore, as defined in Equation 15, the UFair model may be configured to balance between the three fairness metrics equally (as for the experiment), or to favour certain metric(s) with respect to a particular service strategy. For instance, a service provider may allow a level of discrepancy on video quality of media streams whilst giving more priority to maximizing the cost efficiency and minimizing the switching impact.

D. Scalability with the number of clients

One important performance index of a resource allocation algorithm is its scalability. We continue the experiments using test manifests that allows different numbers of user clients (media streams), ranging from 4 (used by previous tests) to 15. The total bandwidth is configured to be fluctuating between 2Mbps to 8Mbps for around 35 times during the course of the 10-minute-long test. This set-up helps us investigate whether the UFair model can perform with the same level of user-level fairness when more clients join



(a) Video quality fairness

(b) Switching impact fairness

(c) Cost efficiency fairness

Fig. 8. Performance of UFair model influenced by the number of clients

the shared network and share the same pool of network resources. Figure 8 suggests that increasing the number of clients does not significantly impact the output of the UFair model. There seems to be a trend of CT fairness reduction when the number of clients increases beyond 10. However, the fairness measurements still stay within the data range observed in Figure 7(c) where the number of clients is 4.

E. Scaling up using multi-tier hierarchical resource allocation

As networks become increasing complex and video applications over IP networks continue to gain popularity, resource allocation also faces challenges related to multi-tier topology, user data sharing, and cross-domain policy. Based on game theory, a number of hierarchical resource allocation designs have been proposed in the past. For instance, Tang and Jain studied the case where the existence of network resource resellers (i.e., middlemen) breaks traditional models, and proposed hierarchical auction mechanisms to induce an efficient Nash equilibrium [36]. Liang et al proposed a hierarchical game, which takes into account the interactions of resource allocation for backhaul and access links [23]. As is summarized in [33], game theory provides a good theoretical tool and yet it still faces challenges related to complexity, peer communication, and convergence time. In face of dynamic and heterogeneous applications of media streaming services in future Internet, we depart from the conventional designs and take a more practical approach by modelling the aggregated utility across tiers of hierarchical networks. Such design does not require interactions with (nor between) user applications while a global optimal solution can then be derived deterministically (avoiding the convergence issue).

1) Utility aggregation: One of the most common use scenario and network topology is illustrated in Figure 9. Media streams are initiated on multiple user clients in a network zone (such as a household, a premise or a VLAN) which are connected to the Internet via a gateway (e.g., $P1$) over an access network. Traffic from all network zones aggregate at an edge node (e.g., node B) before reaching a backbone network or another level of aggregation networks. The level of network resource available for media streams also changes over time as affected by the background traffic or link capacity. Whilst it's theoretically feasible to directly employ UFair model to provision network resource between all HAS streams over 16 user devices (as demonstrated in Section IV-D), the computational complexity may overload an aggregation node (such as the node B in Figure 9) when the networks scale up. Furthermore, exchanging detailed information and measurements related to media streams and user devices cross the boundaries of a premise (beyond the gateway) is sometimes prohibited due to privacy or user agreement.

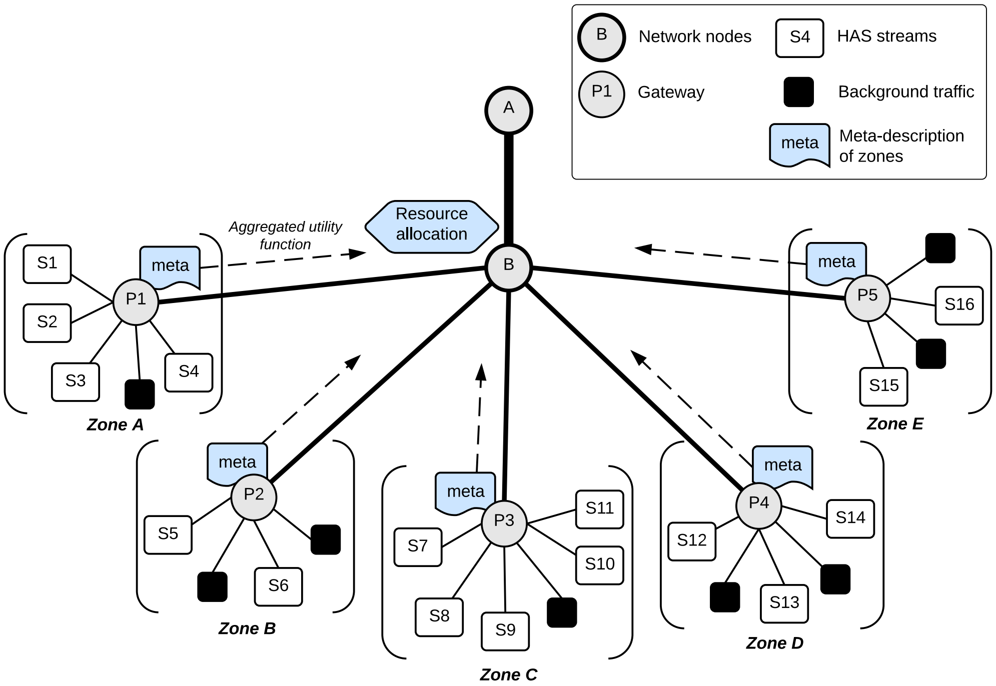


Fig. 9. Evaluation of UFair model in a 2-tier topology

In order to address the scalability challenge of resource provisioning in large-scale content distribution networks, we further extend the UFair model as a *hierarchical* resource allocation model UFair$^{\mathrm{HA}}$. Networks of user devices in a zone are abstracted as a super-node. A household is an example of a zone with physical network boundaries. However, a sub-network in a household, a group of households, or a virtual LAN (VLAN) comprises of networks in multiple physical locations can also be defined as a zone. Multiple zones can be further abstracted as an additional hierarchy of super-zones. In this paper, we take a two-tier network topology shown in Figure 9 as an example to introduce the principles behind the UFair$^{\mathrm{HA}}$ model. For each zone in the topology, a gateway node (such as a home gateway) coordinates all connected user devices and maintains a dynamic meta-description which defines the requirements of network resources of the HAS streaming over the user devices in the same zone. The meta-description contains an aggregated utility function derived from all utility functions of related HAS streams in the zone. It quantifies the mapping between the perceptual video quality and the required throughput for the zone as a whole so that the allocation of shared resources (e.g., over the network between node A and node B) can be carried out (by node B) between zones fairly in a way that the average user experience on all HAS streams in a zone is similar to the same measurements in other zones. In order to minimize the operational complexity at aggregation node (node B), the session information of HAS streams are used only within the boundaries of zones. The resource allocation between HAS streams in a zone is identical to the approach described in Section III-D.

The aggregation of utility functions in a zone is carried out using Equation 16. $\overset{-1}{\mathcal{U}_k}$ is the inverse function of the utility function (or inverse utility function) of a HAS stream k in a zone. It models how much network resource is required for a given quality level. The primary goal of the UFair model is to orchestrate HAS streams in a zone to maintain a similar degree of user experience and to gain (or drop) video quality synchronously. The inverse utility function is ideal to capture such resource allocation attribute for utility aggregation. The inverse aggregated utility function of a zone is defined as the sum of the all inverse utility functions for the related HAS streams. K is the total number of HAS streams in a zone. The principles behind the aggregated utility function also reflect the derivation of total resistance of parallel resistors in electrical circuits. The final aggregated utility function of a zone (Equation 17) is ultimately determined by the number and attribute (e.g., high definition or standard definition) of all HAS streams.

$$\overset{-1}{\mathcal{U}_{zone}} = \overset{-1}{\mathcal{U}_1} + \ldots + \overset{-1}{\mathcal{U}_K} = \sum_{k=1}^{K} \overset{-1}{\mathcal{U}_k} \tag{16}$$

$$\mathcal{U}_{zone} = \overset{-1}{\left(\sum_{k=1}^{K} \overset{-1}{\mathcal{U}_k} \right)} \tag{17}$$

With the aggregated utility functions of all zones, the UFair$^{\text{HA}}$ fairly provisions shared resources between zones:

$$\begin{gathered} \mathcal{U}'_{zoneA}(r_A) = \mathcal{U}'_{zoneB}(r_B) = \ldots = \mathcal{U}'_{zoneE}(r_E), \\ with\ r_A + r_B + \ldots + r_E = BW \end{gathered} \tag{18}$$

$\mathcal{U}'_{zone}$ is the adjusted aggregated utility function of a zone based on the zone's available link capacity (e.g., between node $P1$ and node B).

2) Performance and complexity evaluation: In order to evaluate the hierarchical resource allocation by UFair$^{\text{HA}}$, we defined tests based on the topology in Figure 9. The stream configurations for each of the five zones are given in Table III.

TABLE III

ZONE AND HAS STREAM CONFIGURATION

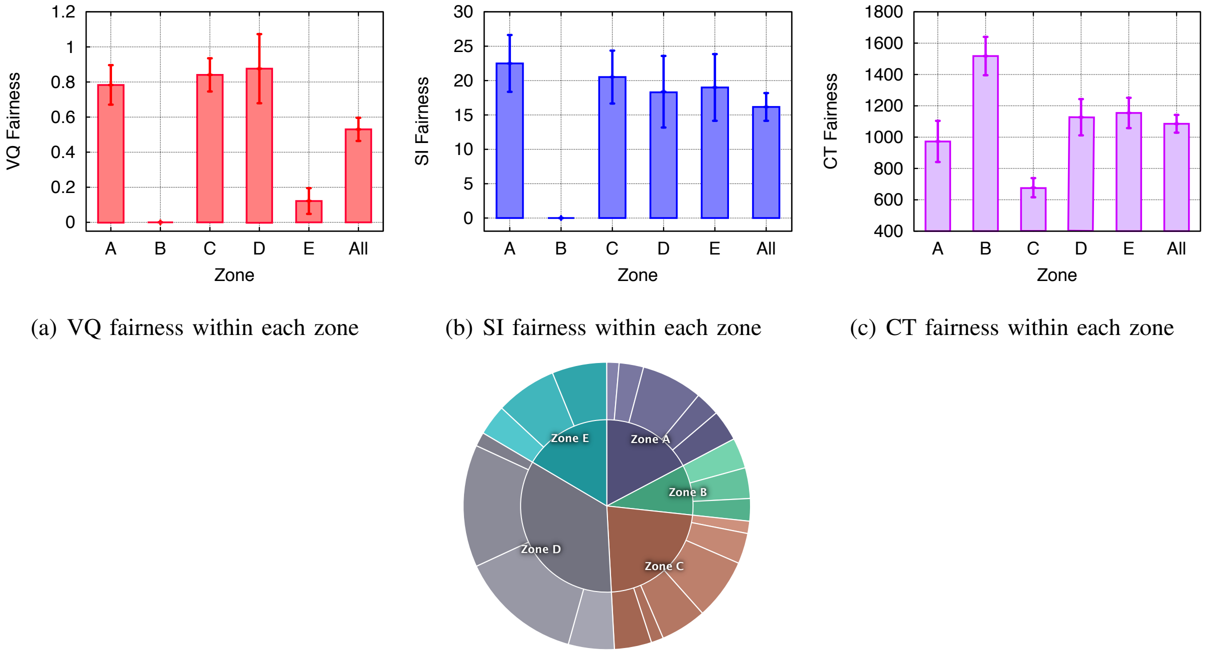
Zone A		Zone B		Zone C		Zone D		Zone E	
Stream ID	Type	Stream ID	Type	Stream ID	Type	Stream ID	Type	Stream ID	Type
stream1	360	stream5	720	stream7	360	stream12	720	stream15	360
stream2	720	stream6	720	stream8	720	stream13	1080	stream16	720
stream3	1080			stream9	1080	stream14	1080		
stream4	720			stream10	720				
				stream11	360				

Each zone has 2 to 5 HAS streams and there are distinctive combinations of stream types for us to study how the resource requirement from different zones are captured. There are 16 HAS stream in total. During the test, the total bandwidth between node A and node B fluctuates between 10Mbps and 30Mbps and the link capacity of all zones fluctuates between 3Mbps and 20Mbps. Upon every major changes in the network (e.g., total bandwidth increased by 5Mbps due to departure of background traffic) inter-zone resource re-allocation are carried out followed by inter-stream fair share within each zone independently and concurrently.

Figure 10 shows the fairness measurements of all zones and between all 16 streams as a whole. Overall, UFair$^{\text{HA}}$ achieved a similar level of fairness compared with results delivered by the UFair model. The readings of VQ fairness and SI fairness are particularly low for zone B due to the fact that the two streams in zone B share the same type and hence allocated with the identical amount of resources. The CT fairness figures are slightly higher than what's achieved by the UFair model. Although the hierarchical algorithm using aggregated utility functions can greatly improve the scalability of fairness-aware resource provisioning, it may lose the cost efficiency compared with resource distribution between HAS streams directly. Figure 10(d) illustrates the allocation of resources between zones and between clients in zones with respect to individual and aggregated QoE requirements.

The experiment is also repeated for 100 times with randomized network fluctuations. The results in Figure 11 demonstrate the consistency of the model's performance in achieving VQ, SI, and CT fairness between clients in all zones using the hierarchical algorithm.

In practice, the number of zones connected to an aggregation point in the topology shown in Figure 9



(a) VQ fairness within each zone

(b) SI fairness within each zone

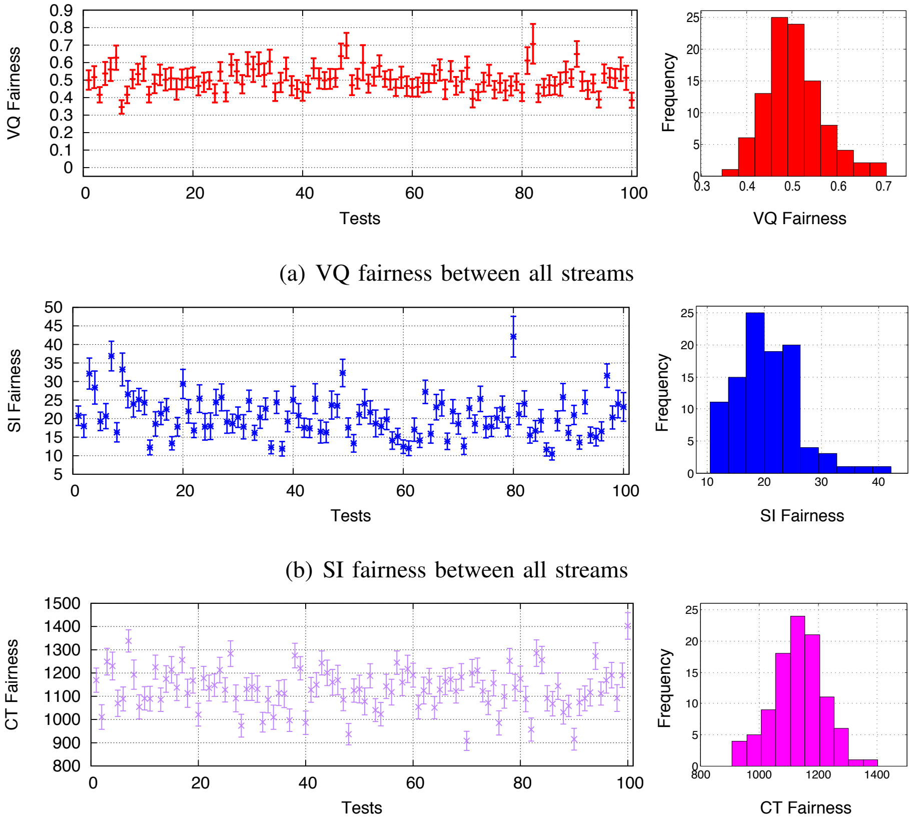
(c) CT fairness within each zone

(d) User fair allocation

Fig. 10. Fairness measurements

can scaled up. For instance, in the UK, British Telecommunications plc connects its exchanges to street cabinets (or Fibre aggregation nodes) via fibre-optic cables. The street cabinets then offers broadband to households or business premises in a size of tens or hundreds depending on the size and location of the cabinet. In order to investigate the impact from the size of the network to UFair$^{\mathrm{HA}}$, we further extended the experimental topology by increasing the number of zones step-wise from 5 to 50. Consequently, the total number of HAS streams is also gradually increased from 16 to 160. As is suggested by the results shown in Figure IV-E2, the number of zones does not significantly affect the resultant fairness orchestrated by the UFair$^{\mathrm{HA}}$ resource allocation model. For an even more complex network with a greater number of zones and HAS streams, UFair$^{\mathrm{HA}}$ may further scale up accordingly by defining multiple hierarchies of zones following the same principle demonstrated in our scalability tests.

Besides the scalability, we also evaluated the computational complexity of the multi-party resource allocation process driven by UFair and UFair$^{\mathrm{HA}}$. We use a metric directly correlated to the code execution time to quantify the run-time complexity when the number of HAS streams increases from 4 to 200. The results shown in Figure 13 compares resultant complexity when streams are grouped in different numbers of zones. The UFair model, which does not distribute media streams (and computational load) between zones, exhibits an exponential increase in computational complexity (y-axis is in *log* scale) when the

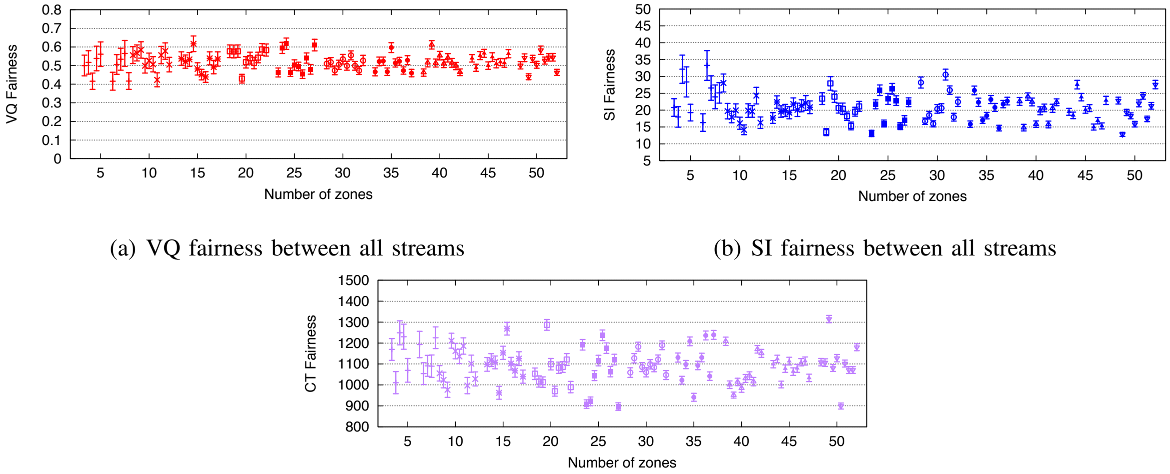


(a) VQ fairness between all streams

(b) SI fairness between all streams

(c) CT fairness between all streams

Fig. 11. Fairness measurements of 100 tests



(a) VQ fairness between all streams

(b) SI fairness between all streams

(c) CT fairness between all streams

Fig. 12. Model evaluation with different numbers of zones

number of stream increases. This is due to the fact that a single process must solve a function group in a size directly proportional to the number of related streams.

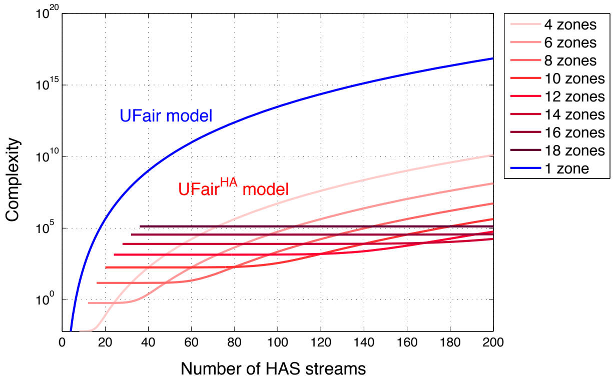


Fig. 13. Computational complexity of the UFair and UFair$^{\text{HA}}$ process

For UFair$^{\text{HA}}$, the evaluation is based on the configuration that each zone has at least two streams and the number of streams is identical between zones. Depending on the design of the network management framework, there is likely a level of additional overheads for the coordination between hierarchies. Benefit from the hierarchical principle and parallel intra-zone resource management, UFair$^{\text{HA}}$ shows significantly lower run-time complexity compared with the UFair model. The results between different zone distributions by the UFair$^{\text{HA}}$ manifest a shared feature: the complexity slowly increases (nearly unnoticeable on the log scale) before a significant change emerges. This is a direct result of computational load being distributed between zones. A higher volume of zones, with its additional initial cost, can better cope with the impact of HAS stream flash-crowd. In the cases that the zones can be flexibly managed, the hierarchical resource allocation function can also maximizes its run-time efficiency by adapting to the network topology using most optimal zone configurations defined in Figure 13.

V. Discussion and conclusion

When using HTTP adaptive streaming, user applications consume network resources independently without coordination between each other. This leads to QoE fluctuations and unfairness between end users. This paper introduces a user-level resource allocation model which maximizes the fairness between user experience on adaptive media streams. To achieve its objectives, our design incorporates three QoE metrics to best capture the QoE of adaptive media, and exploits emerging network architectures to orchestrate fair resource provisioning. The performance, scalability, and feasibility of the model is evaluated in a software defined networking testbed using open technologies such as OpenFlow.

For future work, the notion of user-level fairness still needs more exploration, not least in the context of QoS/QoE and network neutrality. Feasible metrics to best capture user experience and fairness indices must continue to evolve along with the development in media and content distribution technologies. A proposition to incorporate media and user-level context for network orchestration does not principally conflict with the framework of network neutrality. The utility and QoE metrics are defined to a class of application such as HAS streaming and should not be considered as the tools to favor selected service providers. The UFair model is an excellent basis for future experimentation which includes selected domains related to future Internet such as 5G, ICN and MPEG SAND. Future work will also incorporate contextual QoE factors such as the cumulative user experience over video streaming services.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council under Grant EP/L020009/1 (TOUCAN) and EP/K011693/1, and the European FP7 Programme CROWN project under Grant PIRSES-GA-2013-610524. The authors would also like to thank Lyndon Fawcett, Jamie Bird, and Steven Simpson for their technical contribution.

References

[1] ISO/IEC JTC1/SC29/WG11 (MPEG) Report. Technical report, ISO, November 2013.

[2] S. Akhshabi, A. Begen, and C. Dovrolis. An Experimental Evaluation of Rate-adaptation Algorithms in Adaptive Streaming over HTTP. In *Proc. 2nd annual ACM Conference on Multimedia Systems*, MMSys '11, pages 157–168, 2011.

[3] A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Koral. Deep packet inspection as a service. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 271–282. ACM, 2014.

[4] Z. Cao and E. W. Zegura. Utility max-min: An application-oriented bandwidth allocation scheme. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 793–801. IEEE, 1999.

[5] G. Cermak, M. Pinson, and S. Wolf. The relationship among video quality, screen resolution, and bit rate. *Broadcasting, IEEE Transactions on*, 57(2):258–262, 2011.

[6] Cisco. Cisco visual networking index: Forecast and methodology, 20132018. *http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf*, 2014.

[7] A. Farshad, P. Georgopoulos, M. Broadbent, M. Mu, and N. Race. Leveraging SDN to provide an in-network QoE measurement framework. In *IEEE INFOCOM 2015 Workshop on Communication & Networking Techniques for Contemporary Video*. IEEE, 2015.

[8] M.-N. Garcia, F. D. Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnstrm, and A. Raake. Quality of experience and http adaptive streaming: a review of subjective studies. *Proceedings of the 6th International Workshop on Quality of Multimedia Experience (QoMEX)*, 2014.

[9] A. Gember-Jacobson, R. Viswanathan, C. Prakash, R. Grandl, J. Khalid, S. Das, and A. Akella. OpenNF: Enabling innovation in network function control. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 163–174. ACM, 2014.

[10] 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*, pages 15–20. ACM, 2013.

[11] J. Gettys and K. Nichols. Bufferbloat: Dark Buffers in the Internet. *ACM Queue*, 9(11):40–54, Nov. 2011.

[12] D. Hands. Temporal characterization of forgiveness effect. *Electronics Letters*, 37, 2002.

[13] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari. Confused, Rimid, and Unstable: Picking a Video Streaming Rate is Hard. In *Proc. ACM IMC*, pages 225–238, 2012.

[14] 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*, pages 187–198. ACM, 2014.

[15] R. Jain, D.-M. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. 1998.

[16] J. Jiang, V. Sekar, and H. Zhang. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE. In *Proc. ACM CoNEXT*, pages 97–108, 2012.

[17] F. P. Kelly, A. K. Maulloo, and D. K. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, pages 237–252, 1998.

[18] H. Kim and N. Feamster. Improving network management with software defined networking. 51(2):114–119, 2013.

[19] S. Lederer, D. Posch, C. Timmerer, C. Westphal, A. Azgin, S. Liu, c. Mueller, A.Detti, and D. Corujo. ICNRG Internet Draft: Adaptive Video Streaming over ICN draft-irtf-icnrg-videostreaming-03.txt. *https:tools.ietf.orghtmldraft-irtf-icnrg-videostreaming-03*, 2015.

[20] B. Li, Z. Wang, J. Liu, and W. Zhu. Two decades of internet video streaming: A retrospective view. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1s):33, 2013.

[21] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran. Streaming video over HTTP with consistent quality. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pages 248–258. ACM, 2014.

[22] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733, 2014.

[23] L. Liang, G. Feng, and Y. Jia. Game-theoretic hierarchical resource allocation for heterogeneous relay networks. 2013.

[24] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby. User Experience Modeling for DASH Video. In *Packet Video Workshop (PV), 2013 20th International*, pages 1–8. IEEE, 2013.

[25] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang. LayerP2P: Using Layered Video Chunks in P2P Live Streaming. *IEEE Transactions on Multimedia*, 11(7):1340–1352, 2009.

[26] S. H. Low and D. E. Lapsley. Optimization flow control: basic algorithm and convergence. *IEEE/ACM Transactions on Networking (TON)*, 7(6):861–874, 1999.

[27] A. Mansy, B. Ver Steeg, and M. Ammar. SABRE: A Client based Technique for Mitigating the Buffer Bloat Effect of Adaptive Video Flows. In *Proc. 3rd annual ACM Conference on Multimedia Systems*, MMSys '12. ACM, 2012.

[28] P. Marbach. Priority service and max-min fairness. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 266–275. IEEE, 2002.

[29] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Communications Surveys & Tutorials*, to be published. Early Access.

[30] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking (ToN)*, 8(5):556–567, 2000.

[31] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang. QDASH: a QoE-aware DASH system. In *Proceedings of the 3rd Multimedia Systems Conference*, pages 11–22. ACM, 2012.

[32] M. Mu, A. Mauthe, R. Haley, and F. Garcia. Discrete quality assessment in IPTV content distribution networks. *Elsevier Journal of Signal Processing: Image Communication*, 2011. Elsevier.

[33] Q. Ni, R. Zhu, Z. Wu, Y. Sun, L. Zhou, and B. Zhou. Spectrum allocation based on game theory in cognitive radio networks. *Journal of Networks*, 8(3):712–722, 2013.

[34] D. O'Neill, E. Akuiyibo, S. Boyd, and A. J. Goldsmith. Optimizing adaptive modulation in wireless networks via multi-period network utility maximization. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[35] V. Seferidis, M. Ghanbari, and D. Pearson. Forgiveness effect in subjective assessment of packet video. *Electronics Letters*, 28(21):2013–2014, 1992.

[36] W. Tang and R. Jain. Hierarchical auction mechanisms for network resource allocation. *Selected Areas in Communications, IEEE Journal on*, 30(11):2117–2125, 2012.

[37] G. Tian and Y. Liu. Towards agile and smooth video adaptation in dynamic HTTP streaming. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 109–120. ACM, 2012.

[38] C. Timmerer, C. Griwodz, A. C. Begen, T. Stockhammer, and B. Girod. Guest editorial adaptive media streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):681–683, 2014.

[39] C. Timmerer, C. Mueller, and S. Lederer. Adaptive media streaming over emerging protocols. *www-itec.uni-klu.ac.atbibfilesTimmererC012314_revised.pdf*, 2014.

[40] G. Tychogiorgos, A. Gkelias, and K. K. Leung. Utility-proportional fairness in wireless networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 839–844. IEEE, 2012.

[41] W.-H. Wang, M. Palaniswami, and S. H. Low. Application-oriented flow control: fundamentals, algorithms and fairness. *Networking, IEEE/ACM Transactions on*, 14(6):1282–1291, 2006.

[42] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Elsevier Journal Signal Processing: Image Communication*, 19(2), 2004. Elsevier.

[43] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie. A Survey on Software-Defined Networking. *IEEE Communications Surveys & Tutorials*, 17(1):27–51, 2015.

Mu Mu (M'11) is a Lecturer at the Department of Computing and Immersive Technologies, The University of Northampton, Northampton, United Kingdom. Prior to joining the University of Northampton, he was a Senior Researcher in the School of Computing and Communications at Lancaster University where he completed the PhD in Computer Science. His Master's degree of Science was awarded by Darmstadt University of Technology (TU-Darmstadt), Germany. He has taken leading roles in several European and UK research projects with over 40 papers appeared in prestigious conferences and journals. Dr Mu's current research interests include QoE and human factor in multimedia systems, SDN/NFV applications, media orchestration, creative media as well as social and behavioral context integration.

Matthew Broadbent is a Research Associate in the School of Computing and Communications at Lancaster University. His main research interests lie in the use of software-defined networking, and in particular, their ability to aid service delivery and deploy experimental environments. He has been involved in a number of national and international research projects, including OFELIA, GN3plus, Fed4FIRE and more recently, TOUCAN.

Arsham Farshad is a Research Associate within the School of Computing and Communications at Lancaster University, UK. His main research interests lie in monitoring, measurement and management of wireless and data networks. His recent research focuses on video QoE measurement and software-defined virtual infrastructure architecture. He has been involved in a number of national and European research projects, including ARROWHEAD, GN3plus and TOUCAN. He holds a PhD in Informatics from the University of Edinburgh, UK.

Nicholas Hart is a Research Associate at the School of Computing and Communications, Lancaster University, UK. He works on topics of software defined networking as part of the UK EPSRC TOUCAN project. He is a mathematician by qualification, and has worked for many years in development and consultancy roles in the telecommunications industry. Apart from SDN and Cloud Computing, his main interest is in Functional Programming and in particular where FP and Networks intersect.

David Hutchison is Professor of Computing at Lancaster University and founding Director of InfoLab21. He has served on the TPC of top conferences such as ACM SIGCOMM, IEEE Infocom, and served on editorial boards of Springer's Lecture Notes in Computer Science, Computer Networks Journal and IEEE TNSM, as well being editor of the Wiley book series in Computer Networks and Distributed Systems. He has helped build a strong research group in computer networks, which is well known internationally for contributions in a range of areas including Quality of Service architecture and mechanisms, multimedia caching and filtering, multicast engineering, active and programmable networking, content distribution networks, mobile IPv6 systems and applications, communications infrastructures for Grid based systems, testbed activities, and Internet Science. He now focuses largely on resilient and secure networking, with interests in Future Internet and also the protection of critical infrastructures including industrial control systems.

Qiang Ni (M'04-SM'08) is a Professor and the Head of Communication Systems Group at the School of Computing and Communications, Lancaster University, InfoLab21, Lancaster, U.K. Previously, he led the Intelligent Wireless Communication Networking Group at Brunel University London, U.K. He received the B.Sc., M.Sc., and Ph.D. degrees from Huazhong University of Science and Technology, China, all in engineering. His main research interests lie in the area of future generation communications and networking, including Green Communications and Networking, Cognitive Radio Network Systems, 5G, Video Streaming, IoTs and Vehicular Networks in which areas he had already published over 120 papers. He was an IEEE 802.11 Wireless Standard Working Group Voting member and a contributor to the IEEE Wireless Standards.

Nicholas Race is a Reader within the School of Computing and Communications at Lancaster University. He has published over 70 refereed papers in the areas of Media Delivery (including IPTV Systems and Content Distribution), Software-Defined Networks (SDN) and Networking Testbeds (particularly those based on Wireless Mesh technologies). He has been involved in many European projects, and at Lancaster was the principal investigator of OFELIA, GN3plus and Fed4FIRE.