A Scheduling Framework for Adaptive Video Delivery over Cellular Networks

Jiasi Chen *
Princeton University
jiasic@princeton.edu

Rajesh Mahindra, Mohammad A. (Amir) Khojastepour
NEC Laboratories America, Inc.
rajesh, amir@nec-labs.com

Sampath Rangarajan
NEC Laboratories America, Inc.
sampath@nec-labs.com

Mung Chiang
Princeton University
chiangm@princeton.edu

Abstract

As the growth of mobile video traffic outpaces that of cellular network speed, industry is adopting HTTP-based adaptive video streaming technology which enables dynamic adaptation of video bit-rates to match changing network conditions. However, recent measurement studies have observed problems in fairness, stability, and efficiency of resource utilization when multiple adaptive video flows compete for bandwidth on a shared wired link. Through experiments and simulations, we confirm that such undesirable behavior manifests itself in cellular networks as well. To overcome these problems, we design an in-network resource management framework, AVIS*, that schedules HTTP-based adaptive video flows on cellular networks.* AVIS *effectively manages the resources of a cellular base station across adaptive video flows.* AVIS *also provides a framework for mobile operators to achieve a desired balance between optimal resource allocation and user quality of experience.* AVIS *has three key differentiating features: (1) It optimally computes the bit-rate allocation for each user, (2) It includes a scheduler and per-flow shapers to enforce bit-rate stability of each flow and (3) It leverages the resource virtualization technique to separate resource management of adaptive video flows from regular video flows. We implement a prototype system of* AVIS *and evaluate it on both a WiMAX network testbed and a LTE system simulator to show its efficacy and scalability.*

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

Keywords

Cellular Networks, Adaptive Streaming, Proportional fairness, QoE

*This work was done when Jiasi Chen was an intern at NEC Labs America.

MobiCom'13, September 30–October 4, Miami, FL, USA.

http://dx.doi.org/10.1145/2500423.2500433 ...$15.00.

1. INTRODUCTION

As cellular data traffic surges and wireless spectrum becomes scarce, video quality of experience (QoE) degrades. Dynamically adaptive streaming [1] is gaining popularity for streaming video over cellular networks. Adaptive streaming is a technique of video streaming over HTTP where multiple versions of the source video are pre-encoded at different bit-rates on the video server. Adaptive streaming leverages the underlying TCP transport layer to estimate the available capacity for the flow and choose the most appropriate video bit-rate based on the estimated capacity. Hence, this technology attempts to maintain QoE for the user by dynamically adapting the video quality to the changing network conditions. Some recent measurement studies on wired networks [2–5] suggest that multiple adaptive video streaming flows experience performance problems when sharing a bottleneck link. These problems manifest as (a) unfairness in the allocation of bit-rates among the competing flows (b) instability due to unnecessary switching of the video bit-rate, and (c) inefficient utilization of the link. To understand the effect of wireless links on the performance of adaptive video streaming, we perform extensive LTE system simulations and experiments on a WiMAX base station prototype. Our experiments confirm that similar problems manifest themselves on wireless networks.

Although current cellular base stations incorporate sophisticated radio resource management techniques for flow scheduling, the framework lacks mechanisms for operators to effectively allocate resources across users who stream adaptive videos. Schedulers in cellular radio resource management are specifically designed to manage resources for traditional single-rate video streaming flows and elastic data flows; they are not directly applicable for adaptive video streaming flows. Adaptive video flows have unique properties that differ from traditional videos: (a) Adaptive videos are encoded at multiple bit-rate versions and (b) They continuously adapt their bit-rate based on the measured throughput. These characteristics of adaptive video flows places them somewhere between elastic and non-elastic traffic types. Hence, the scheduling framework for adaptive video streaming should take into account these characteristics while allocating resources across such flows.

In this paper, we design and implement AVIS, a resource management framework that addresses the above challenges effectively. Specifically, AVIS is designed to manage the radio resources of a cellular base station across multiple adaptive video flows to meet three goals: (a) optimal allocation as desired by the operator (b) stability of bit-rates allocated to a user and (c) maintain high resource utilization. Firstly, AVIS separates the resource management of adaptive video flows from regular video flows and other data flows using resource slicing techniques [6]. Secondly, AVIS's

scheduler has two components: (a) an *Allocator* that optimally allocates the bit-rates to the different adaptive video flows to ensure fairness and high utilization and (b) an *Enforcer* that schedules the allocated bit-rate to each flow to ensure stability.

In the design of AVIS, we faced several challenges specific to the domain of wireless networks and adaptive video streaming: (1) The goals of the resource allocation are potentially conflicting. Allocating resources to the video flows to achieve fairness may lead to significant bit-rate switches for users, while reducing bit-rate switches may cause unfairness. An effective framework should incorporate appropriate mechanisms for network operators to achieve the desired balance between these goals. (2) The wireless link of a base station is dynamic and its capacity fluctuates depending upon user arrival/departure, locations of users and resource allocation policy. (3) The videos are encoded with multiple discrete bit-rates and for each bit-rate version the instantaneous rate of the video fluctuates significantly around the average bit-rate, complicating the scheduling problem.

AVIS is designed as a gateway level solution with minimum dependence on the specific cellular technology; hence, the design is applicable to multiple 4G wireless access networks such as WiMAX, LTE and LTE-A. To the best of our knowledge, this is the first detailed design, implementation and evaluation of a resource management framework for adaptive video flows on cellular base stations. Overall, we make the following contributions:

- We develop a novel flow management framework that jointly performs optimal scheduling of resources across multiple adaptive video streaming flows and enforces resource isolation across the flows. To avoid degrading the QoE for a user due to frequent bit-rate switching, AVIS enables an operator to effectively maintain a balance between (a) optimal bit-rate allocated to the different users and (b) the average bit-rate switches perceived by the users. Hence, AVIS strives to achieve fair allocation while ensuring good QoE for the users.
- We present the resource allocation as a utility optimization problem with both discrete and continuous optimization formulations.
- We present a detailed design, implementation and evaluation of the system. We evaluate AVIS on both a LTE system simulator and a mobile WiMAX (802.16e) network testbed containing a PicoChip WiMAX ASN (Access Service Network) gateway that runs AVIS, a PicoChip WiMAX base station [7], and Intel-based WiMAX clients [8].
- We design AVIS such that it does not require any client and server modifications, facilitating quicker deployment.

The rest of the paper is organized as follows. Section 2 discusses background on cellular networks and dynamic adaptive video flows followed by the motivation for AVIS. Section 3 presents our design in detail. Section 4 describes our simulation and prototype setup and provides results from our evaluations. We discuss the limitations and future work in Section 5. Section 6 presents related work to place AVIS in context and Section 7 concludes.

2. BACKGROUND AND MOTIVATION

In this section, we first provide a brief background of resource management on cellular networks and HTTP-based adaptive video streaming technology. We then highlight the drawbacks of current resource allocation techniques for adaptive video streaming followed by design considerations for AVIS.

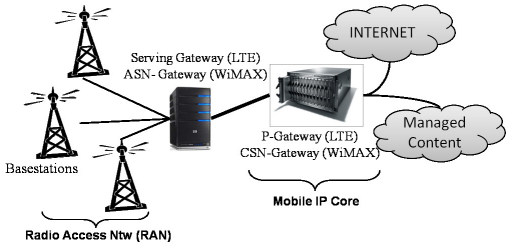


Figure 1: A typical Cellular Network

2.1 Cellular Background

Figure 1 shows a simplified 4G (e.g., LTE or WiMAX) cellular network architecture consisting of two parts: the Mobile IP Core and the Radio Access Network (RAN). For instance, in LTE, the Mobile Core includes the Serving Gateway and the PDN (Packet Data Network) gateway which provide the functionalities of IP connectivity, authentication, authorization and accounting (AAA). The serving gateway typically handles and routes traffic to and from hundreds of base stations. The RAN includes base stations or eNodeBs that perform RRM (Radio Resource Management), interference mitigation, and handover initiation. Base stations incorporate downlink and uplink MAC schedulers which achieve efficient wireless resource allocation across multiple user flows. In both LTE and WiMAX, wireless (radio) resources are OFDMA frames (or subframes) which are divided into "resource blocks or slots" in the time and frequency domain. The task of the downlink and uplink schedulers is to fill these resource blocks or slots with data packets from one or multiple user flows. To support diverse QoS requirements, flows are mapped to one of the following bearer classes: (a) **GBR bearers** which are suited for real-time applications such as VoIP and video. Each flow has an associated minimum GBR (Guaranteed Bit Rate) that defines the minimum allocation the flow desires to receive. Typically in the case of video traffic, the minimum GBR rate is set to the average bit-rate of the video. A maximum GBR rate is also defined to limit the maximum resource allocation for the flow. (b) **Non-GBR bearers** which do not receive any minimum resource allocation. These bearers are used for applications such as web browsing or FTP transfers.

Current base station schedulers allocate resources to the GBR flows by employing variations of the proportional fair (PF) scheduling policy. Such PF schedulers strive to achieve fairness of resources allocated across the users. In addition, the scheduler allocates resources to the users' flows in proportion to their GBR rate. As compared to schedulers that strive to achieve throughput fairness, PF schedulers ensure higher efficiency of the base station resources (i.e., higher base station capacity). Once all GBR flows are satisfied, the unused resources are allocated to the non-GBR flows.

2.2 Adaptive Video Streaming

Several proprietary solutions in the industry such as [9–12] and certain standardized solutions like Dynamic Adaptive Streaming over HTTP (DASH) [1] employ HTTP-based adaptive video streaming. The common key idea is to fragment a video into multiple segments or chunks, where each chunk is encoded using several bit-rates or resolutions. Each chunk, typically containing less than 10 seconds of video content, is stored as a regular file on the web-servers and is downloaded by the clients periodically using the standard HTTP GET requests (see Figure 2). This mechanism ensures that (a) the current web servers and CDNs do not have to be significantly modified to support adaptive streaming. (b) the video

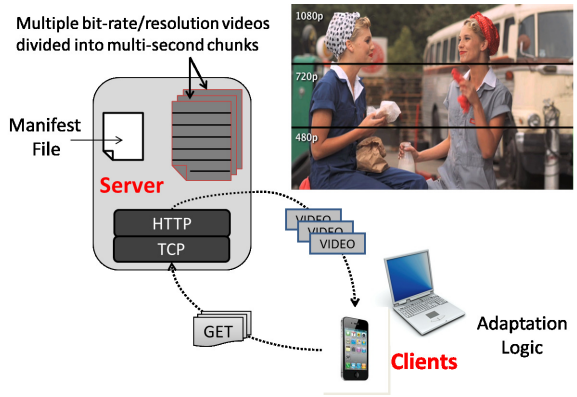


Figure 2: How DASH works

traffic can traverse NATs and firewalls. To enable such a framework, a file describing the list of the chunks of all the video bit-rate versions, including the corresponding HTTP link is downloaded by the client prior to streaming. For instance, the DASH standard (MPEG Dynamic Adaptive Streaming over HTTP [1]) defines a Media Presentation Description (MPD) file, an XML file that contains the HTTP URLs for each video chunk. Typically the client video player implements the adaptation algorithm that chooses the most appropriate bit-rate for the next requested chunk, based on current network and processor or memory conditions. An estimate of the TCP throughput is maintained at the client to predict the future network conditions. While most of the framework is standardized as part of the MPEG based standard DASH, the adaptation algorithm to select the most appropriate bit-rate for future chunks is left to the specific implementation. In the rest of the paper, we refer to dynamically adaptive video flows as DASH video flows.

2.3 Performance Metrics

To help motivate and design AVIS, we define three metrics that are critical for the performance of the network when multiple DASH video flows share the same base station. These metrics are important from the perspective of both the mobile operator and the users.

1. *Fairness*: In wireless systems, users may have different link qualities or transmission rates depending on their distance from the base station and mobility pattern. Hence, base station schedulers perform proportional fair allocation across the users with the aim of achieving fairness in terms of the resources allocated to the users. First, we define the fairness metric for each user i as $F_i = r_i/C_i$ where r_i is the rate allocated to the user and C_i is the transmission rate of the user depending on its Signal-Noise Ration (SNR). Then, based on the above fairness metric, the fairness index is defined based on the Jain Fairness Index [13].

$$JF = (\sum_{i=1}^{N} F_i)^2/(N\sum_{i=1}^{N} F_i^2) \quad (1)$$

where N is the total number of active users in the system. Hence, the above fairness index measures resource fairness across users instead of throughput fairness.

2. *Stability*: From the users' perspective, in addition to the video bit-rate, the QoE of a DASH video stream is greatly impacted by the frequency of bit-rate switches [14, 15]. To measure the stability of the system, we use the frequency of bit-rate switches perceived by the users during a video session. The frequency of bit-rate switches S_i for user i is defined as the total number of bit-rate switches perceived by the user in W units of time. The lower the frequency of bit-rate switches per user, the higher the stability of the system.

3. *Efficiency*: Wireless resources are scarce and operators desire optimal usage of resources of their base stations. Hence we define the resource utilization as the percentage of the ratio of the total resources allocated (R_A) to all the active users to the total resources R_T available at the base station.

$$U = 100R_A/R_T \quad (2)$$

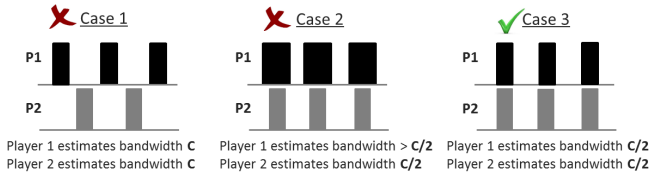
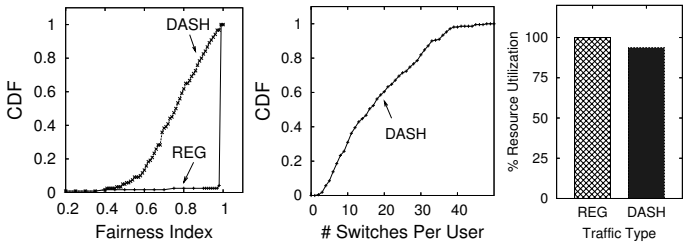


Figure 3: Chunk overlap scenarios.

2.4 Motivation

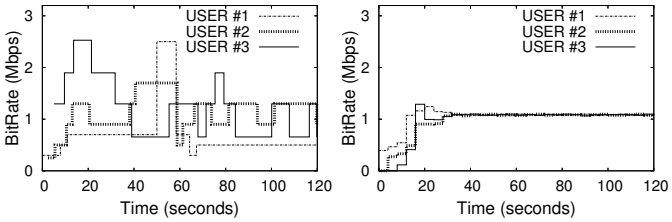
Several studies in the past [2–4] have shown that commercial players perform poorly when multiple DASH flows share a link on the Internet. Specifically, a recent study [5] confirms that most players fail to meet the above mentioned goals: (a) Fairness of resource allocation (b) Stability of bit-rate selection and (c) Efficient resource utilization. The main take-away from these measurement studies is that the fundamental problem with DASH flows is the inaccurate estimation of the network bandwidth. Since the players download chunks periodically and estimate the per-chunk throughput, the temporal overlap of the chunks of different players may cause either under-estimation or over-estimation of the underlying bandwidth (see Figure 3). These studies are performed on wired networks; the dynamic characteristics of the wireless link due to mobility of users, mobile user access patterns and channel quality fluctuations will only adversely impact the interaction between different players.

A fundamental difference between wireless links in cellular networks and wired links on the Internet is the presence of per-flow queues and a resource management scheduler at the base stations. Typical base station schedulers are designed as variants of the PF scheduler that aim to achieve fair resource allocation across the users. The schedulers also provide mobile operators with the ability to control the allocation of resources across users through appropriate configuration. In LTE, mobile operators typically leverage the GBR parameter to ensure that the users receive an allocation proportional to the requirement of their video stream. For instance, if two users stream videos with bit-rates of 1 and 2 Mbps respectively, the operator sets the GBR rates of the two users as 1 and 2 Mbps respectively to ensure that the videos are streamed smoothly. Despite providing effective resource isolation across regular video flows, current schedulers are not suited for resource management of DASH flows for the following reasons: (a) Adaptive videos are encoded at multiple bit-rates, so it is not clear how the GBR rate can be set for such flows. Hence, operators are unable to control the allocation of resources to the users that stream DASH videos. (b) Current schedulers are designed for single bit-rate videos and do not account for the bit-rate switches. (c) Current schedulers optimize resource utilization. Therefore, if certain flows do not have sufficient traffic demand to use their allocated resources, the schedulers assign the resources to other flows. Such a design is not suited for DASH flows that download small-sized chunks periodically (as



(a) Fairness (b) Stability (c) Utilization

Figure 4: State of the Art.



(a) Video (b) FTP (Iperf)

Figure 5: Adobe OSMF over WiMAX.

opposed to downloading an entire file) and estimate the network bandwidth based on the chunk download times.

To illustrate the above mentioned problems, we conduct a large-scale simulation on a 3GPP-compliant in-house LTE system-level simulator. We implemented the adaptation algorithm in the clients that keeps track of the moving average of the TCP throughput and requests chunks of the highest rate that can be supported by the estimated throughput. We set up a network of 21 base stations, with each base station serving 10 mobile users. The users at each base station are randomly placed, so that they have diverse link qualities. We experiment with 2 cases where all users stream (a) regular single-rate videos (REG) with an average bit-rate of 1Mbps (b) DASH videos (DASH) encoded at multiple bit-rates $\{0.1, 0.25, 0.5, 1, 2, 3\}$Mbps. For simplicity, we set the GBR rate of all the users to the same value to ensure that the scheduler strives to achieve a fair resource allocation. We operate each base station at just above saturation, i.e., the resources of each base station are fully utilized. We plot the Fairness Index (as defined in Equation (1)) for every base station for both the cases in Figure 4(a). From the figure, it is clear that the base station scheduler is effective in allocating resources fairly across the users streaming regular videos. However, in the case of DASH flows, the base station scheduler fails to allocate resources fairly across the flows due to the different characteristics of the DASH flows. In addition to the fairness problem, the scheduler also fails to ensure the bit-rate stability of the flows. As seen from Figure 4(b), most of the users witness significant bit-rate switching in the case of DASH flows. Although in some cases the resource utilization of a base station drops when streaming multiple DASH flows, on average the loss is not significant as shown in Figure 4(c).

To further confirm the above results on a real system, we performed an experiment on a WiMAX base station with six WiMAX mobile clients streaming DASH videos using the Adobe OSMF player [16]. As shown in Figure 5(a), the video bit-rates of three out of the six clients do not converge to their fair allocation. Moreover, the bit-rates of the video of each users frequently switch which results in poor QoE. For reference, we plot the allocation with FTP file download for the same setup in Figure 5(b); in this case the base station scheduler achieves a fair allocation. Hence, a base station supporting multiple DASH video flows fails to achieve the goals of Fairness and Stability (defined in Section 2.3), despite the presence of a sophisticated resource management framework on the base station.

3. AVIS DESIGN

AVIS is a resource management framework that enables mobile operators to effectively achieve (a) desired resource allocation (b) stable bit-rates and (c) high resource utilization across multiple DASH video flows. Before diving into the detailed design of the components of AVIS, we explain the considerations leading to the design of AVIS.

(1) *Network vs Client*: Designing AVIS as a network-based solution has the following advantages: (a) It is easy to deploy since client solutions require changing the players of every content or application provider. Content providers may want to use their proprietary adaptation algorithm to differentiate themselves. Moreover, with a client-level solution the operator loses control of resource allocation across the users. (b) The wireless link is extremely dynamic due to user mobility and link quality fluctuations. The adaptation algorithm in clients typically takes several seconds to react and may fail to converge with a distributed algorithm. A centralized in-network solution is more effective in converging to a fair allocation.

(2) *Gateway-level solution*: Although AVIS is designed to manage the resources of each base station independently, we design AVIS as a gateway solution for the following reasons: (a) Typically base stations have minimal computational resources since they need to be deployed at a large number of locations. Implementing AVIS on base stations would require substantial increase in computational and memory requirements of base stations. (b) Network functionality equipment like DPIs (Deep Packet Inspection) that can provide important meta-data information about video flows to AVIS are typically co-located with the Mobile Core gateways.

(3) *Independent scheduler*: One option is to design AVIS as a scheduler that jointly optimizes the resource allocation across the DASH flows, regular video flows and other data traffic. However, in our design we leverage wireless resource slicing techniques [6] to separate the resource management of DASH videos from regular video flows and data flows. Such a design framework has several benefits: (a) It allows operators to set the allocation to the different slices or traffic types based on the long-term resource usage as seen in their networks. The resource slicing technique ensures that any change in one slice due to new users or user mobility does not impact the allocation of resources to other slices. (b) The resource management techniques for DASH video flows and regular video flows can be applied independently and operators can choose their own unique combination. This enables the system to accommodate future innovations.

Drawing motivation from the above mentioned points, AVIS is designed as a gateway-level solution external to base stations (for instance as a plug-in module on serving gateways in LTE networks) as shown in Figure 6. The resource management for different traffic types is performed within individual slices, and the resource slicing technique (including the slice scheduler and the synchronizer, an adaptive downlink shaper) are borrowed from a previous work [6]. AVIS is instantiated as a separate slice as shown in Figure 6 for each base station and schedules the resources allocated to it by the slice scheduler across the DASH flows. AVIS is designed as a split architecture with two novel components: (a) an *Allocator*

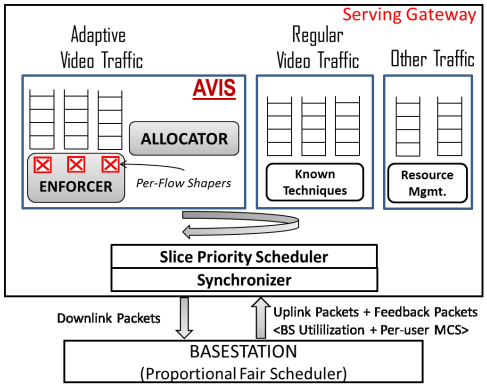


Figure 6: AVIS Architecture.

that allows operators to define the allocation for their users who are streaming DASH videos, much like the GBR bearer framework for regular single-rate video flows and (b) an *Enforcer* designed as a PF scheduler to ensure resource isolation across the DASH flows and enforce the appropriate bit-rate selected by the allocator for each flow. We now explain the design of each component in more detail.

3.1 AVIS's Allocator

The goal of the allocator is to manage the wireless resources of a base station across a set of users streaming DASH flows. Specifically, it takes as input the various available choices of video bit-rates for all the users and converts these bit-rates to their actual radio resource requirements. It then computes the optimal distribution of resources across the users by selecting the appropriate bit-rate for each user. The allocator is invoked periodically every I seconds to ensure adaptability to user arrivals/departures and mobility. In what follows, we first formulate this resource allocation problem in a discrete-optimization framework and then describe the utility function in the context of achieving the two primary goals of fairness and stability. We then formulate the same problem in a continuous-optimization framework to ease computational complexity in the case of a system with large number of users.

3.1.1 Discrete-Optimization Framework

Let T be the total number of resource blocks assigned to the DASH flows of a base station. These resources are required to be distributed among N active users. AVIS assumes that it can obtain bit-rate information about the different encoded versions of the videos of each user i. Such information can be obtained from DPI middleboxes that are part of existing cellular networks [17]. Let M_i be the total number of the available encoded video bit-rate versions for a user i and let r_{ij} denote the bit-rate of version j for user i. Since the users may have different link qualities, let C_i denote the physical transmission rate depending upon the modulation and coding scheme (MCS) used by the base station for user i. This rate represents the maximum bits that can be transmitted to the user per resource block. Note that the base station performs coarse time-scale rate adaptation for each user depending upon the signal-to-noise ratio (SNR) for that particular user. Hence, AVIS obtains the average transmission rate C_i for each user from the base station. We define the utility u_{ij} for each user as a function of the video bit-rate index j, x_{ij} as an indicator variable to represent the bit-rate selected for a user, a penalty function f_{ij} to avoid frequent bit-rate switches, and α as a parameter to trade-off between optimal allocation and stability. We then formulate the resource allocation problem as shown below.

$$\textbf{Problem 1: } \max_{x_{ij}} \sum_{i=1}^{N}\sum_{j=1}^{M_i}(u_{ij} - \alpha f_{ij})x_{ij} \tag{3}$$

$$\text{subject to } \sum_{i=1}^{N}\sum_{j=1}^{M_i}\lceil\frac{r_{ij}}{C_i}\rceil x_{ij} \leq T \tag{4}$$

$$\sum_{j=1}^{M_i} x_{ij} = 1, x_{ij} \in \{0,1\}\ \forall\ i \tag{5}$$

The first constraint Eq (4) ensures that the video bit-rates allocated to the different users do not exceed the physical limit of the resources available at the base station (represented by T resource blocks). To convert the bit-rate of the video flow of a user to actual resource blocks requirements, we scale the bit-rate for each version j of the video of user i by its average transmission rate C_i and bound the value to the next highest integer, i.e., $\lceil\frac{r_{ij}}{C_i}\rceil$. The second constraint Eq (5) ensures that a unique version of each video must be selected for a particular user. Hence, the indicator variable x_{ij} is set to 1 if bit-rate version j is selected for user i, and 0 otherwise.

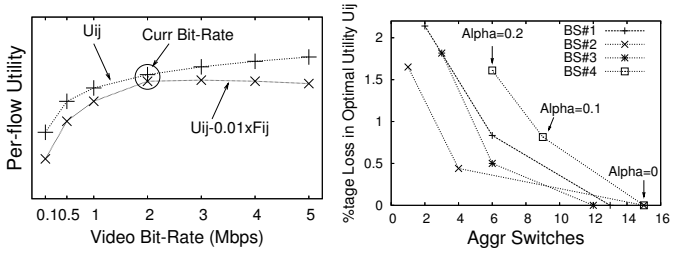
Desired Allocation: : Maximizing the first part of the objective function in Problem 1 (i.e., the aggregate utility u_{ij} across all users) ensures that the resources of the base station are optimally allocated to the different users in accordance with the operators' policy. In our design we define the *bit-rate utility* u_{ij} as follows.

$$u_{ij} = P_i \times \log r_{ij} \tag{6}$$

where P_i is the relative priority of the user. This choice of the utility function has the following benefits: (a) Such a utility implies that the marginal utility of the video for a user decreases as the bit-rate of the video increases. This implies that an upgrade to the next higher bit-rate or a degrade to the next lower bit-rate for a user is visually more perceptible at lower bit-rates. (b) The *log* function is typically employed as a utility function for allocation of resources [18, 19]. The log utility ensures that AVIS strives to achieve a proportional fair allocation across the users. Although the utility u_{ij} only captures the bit-rate r_{ij} allocated to a user, the cost factor (that depends on the users transmission rate C_i) in the constraint stated in Eq (4) ensures that AVIS allocates resources proportional to both the bit-rate of the video and the link quality of the user.

Trade-off between Fairness & Utilization: : Due to the discrete set of bit-rates of DASH videos, allocating resources to achieve optimal fairness may result in loss in resource utilization. Optimal fairness implies that the resources are allocated equally across all users to maximize the fairness index (as defined in Equation (1)). For instance, consider 3 users with similar link quality streaming DASH videos on a base station with a capacity of 4.5 Mbps. The different bit-rates of each of the DASH videos are {0.5,1,2,3} Mbps. While allocating 1 Mbps video bit-rate to each user would be optimally fair, AVIS instead allocates the bit-rates 2, 1 and 1 Mbps to the users respectively to ensure higher resource utilization. On the contrary, a resource allocation solution that ensures optimal resource utilization may be highly unfair to users with relatively bad link qualities. For instance, in the above example an allocation of 2, 2 and 0.5 Mbps to the users respectively would utilize the link entirely. Hence, AVIS strives to achieve a balance between fair resource allocation and optimal utilization of resources.

Trading-off Fairness and Utilization for Stability: : While maximizing the first part of the objective function in Problem 1 (i.e., the aggregate bit-rate utility u_{ij} for all users) ensures (a) that the



(a) Utility Functions (b) Trade-off with α

Figure 7: Design Consideration for AVIS.

users receive the desired allocation according to operators policies and (b) high resource utilization, it may result in frequent bit-rate switching for users. Depending upon the rate of arrival or departure of users and user mobility, the frequency of bit-rate switching across certain users may be relatively high, causing annoyance to those users and degrading their QoE [15]. To address this issue, AVIS defines a penalty function f_{ij} that is subtracted from the utility function u_{ij}. The function f_{ij} ensures that users who experienced higher number of switches during the previous W seconds are not subject to further bit-rate switching. Hence f_{ij} is defined as a monotonically increasing function of S_i where S_i is the number of bit-rate switches that a user has perceived in the previous W units of time. Furthermore, f_{ij} should penalize the different bit-rates of a user depending on the user's current bit-rate. For instance, if a user's bit-rate is changed directly to 2 Mbps from 0.5 Mbps, by-passing the bit-rate version of 1 Mbps, this property ensures that AVIS counts it as two switches in the penalty function. Although there may be several possible choices for the function f_{ij} that satisfy the above requirements, AVIS employs the following function definition based on inference from a simulation result discussed later. We define j^* as the user's current bit-rate.

$$f_{ij} = (|j - j^*| + 1)S_i \tag{7}$$

To understand the objective function in Problem 1 conceptually, Figure 7(a) depicts a graph for both functions, i.e, the bit-rate utility u_{ij} (that does not consider bit-rate switching) and the function $(u_{ij} - \alpha f_{ij})$ that does consider a penalty for bit-rate switching. The functions are plotted for specific values, in this case $\alpha = 0.01$, $S_i = 10$ and $j^* = 2$ Mbps for a video with bit-rate versions = $\{0.1, 0.5, 1, 2, 3, 4, 5\}$ Mbps.

Setting α: : The parameter α allows an operator to balance the goals of (a) optimal bit-rate allocation by optimizing u_{ij} and (b) frequency of bit-rate switching per user (stability). With $\alpha = 0$, AVIS ensures optimal bit-rate assignment (i.e., maximizes u_{ij}) at every execution step and hence strives to achieve proportional fair allocation and high resource utilization. On the contrary, increasing the value of α linearly increases the effect of the penalty function f_{ij}. Hence, setting higher values of α will ensure that AVIS stresses more on bit-rate stability and QoE rather than on optimal allocation, trading-off fairness and resource utilization for QoE. To further illustrate this trade-off, we conduct a simulation for a network of 4 base stations (BS) and plot the loss in the aggregate bit-rate utility value compared to the optimal bit-rate utility value versus the total number of switches for different values of α in Figure 7(b). As can be seen from the graph, increasing α decreases the aggregate number of bit-rate switches for users, while also increasing the loss in the aggregate bit-rate utility. The operator can set the value of α appropriately to balance between optimal allocation and stability of bit-rates.

Algorithm 1 AVIS Allocator

Variables: i: User index, j: Bit-rate index, N: Number of users, τ: Resources
Inputs: Utilities g_{ij}, Video Bit-rates r_{ij}, Transmission rates C_i, Total resource blocks T
Output: Selected bit-rate for each user x_{ij}, $x_{ij} \in [0, 1]$
Repeat: Every I units of time

1: sort $\lceil \frac{r_{i1}}{C_i} \rceil \leq \lceil \frac{r_{i2}}{C_i} \rceil \leq \ldots \leq \lceil \frac{r_{iM_i}}{C_i} \rceil$
2: if $\sum_{i=1}^{N} \lceil \frac{r_{i1}}{C_i} \rceil \geq T$
3: Allocate each user the lowest bit-rate and EXIT.
4: end if
5: for τ from 0 to T do
6: $V(0, \tau) := 0$
7: end for
8: for i from 1 to N do
9: for τ from 0 to T do
10: $V(i, \tau) := -\infty$
11: end for
12: end for
13: for i from 1 to N do
14: for τ from 0 to T do
15: for j from 1 to $NUM(r_{ij})$ do
16: $V(i, \tau) := \max(V(i, \tau), V(i-1, \tau - \lceil \frac{r_{ij}}{C_i} \rceil) + g_{ij})$
17: end for
18: end for
19: end for
20: Output the solution that gives V(N,T)

Allocator Algorithm: : Algorithm 1 summarizes the dynamic program for the AVIS allocator that solves Problem 1. AVIS first ensures that at least the base version of the video can be selected for each user. To achieve this feasibility check, the allocator sorts the versions of each video for all the users such that $\lceil \frac{r_{i1}}{C_i} \rceil \leq \lceil \frac{r_{i2}}{C_i} \rceil \leq \ldots \leq \lceil \frac{r_{iM_i}}{C_i} \rceil$. The solution is feasible if at least the base versions can be supported, i.e., $\sum_{i=1}^{N} \lceil \frac{r_{i1}}{C_i} \rceil \leq T$. In case this condition is not satisfied, the allocator simply selects the base version for each user (See steps 1-4).

Problem 1 falls in the category of Multi-choice Knapsack problem [20]. Each user i is analogous to a *class* and the video bit-rate versions j of each user are the *items* within each class. The constraint given by equation (5) in Problem 1 says that the allocator must pick exactly one item (bit-rate) from every class (user). Each item has a utility $g_{ij} = (u_{ij} - \alpha f_{ij})$ as defined in the objective function of Problem 1. The *cost* of each item is the resources required to transmit that bit-rate version defined as $\tau_{ij} = \lceil \frac{r_{ij}}{C_i} \rceil$. Let $V(i, \tau)$ denote the optimal utility of users 1 through i given τ resource blocks. The initialization is done in two steps as shown in steps 5-12. The matrix V is built iteratively for all the users, and for each user the problem is solved for all the capacities $0..T$ (i.e., resource blocks). The optimal utility $V(i, \tau)$ depends on which version j is chosen for user i. So for each version j, the allocator checks the optimal utility obtained by $i - 1$ users given $(\tau - \tau_{ij})$ resource blocks, and adds the utility of the item j for user i: g_{ij}. To obtain optimal utility $V(i, \tau)$, it picks the bit-rate version j that has the highest utility (See steps 13-19). Finally, the optimal solution is obtained by back-tracking to find the combination of bit-rate versions for each user i that yielded $V(N, T)$.

3.1.2 Continuous Optimization Framework

In the above, we modeled the resource allocation problem as a discrete-optimization based knapsack problem. The dynamic program has a complexity of $\mathcal{O}(NMT)$ where N is the number of

users, $M = \max_i M_i$ is the maximum number of versions of the video of a user, and T is the total number of resource blocks. The problem is pseudo-polynomial since the complexity is proportional to the value of T. Hence for large values of T, the computational complexity grows significantly with large number of users N and video bit-rate versions M. To get an idea of the possible values for T, in LTE the total number of resource blocks to be scheduled every second is 24000. Although the actual complexity is lower since each user is allocated its minimum bit-rate video version, in order to ease the computational complexity of the system with large number of users N and video versions M, one alternative approach is to formulate Problem 1 as a continuous optimization problem. Specifically, we define the optimization variable r_i that represents the rate allocated to each user as a continuous variable.

$$\textbf{Problem 2}: \max_{r_i} \sum_{i=1}^{N} (u_i - \alpha f_i) \tag{8}$$

$$\text{subject to } \sum_{i=1}^{N} \frac{r_i}{C_i} \leq T \tag{9}$$

$$r_{i1} < r_i \leq r_{iM_i}, \ \forall \ i \tag{10}$$

The continuous problem is defined with similar goals as those for the discrete problem (Problem 1). The first constraint Eq (9) ensures that the resource constraint of the base station is met while the second constraint Eq (10) ensures that the rate allocated to the user is at least the lowest bit-rate version of the video and is also within the maximum bit-rate version for the video. The utility for each user is defined as a log function of the allocated rate.

$$u_i = P_i \times \log r_i \tag{11}$$

The penalty function to control the frequency of bit-rate switching per user is defined similar to the discrete case.

$$f_i = (\sqrt{(r_i - r_i^*)^2} + 1) S_i \tag{12}$$

where r_i^* is the current bit-rate streamed to user i.

Once the solution to Problem 2 is solved using well known techniques like the interior point method, the continuous optimal variables r_i need to be quantized to a discrete value r_{ij} among the available choices for bit-rates of user i. For instance, if the available video bit-rates of a DASH flow are $\{1, 2, 3\}$ Mbps and the optimal solution of the continuous resource allocation problem is $r_i = 1.45$ Mbps, then r_i has to be quantized to either 1 or 2 Mbps to obtain r_{ij}. Although it may be possible to design an optimal quantization algorithm to return a solution that matches the solution obtained by discrete optimization (Problem 1), we found that such techniques result in similar computational complexity as the discrete optimization problem. Hence, we simplify our approach to ensure that we achieve lighter computational complexity, which trades-off with the optimality of the resource allocation. The solution is based on a greedy approach aimed at minimizing the loss in utilization of the wireless resources and contains the following steps:

1. Sort the users in increasing order of wireless resources required to upgrade their bit-rate r_i to the next higher bit-rate version.
2. Round the solution r_i to the next lowest bit-rate version for all users i.
3. Compute the amount of wireless resources that are unused after satisfying all the users with the rounded version of their optimal bit-rates (Step 2).
4. Scan through the sorted list to upgrade the maximum number of users until the resources are exhausted.

Clearly, this approach trades-off solution optimality in order to achieve a reduction in the computational complexity. As we experimentally show later, the above scheme achieves results similar to the discrete optimization solution.

The above algorithm achieves a complexity of $\mathcal{O}(N)$, although the step-wise computation time for solving the continuous problem (Problem 2) may be larger than the step-wise execution time for the dynamic program that solves the discrete optimization problem. AVIS can be deployed with the continuous optimization framework in the case of a system with large number of users N and high number of video bit-rate versions M for two reasons: (a) The benefit of reduction in computational complexity is higher for larger values of N and M for the case with the continuous optimization framework as opposed to the discrete optimization framework. (b) The error in the optimal solution obtained with the continuous optimization framework will be lower in general when M is large. In our prototype, AVIS employs the discrete optimization as defined in Problem 1 by default.

3.2 AVIS's Enforcer

Once the allocator computes the appropriate bit-rate version r_{ij} for each user i, it feeds this information to the enforcer. The AVIS enforcer is designed similar to the PF scheduler that is typically employed on base stations, including the ability to perform per-flow traffic shaping. The enforcer is configured to meet three key requirements: (a) It ensures isolation of resources across the DASH flows. Hence if the link quality of a user improves or degrades, the resource allocation for the other DASH flows is not affected. (b) It ensures stability by making sure that the bit-rates allocated by the allocator are enforced for every user and (c) It ensures high resource utilization.

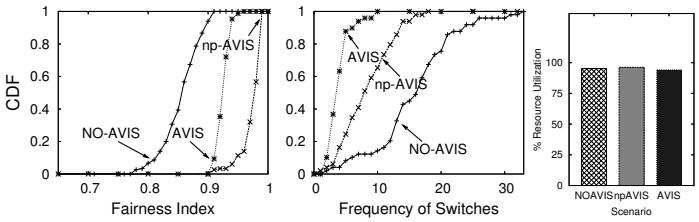
The enforcer is designed as a weight-based packet scheduler that schedules the packets of the different DASH flows proportional to the flow's minimum rate. To ensure that each flow receives sufficient TCP throughput to support the bit-rate chosen by the allocator, the enforcer sets the minimum rate for each flow equal to the allocated bit-rate r_{ij}.

$$MinRate_i = r_{ij} \tag{13}$$

In addition, the enforcer considers the transmission rate of the users when scheduling the packets. This ensures that the drop of a user's transmission rate due to mobility does not affect the resource allocation for the flows of other users.

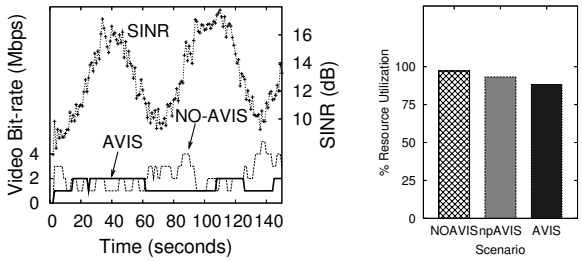
To meet the goal of stability, the enforcer employs per-flow traffic shapers. The shapers cap the rate available to a flow despite the presence of unused resources. The maximum rate for each flow is set based on the allocated bit-rates r_{ij}. This ensures that the TCP throughput of a flow will not exceed the allocated bit-rate of that flow. Despite availability of unused resources, the flow will not switch to a higher bit-rate, thus ensuring stability. Such a design ensures that AVIS can enforce the computed bit-rates for the users indirectly without any video server or client modifications.

To meet the final goal of high utilization, the shaper rate for each flow is set to the mean of the allocated bit-rate and the next higher bit-rate of the video for that flow. This design choice was based on an observation that most videos have VBR traffic patterns and their instantaneous rates can fluctuate around their average bit-rates. If the instantaneous requirement for a certain flow is above its average bit-rate, the enforcer can borrow resources at fine time-scales from a flow whose instantaneous rate is below its average bit-rate. Thus, setting the maximum rate for the flows above their average



(a) Fairness (b) Stability (c) Utilization

Figure 8: Efficacy of AVIS.



(a) Stability (b) Utilization

Figure 9: User mobility.

bit-rate ensures that the enforcer can leverage statistical multiplexing across the video flows to obtain high resource utilization. While the shaping rate could be set to higher values, AVIS sets it in the following way to ensure that the client players do not switch to higher bit-rates than those selected by the allocator.

$$MaxRate_i = \frac{r_{ij} + r_{ij+1}}{2} \tag{14}$$

While the maximum shaping rate is critical to the operation of the enforcer, the interval at which the shaping is performed is equally important. Hence while the allocator operates at coarse time-scales, the enforcer schedules the flows at much finer time-scales. For instance, in our prototype and simulations, the allocator executes every 10 seconds, while the enforcer schedules at 10 millisecond granularity with traffic shaping performed every 250 milliseconds.

4. EVALUATION

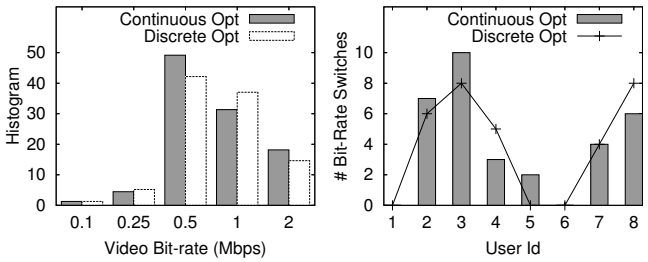
In this section, we evaluate AVIS extensively through simulations and a prototype implementation. In both cases, the code is written primarily in C/C++ and is around 1000 lines of code for the entire framework. We first show results from simulations, followed by the prototype description and experimental evaluations. The setup of each experiment is explained while describing the experiment. In both simulations and experiments, we set $\alpha = 0.1$ (see Problem 1). AVIS keeps track of the number of bit-rate switches for each user in the past W seconds, and we set $W = 30$ seconds.

4.1 LTE System Simulation

We first study the efficacy of AVIS using large-scale simulations. The simulations are performed with a 3GPP-compliant LTE system-level simulator in which we implemented the allocator and the enforcer components of AVIS. The simulator is built using C++ and is capable of simulating a network of up to 57 LTE base station cells. It supports the 3GPP LTE Release 8 [21] features for both Physical layer and MAC layer stacks. The simulator supports sophisticated wireless channel models, including fast fading and user mobility, and also includes the MAC layer flow scheduling framework. At higher layers of the stack, it supports the TCP transport protocol with provisions for HTTP, FTP, and RTP-based Video and VoIP traffic for each user. We implemented DASH video streaming on top of the HTTP protocol as well as the adaptation algorithm in the clients. The adaptation algorithm keeps track of the moving average of the TCP throughput and requests chunks of the highest rate that can be supported by the estimated throughput. We implemented the AVIS functionality, including both the allocator and the enforcer, within the simulator and instantiated an instance of AVIS for each base station.

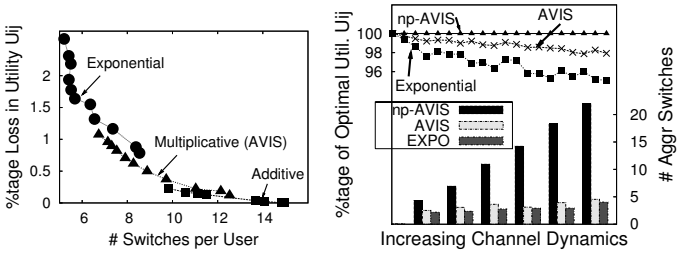
Efficacy of AVIS: In this simulation, we compare AVIS to a vanilla system with no AVIS. We set up a network of 21 base stations, with each base station serving 8 users. The users are randomly distributed at each base station and all the users stream DASH videos of length 5 minutes with multiple bit-rate versions: $\{0.1, 0.25, 0.5, 1, 2, 3\}$ Mbps. We compare the three performance metrics defined in Section 2.3 for the system with and without AVIS. For the case with NO-AVIS, we set the GBR rate of each flow to the same value. For AVIS, we set the same utility values for each flow according to Equation (6). This configuration ensures that the fairness of the resource allocation is comparable between the case with AVIS and the case without AVIS. We plot the fairness index (as defined by Equation (1)) for all the base stations in the network for both the cases. As seen from Figure 8(a), AVIS ensures higher fairness among the competing DASH flows than the case with NO-AVIS, despite fluctuations in the link conditions of the users. With AVIS, the fairness index is more than 0.9 for most of the video session time for all the users across all the base stations, while with NO-AVIS, the fairness index is below 0.85 for 50% of the total video session time of the users. We also plot the fairness index for the case where AVIS does not consider a penalty function for bit-rate switching, denoted by np-AVIS. As seen from the Figure 8(a), np-AVIS achieves much higher fairness as compared to AVIS since it optimizes for the aggregate bit-rate utility u_{ij} across all users (Problem 1 with $\alpha = 0$) at every execution interval. However as shown in Figure 8(b), np-AVIS results in users experiencing a higher frequency of bit-rate switching. AVIS, on the other hand, employs the switch penalty function as defined by Equation (7) ensuring that users perceive more stable bit-rates, and therefore improving their QoE. In either case, AVIS achieves lower frequency of bit-rate switching among the users than the case with NO-AVIS as depicted in Figure 8(b). Finally as shown in Figure 8(c), AVIS achieves its goals while maintaining high utilization of the base station resources.

User Mobility: In this simulation, we show the efficacy of AVIS in presence of high user mobility. We set up a base station with 8 active users including several vehicular users. All the users stream DASH videos of length 5 minutes with multiple bit-rate versions: {0.1, 0.25, 0.5, 1, 2, 3} Mbps. To show the variation in the link quality of the vehicular clients, we plot the signal to interference-plus-noise ratio (SINR) of one of the vehicular users in Figure 9(a). We also plot the bit-rates as perceived by this user with AVIS and NO-AVIS. As seen from Figure 9(a), AVIS achieves higher stability since the user perceives lower frequency of bit-rate switches than the case with NO-AVIS. The higher stability is achieved since AVIS employs the penalty function for recent bit-rate switches perceived by the user. The stable bit-rates also shows the efficacy of the enforcer in ensuring bit-rate allocations in presence of channel fluctuations. However as seen in Figure 9(b), AVIS compromises on resource utilization to ensure higher stability for the user. In this case, the resource utilization achieved by AVIS is about 88% as compared to 97% achieved by NO-AVIS. np-AVIS achieves higher



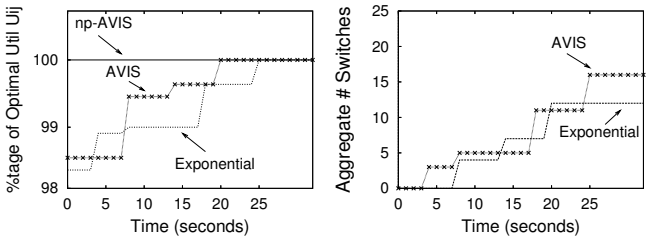
(a) Bit-Rate (b) Switches

Figure 10: Comparison of Optimization Frameworks.



(a) Tradeoff with diff F_{ij} (b) Effect of Capacity changes

Figure 11: Penalty Functions.



(a) Aggr. Utility (b) Aggr. Switches

Figure 12: Convergence under Stable Conditions.

resource utilization of 94% at the cost of lower stability (the bit-rate switches are not shown for np-AVIS for the sake of clarity).

Continuous Optimization Framework: To compare AVIS that employs the continuous optimization framework (as described in Section 3.1.2) with AVIS that employs the discrete optimization framework, we repeat the first simulation with AVIS employing the continuous optimization framework. In Figure 10(a), we compare the distribution of the video bit-rates allocated to all the users using both the optimization frameworks. We also plot the average number of bit-rate switches perceived by a random set of 8 users in Figure 10(b). Clearly, AVIS with continuous optimization framework is effective in emulating AVIS that employs the discrete optimization framework. We employed the non-linear optimization library OPT++ [22] to solve the continuous resource allocation problem.

Choice of Penalty Function: The penalty function f_{ij} employed in the objective in the resource allocation problem (Problem 1) ensures that the operator can balance between optimal bit-rate allocation (which is obtained by computing the optimal value of aggregate bit-rate utility u_{ij} across all users in Problem 1) and stability of bit-rates across its users. The choice of the penalty function f_{ij} is an important design consideration for AVIS, as it directly impacts the effect of a user's recent bit-rate switch on the user's future bit-rate allocation. Hence, we benchmark the tradeoff between the optimal allocation and stability for the following penalty functions that we considered during our design. The functions are called additive, multiplicative and exponential respectively.

$$f_{ij}^{(1)} = |j - j^*| + 1 + S_i \tag{15}$$

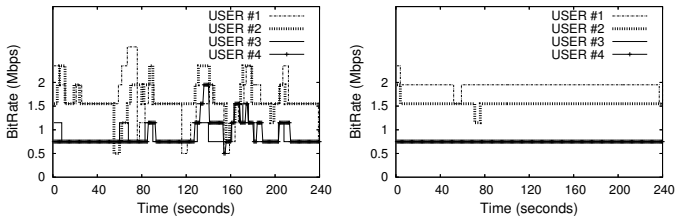
$$f_{ij}^{(2)} = (|j - j^*| + 1)S_i \tag{16}$$

$$f_{ij}^{(3)} = 2^{|j-j^*|+1+S_i} \tag{17}$$

Note that if $S_i = 0$, we set $f_{ij} = 0$. We simulate a setup with several base stations for different runs of AVIS, using the above penalty functions for a range of values for $0 \leq \alpha \leq 1$ (as defined in Problem 1). Figure 11(a) shows the trade-off curves between the loss in the optimal bit-rate utility and the total number of bit-rate switches per user. Clearly, the additive function is more or less insensitive to the number of bit-rate switches and fails to lower the number of bit-rate switches even for high values of α. On the other hand, the exponential function is too aggressive to recent bit-rate switches and is highly sensitive to α. Hence, AVIS employs the multiplicative function since its trade-off curve lies at a sweet spot and allows the operator to further balance between the two metrics by setting α appropriately.

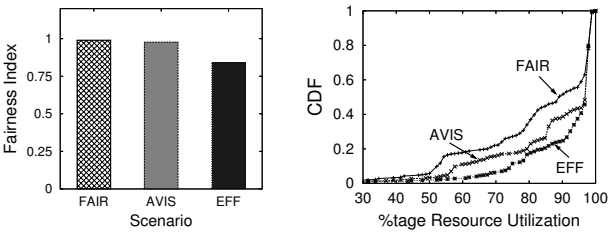
In the same experiment, we also plot the effect of increasing channel dynamics on the loss in optimal bit-rate utility and number of bit-rate switches for the users. The channel dynamics are increased by increasing the frequency of user arrivals and the mobility speed of the users. Clearly, from Figure 11(b), the loss in the bit-rate utility obtained by AVIS diverges from the optimal utility obtained by np-AVIS with increasing channel dynamics. However, AVIS ensures that the number of bit-rate switches for the users is lower than that with np-AVIS. This is because np-AVIS optimizes for the bit-rate utility u_{ij} at every execution step and changes the allocated video bit-rate of the users when the effective capacity changes. On the other hand, AVIS considers the past bit-rate switches for the user before making a decision on changing its bit-rate in response to a change in the effective capacity. We also plot the result of AVIS with exponential penalty function (Equation 17). As expected, the exponential penalty function causes a further drop in the optimal allocation as compared to AVIS. However, the reduction in the number of bit-rate switches is not significant when compared to AVIS. This result further justifies our choice of the multiplicative penalty function in AVIS.

Convergence Under Stable Channel Conditions: As seen in the previous simulation, AVIS may not achieve the optimal bit-rate allocation as achieved by np-AVIS since AVIS employs a penalty function f_{ij} for bit-rate switching. However, an interesting question to ask is whether the solution of AVIS eventually converges to that obtained by np-AVIS if the conditions of the system remain stable. Stable conditions imply that there is no new user arrival or departure and the link qualities of the users remain stable. We conduct a simulation such that the system is unstable for a certain period of time and is then kept stable. We plot the aggregate utilities obtained by AVIS, np-AVIS and AVIS with an exponential penalty function in Figure 12(a). As expected, np-AVIS converges to the optimal bit-rate allocation immediately while AVIS takes around 20 iterations. AVIS with exponential penalty function takes more iterations than AVIS to converge; however, it causes fewer bit-rate switches as shown in Figure 12(b). Although in a practical system the conditions typically do not stay stable for too long, this result confirms that the solution obtained by AVIS moves in the right direction towards the optimal bit-rate allocation even while minimizing the switches in bit-rates of the users.



(a) No-AVIS (b) AVIS

Figure 13: Efficacy of AVIS.



(a) Fair Allocation (b) Utilization

Figure 14: Tradeoff between Fairness and Utilization

4.2 WiMAX Prototype Experimentation

We have developed a prototype system to validate the proposed solution on a WiMAX testbed. The testbed consists of an Access Service Network (ASN) gateway, three PicoChip [7] WiMAX femtocell base stations (IEEE 802.16e compliant), and several Intel WiMAX [8] clients (See Figure 15). The ASN gateway provides an interface to the base station for setting up service flows in the downlink and uplink directions for each client when it registers. We implement AVIS as a user-level Click module [23] in the ASN-gateway. Click intercepts all data packets from the base station in the downlink. We configure the Click classifier to route packets belonging to the DASH flows through the AVIS module. The PicoChip base station provides feedback on the average MCS per client to the AVIS scheduler every I units of time. When this feedback is received, the AVIS allocator selects the video bit-rate version j for each user i and sets the shaper rate appropriately for each flow. AVIS maintains a separate queue for each flow and performs weighted packet scheduling and rate-shaping for each flow depending on the bit-rates selected by the allocator. On the client-side, we use Adobe OSMF [16] player that runs as a browser-plugin. We modified the player to record certain parameters such as chunk throughput, bit-rate, etc., at the clients. The videos are fetched over the Internet from an Akamai- hosted CDN server [10]. All the videos are about 4 minutes in length and are encoded to the following bit-rate versions: {0.5,0.75,1.15,1.55,1.95,2.35,2.75,3.25} Mbps.

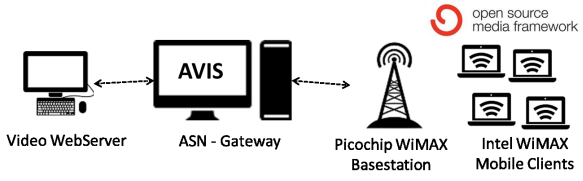
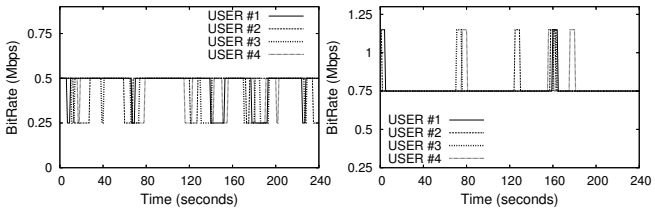
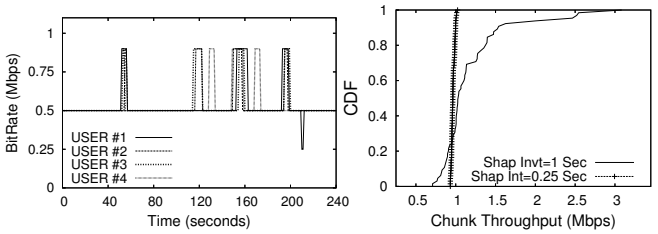


Figure 15: AVIS WiMAX prototype (*Attribute:The Noun Project*).



(a) Low Shaping Rate (b) High Shaping Rate

Figure 16: Per-flow Shaping Rate



(a) Shaping Interval = 1sec (b) Estimated Throughput

Figure 17: Effect of Shaping Interval.

Efficacy of AVIS: We conduct an end-end system evaluation of AVIS and compare it with a baseline WiMAX system without AVIS. In this scenario, we set up the base station with 4 clients placed randomly in an indoor environment. The transmission rate (or MCS) of the clients is such that Users $1, 2$ receive packets at 16QAM while Users $3, 4$ receive their packets at QPSK. In Figure 13, we show a time-series plot of the bit-rates of the videos that are streamed to the users for both AVIS and no-AVIS. AVIS allocates bit-rates of 1.95 Mbps, 1.55 Mbps to Users 1 and 2 respectively and 0.75 Mbps each to Users $3, 4$; while with no-AVIS, the bit-rates of the users do not converge to any specific bit-rate.

In this experiment, we also demonstrate the efficacy of AVIS in effectively balancing the goals of fairness and optimal resource utilization. We repeat the above experiment for two other cases, one that achieves optimally fair allocation (named as FAIR) and another scheme that achieves optimal resource utilization (named as EFF). We plot the fairness index (as defined in Equation (1)) in Figure 14(a) and the CDF of resource utilization (as defined in Equation (2)) in Figure 14(b) for all 3 scenarios. Clearly, AVIS is effective in achieving a balance between optimally fair allocation and optimal resource utilization. Note that while the FAIR scheme allocated bit-rates of 1.55 Mbps and 0.75 Mbps to Users $1, 2$ and Users $3, 4$ respectively, the EFF scheme allocated bit-rates of 2.75 Mbps and 1.95 Mbps to Users 1 and 2 respectively and 0.5 Mbps each to Users $3, 4$.

Benchmarking the Enforcer: Although the enforcer leverages the PF scheduler to ensure bit-rate allocation to each user, configuring the per-flow shapers is critical to its performance. We evaluate two metrics that we observed that affect the efficacy of the enforcer, in terms of bit-rate stability for the users. We set up the WiMAX base station with 4 clients placed at similar locations so that their link qualities are similar (QPSK).

(a) Shaper Rate: AVIS employs a shaper to ensure a maximum rate for each flow depending upon its assigned bit-rate (Equation 14). While it seems intuitive to set the shaping rate to the bit-rate assigned to a user, such a setting causes the player to switch to a lower bit-rate, mainly due to buffer underflow. In our experiment, the three flows are allocated a bit-rate of 0.5 Mbps each by the al-

locator and the enforcer sets their shaping rates to 0.5 Mbps. As shown in Figure 16(a), the bit-rates for all the videos switch to the lower bit-rate of 0.25 Mbps frequently. While it is possible to set the shaping rate to higher values to increase stability and utilization, setting it too high causes the player to upgrade to the next higher bit-rate version. For instance, setting the shaper rate to the next higher bit-rate than the one allocated to the user causes the player to switch between the assigned bit-rate and the next higher bit-rate version. In our experiment, we set the shaping rate of the four users equal to 1.15 Mbps once the allocator assigns the bit-rate 0.75 Mbps to each user. As shown in Figure 16(b), the users occasionally switch to a higher bit-rate. Hence taking the middle-ground, we set the shaping rate as the mean of the assigned bit-rate and the next higher bit-rate as given in Equation 14.

(b) Shaping Interval: The shaping interval determines the time interval or granularity at which the shaping for each flow is performed. In our initial design, we set the shaping interval to 1 sec, based on the default value used by most commercial base station schedulers. In this experiment, all four users are assigned a bit-rate of 0.5 Mbps and the shaping rate is set to 0.65 Mbps with a shaping interval of 1 second. Clearly, as shown in Figure 17(a), this setting sometimes causes the players to switch to a higher bit-rate version. We observed that some video chunks (especially during static scenes) have relatively smaller size and may be downloaded within a second. A series of such chunks causes the player to overestimate the throughput as shown in Figure 17(b). Hence, we configure AVIS with a shaping interval of 250 milliseconds.

5. DISCUSSION AND LIMITATIONS

AVIS is designed as a non work-conserving scheduler, which may result in resource under-utilization at certain times. However, by ensuring that the AVIS allocator is executed at short time-scales (every I units of time), the operator can ensure that AVIS is adaptive to the dynamic capacity changes at a base station. However, reducing the execution time I increases the amount of feedback from the base stations to the gateway. While the overhead of feedback is insignificant compared to the data traffic, the operator needs to consider this trade-off during deployment. Another point to note is that AVIS is built on a resource virtualization platform that incorporates a work-conserving slice scheduler. In the worst case that AVIS does not utilize its share of resources during certain execution periods, the slice scheduler would redistribute the unused resources to other traffic classes that have sufficient traffic. Hence, the overall base station resource utilization would not suffer.

Although we show the efficacy of AVIS in large-scale using simulations, the evaluation testbed contains a small number of clients primarily since the base station available to us is a femtocell base station that supports a maximum of 8 clients. Nevertheless, AVIS should scale to larger settings in real cellular networks, since the flow management in AVIS is done on a per-base station instance and each base station only handles a small number of flows. Since AVIS is designed as a gateway solution, the processing power for AVIS instances can be shared across base stations for better multiplexing.

To ensure stability of bit-rate allocation to the users, AVIS employs a penalty function f_{ij} that is a function of S_i (see Equation 7). The metric S_i is simplistic in the paper as it only captures the total number of bit-rate switches that a user has perceived in the previous W units of time. However, it can be extended to more sophisticated formulations: For instance S_i could be defined similar to the *instability* metric in [5] that gives more weight to recent bit-rate switches and scales the number of switches by the average bit-rate achieved by the user.

There may be a few other scenarios that occur in practice, such as the presence of misbehaving clients that do not perform rate adaption according to the underlying TCP throughput. Also, for certain clients, the wired network or the computational resources at the client may be the bottleneck, forcing the client to use a bit-rate lower than that allocated by AVIS. Although AVIS is not explicitly designed to handle such scenarios, the network can either measure the TCP throughput of a flow or parse the uplink HTTP GET requests to detect clients that are not requesting the bit-rate allocated to them by AVIS. An interesting avenue for future work is to extend the allocator in AVIS to account for such cases and redistribute the resources unused by such flows to other active flows. However, the allocator must ensure that it allocates the appropriate resources to a flow once its conditions improve, such as if, for instance, the wired bandwidth increases after a short period of congestion or computational resources at the client become available.

6. RELATED WORK

A large number of efforts have focused on optimizing video delivery (both in the wired and wireless network domains). We cover the most relevant categories.

Client-side adaptation: One approach to achieve fairness and stability across multiple adaptive video flows is to employ more efficient adaptation logic like FESTIVE [5] in the video players. However, such techniques require significant changes to the video players which may not be feasible, hindering their deployment. Moreover, it is harder for players to converge to a fair bit-rate allocation efficiently in a distributed fashion, as opposed to an in-network solution like AVIS that possesses knowledge of the base station capacity. AVIS also ensures that network operators have control over the resource allocation across all their users that stream DASH videos.

Base Station Scheduling: There is a plethora of work, e.g., [24–28], on radio resource management for wireless networks specifically targeted at optimizing video delivery. However, these schemes are designed for single-rate video streams and are not directly applicable for DASH video flows since they do not consider client-side adaptation of the incoming video streams. In fact, such schemes are complementary to AVIS and can be employed to optimize the video delivery under heavy-load conditions where the base video bit-rate version cannot be supported for all the users. These schemes essentially rely on techniques such as intelligent frame dropping or distortion based scheduling to ensure that users receive good QoE in presence of base station congestion. Other schemes assume knowledge of the probability distribution of the wireless link conditions, and find the optimal average rate allocation region. AVIS, on the other hand, is designed as a practical solution that is not based on stochastic channel models.

In-network Optimizers: A recent work [29] proposes the design of an in-network per-flow proxy that resides in between the server and the client in the Internet. It performs in-line measurements to estimate the end-to-end bandwidth and guides the client player in performing bit-rate switching. AVIS, on the other hand, is an in-network optimization framework designed to perform multi-flow scheduling for DASH flows streamed over cellular networks. There are also a few approaches that involve modifying the video stream at an intermediate node in order to adapt to link capacity variations [30, 31]. However, transcoding requires significant amount of computational resources and signaling to inform the receiver of the change in coding. These techniques are less effective for DASH video flows where the original server performs adaptive bit-rate streaming anyway.

7. CONCLUSION

In the past, numerous research efforts have focused on efficient resource management schemes, especially on wireless access links for video streaming over unreliable transport protocols like UDP. However, industrial solutions have largely adopted TCP/HTTP as the de facto for video streaming. More recently, adaptive video streaming over HTTP (DASH) is gaining widespread popularity in commercial players to ensure smooth video streaming. Hence, there is a need to revisit resource management techniques to ensure efficient video streaming over wireless links.

In this paper, we presented the detailed design and implementation of a flow management framework for adaptive video delivery over cellular networks, namely AVIS. We demonstrated the efficacy of AVIS using both a LTE system simulator and experiments on a WiMAX prototype implementation. AVIS is effective in allocating the resources of a base station across multiple adaptive video flows and effectively balances between three important goals: (a) Fair allocation (b) Stability of a user's bit-rate and (c) Efficient resource utilization of the base station. Since the three goals are self-conflicting and quality of a video stream is highly subjective, AVIS is designed as a general framework which incorporates appropriate knobs for the mobile operators to achieve desired resource allocation across their users. To ease deployment, AVIS is designed to control the video bit-rate of each user without modifications to the video server or the client player.

Acknowledgments

This paper benefited significantly from MobiCom 2013 reviewer comments, for which we are very thankful. We thank our shepherd, Songwu Lu, for overseeing the editing process. We would also like to thank Honghai Zhang for the initial discussion on this line of work. Jiasi Chen and Mung Chiang were supported in part by AFOSR MURI grant FA9550-09-1-0644 and by NSF NeTS grant CNS-1011962.

8. REFERENCES

[1] MPEG DASH standard. http://dashif.org/mpeg-dash.

[2] S. Akhshabi et al. An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP. In *MMSys*, 2011.

[3] S. Akhshabi et al. What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth? In *NOSSDAV*, 2012.

[4] C. Muller et al. An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments. In *MoVid*, 2012.

[5] J. Jiang et al. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming. In *CoNEXT*, 2012.

[6] R. Kokku et al. CellSlice: Cellular Wireless Resource Slicing for Active RAN Sharing. In *COMSNETS*, 2013.

[7] PicoChip Femtocell Solutions. http://www.picochip.com/.

[8] Intel WiMAX. http://tinyurl.com/73dp8wz.

[9] Adobe HTTP Dynamic Streaming. http://tinyurl.com/aqnv7y5.

[10] Akamai HD Streaming. http://tinyurl.com/yaknp7a.

[11] Microsoft Smooth Streaming. http://tinyurl.com/cyqldms.

[12] Apple http live streaming. http://tinyurl.com/cwkv9kz.

[13] R.K. Jain et al. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. In *Technical Report DEC*, 1984.

[14] N. Cranley et al. User perception of adapting video quality. In *Int. Journal of Human-Computer Studies*, 2006.

[15] B.Shenet et al. A Quest for an Internet Video Quality-of-Experience Metric. In *HotNets*, 2012.

[16] Adobe OSMF Player. http://www.osmf.org.

[17] Sandvine. http://www.sandvine.com.

[18] F. Kelly. Charging and rate control for elastic traffic. In *European Transactions on Telecommunications*, 1997.

[19] M. Andrew et al. Optimal Utility Based Multi-User Throughput Allocation subject to Throughput Constraints. In *Infocom*, 2005.

[20] H. Kellerer et al. *Knapsack Problems*. Springer, 2004.

[21] 3GPP Standard Rel 8. http://www.3gpp.org/Release-8.

[22] OPT++: An Object-Oriented Nonlinear Optimization Library. https://software.sandia.gov/opt++/.

[23] R. Morris et al. The click modular router. *SIGOPS Oper. Syst. Rev.*, 33(5), 1999.

[24] R. Mahindra et al. Farsighted Flow Management for Video Delivery in wireless networks. In *COMSNETS*, 2011.

[25] G. Liebl et al. Advanced Wireless Multiuser Video Streaming using the Scalable Video Coding Extensions of H.264/MPEG4-AVC. In *IEEE ICME*, 2006.

[26] M. Lu et al. Video streaming over 802.11 wlans with content-aware adaptive retry. In *Proc. IEEE Int. Conference on Multimedia and Expo*, 2007.

[27] H. Zhang et al. Cross-Layer Optimization for Streaming Scalable Video over Fading Wireless Networks. In *IEEE JSAC*, 2010.

[28] M. Burza et al. Adaptive Streaming of MPEG-based Audio/Video Content over Wireless Networks. *Multimedia*, 2007.

[29] R. Mok et al. QDASH: QoE-aware DASH system. In *MMSys*, 2012.

[30] Openwave Media Optimizer. http://www.openwave.com/solutions.

[31] B. Shen et al. Dynamic video transcoding in mobile environments. In *Proc. IEEE Multimedia*, 2008.