Improving mobile video streaming with link aware scheduling and client caches

Randeep Bhatia*, T V Lakshman*, Arun Netravali†, Krishan Sabnani*
*Bell Labs, Alcatel-Lucent †OmniCapital Group

***Abstract*—The rapid growth in multimedia traffic is straining mobile networks thus necessitating the need for efficient content delivery mechanisms. In this paper we present the design and analysis of a scheme for streaming non-live, pre-recorded content (e.g. Video on Demand) that opportunistically takes advantage of the "slow fading" variations in the wireless link quality. The proposed scheme works by selectively sending more content to sessions at times when they have better link quality while providing sufficient rate guarantees to keep their buffers from under-flowing. We establish analytically that the performance of such scheme is within two times that of any optimal scheme and that it results in throughput gains, per user and aggregate, that increase in proportion to the number of streaming users. Our performance evaluations indicate that by exploiting slow time-varying channels the streaming capacity can more than double with significant benefits to the users at the edge of the cell.**

I. Introduction

Mobile data traffic has been growing at an exponential rate with traffic volume doubling every year and with some predicting more than 18-fold increase in data traffic in five years [4]. As reported in [4], [15] every other bit being carried on mobile networks is now video, a significant portion of which is video on demand (non-interactive, pre-recorded content) with more than 25% coming from Youtube alone. Sustaining a constant high quality experience under wide variations in radio link channel quality due to fading is one of the major challenges for large scale delivery of video over mobile networks. Although considerable research has gone into designing schemes for the effective handling of fast fading, the slow-time variability in the wireless channel (e.g. by shadow fading) has not received the same attention. Simulations show that channel impairments due to slow fading can adversely impact the performance especially for users at the edge of the cell since their average channel quality is already quite low. In this work we directly address these challenges by designing and analyzing schemes that are particularly suited for content delivery under slow fading channels.

Existing schemes (e.g. proportional fair scheduling at the Base Station [11], [2]) take advantage of the fast variations in the wireless channels (from fast fading) by leveraging multi-user diversity. The small time windows (order of milliseconds) that they operate over is not sufficient to capture the effects of slow fading which can take many seconds or even minutes to exhibit any variations. Neither do these schedulers incorporate the state of the application buffers in the scheduling decisions, for example, to assign more resources to multimedia sessions at times when their buffers are running critically low. In addition, in many of these schedulers only a single user (or a few users) must be served in each time slot, due to the restrictions imposed by the underlying MAC layer, which can also limit their effectiveness.

The existing content streaming schemes (e.g. http adaptive streaming, progressive download) are also not very effective in dealing with channel variability due to slow fading. This is because they make independent and sometimes sub-optimal content transfer and adaptation decisions [18]. For instance in http adaptive streaming each client greedily maximizes the use of any available bandwidth by boosting up its video to the highest possible resolution. Since video streaming is typically carried out by stationary users who can sometimes receive highly unfair service by the radio network [17], this results in some users receiving very poor experience due to the lower rate allocated to them by the network.

In this paper we present one of the first scheme that comprehensively deals with slow fading with substantial performance gains. The proposed scheme works by exploiting the slow variations in the link quality to transfer more content at times when the users link is operating at higher efficiency. This delivers significant enhancement in the effective throughput for both the user and the network. In particular this is especially beneficial to the users at the edge of the cell. This is because their rate shows wide variation due to their link quality being significantly impacted by slow fading and our schemes effectiveness goes up with the increase in the variations in the link quality.

Although our scheme can be applied effectively for any type of data transfer under a slow fading channel, it is particularly suited for delivering non real time, recorded mobile multimedia content for which sustaining a high quality of experience is paramount. This is because under our scheme rate guarantees can be provided even over very short time intervals. This ensures interruption free playback at all times even when the user is getting lower data rates due to poor channel conditions.

Our scheme handles the inefficiencies of commonly used streaming technologies resulting from independent and "greedy" content transfer and adaptation decisions [18], whereby each session utilizes all its bandwidth resources to the maximum to transfers content at the highest possible rate. In our scheme the decisions are made globally by considering the channel condition of all other users in the cell. By allocating channel capacity in proportion to their link efficiency our scheme ensures that even users at the edge of the cell, can transfer more content as well as the aggregate network throughput is increased.

In our scheme the scheduling decisions are based on session performance metrics tracked over time windows, appropriately sized to incorporate the variability due to slow fading. Thus it overcomes the limitation of Proportional Fair Schedulers [11] that operate over much smaller time slots. Our scheme is also buffer aware. In our scheme data transfers are temporarily curtailed to high performing users once they are able to build large buffers and the freed up network resources are used to send more data to other users with low buffer occupancy. This ensures high streaming experience to all users including those who would have otherwise received unfairly poor performance. As observed in [17] such unfairness is commonly exhibited by existing schemes, particularly in stationary environments (ie for typical video streaming environment), where a single session may occupy most of available bandwidth and leave only a little portion for the other sessions.

Our scheme is very simple and can be efficiently implemented. It is suitable for deployment at the application layer and also at a lower layer (e.g. MAC scheduler). Although we highlight a few different deployment architectures our main focus is the design and analysis of the scheme. Through analysis and simulations we demonstrate that the scheme is highly effective in enhancing the efficiency of multimedia content delivery over mobile networks.

Our Contributions

We have designed a simple scheme for content streaming which is highly effective in dealing with slow fading on the wireless link. We show how to implement this scheme on top of existing systems with server and client side monitoring and flow regulation. We have evaluated the performance of the scheme using an open source LTE simulator [14]. We find that with our scheme 100% more users can be streaming at the same time compared to existing schemes. We have also validated these gains analytically. In particular we show that with this scheme the network throughput can increase by as much as $O(\sqrt{\log k})$ for k streaming users. In addition, we show that our scheme performs within 2 times of any scheme for optimizing network efficiency. We also show that with our scheme the rates delivered exceed those by the proportionally fair scheme, thus making it ideally suited for video streaming.

II. A Motivating Example

Consider two users A and B with time varying channel as shown in Figure 1. Here solid line (dashed line) is used to represent the channel for user A (B). User A is in better radio geometry than user B and hence receives better average channel quality (and rate). Depending on their channel quality user A's channel toggles between 7 Mbps and 5 Mbps, while user B's channel toggles between 3 Mbps and 1 Mbps. Thus if A was the only user in the system its rate would toggle between 7 and 5 Mbps for an average rate of 6 Mbps. Likewise if B was the only user in the system it would get an average rate of 2 Mbps. When both users are active at the same time then they will share the network proportionally resulting in an average rate of $(6+2)/2 = 4$ Mbps, with A getting a rate of $6/2 = 3$ Mbps and B getting a rate of $2/2 = 1$ Mbps. Now consider the scheme where users are picked for data transfer based on how much better their channel quality is compared to their average channel quality. First, consider picking only one user for data transfer in each time interval, namely the one that maximizes the ratio of their current rate to their average rate. Thus with reference to Figure 1, A will be picked for data transfer in the first time interval while only B will do data transfer in the second time interval and so on. Thus, only one user is active at any given time and when active, A will get a rate of 7 Mbps and B will get a rate of 3 Mbps. Since each user will only be active half the time, users A and B will get an average rate of $7/2$ and $3/2$ Mbps respectively for an overall average rate of $7/2 + 3/2 = 5$ Mbps. Thus by prioritizing users with above average channel quality for data transfer not only each user gets higher rate (16.7% more for A and 50% more for B) but also the overall capacity of the network is improved (25% more).

The prioritization of the users however does not have to involve selecting only one user in each time interval. Let us now consider a modified version of the scheme where a portion of the time interval is allocated to each user in proportion to their current to average rate ratios (CARR). Thus with reference to Figure 1 since A's CARR is $7/6$ and that of B is $1/2$ they will get assigned fractions 0.7 and 0.3 of the first time interval respectively. Likewise A and B will be assigned the fractions 0.36 and 0.64 of the second time interval respectively. Thus with this assignment user A's (user B's) average rate over the two intervals will be $(0.7*7+0.36*5)/2 = 3.35$ ($(0.3*1+0.64*3)/2 = 1.11$) Mbps, implying an 11% increase in both the users rate and the overall networks capacity.

Both of these schemes are a special case of the scheme presented in Section IV (second one for $l = 1$ and the first one for very large values of l). Note that both these schemes benefit all users including the users in bad geometry. We show this result more formally both analytically and via simulations.

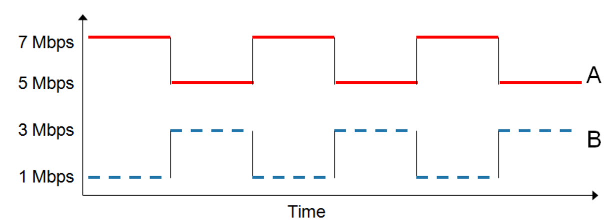

Fig. 1. Improving performance by exploiting time based channel variability

III. System Design

Our scheme can be implemented alongside the MAC scheduler in the Base Stations or may be implemented at a higher layer as described here. There is a network server component (NSC) that has the global view of the streaming sessions and can optionally regulate the flow of the content from the content server (e.g. CDN cache) to the device media clients. The NSC tracks per session information including flow rates, buffer state and client's cell site location. The collected information is used to identify the set of sessions within a given cell and their per time slot rates $r_i(j)$ and average rates R_i which in turn is used to compute the data transfer schedule for the sessions according to the algorithm described in Section IV-A. Figure 2 shows an architecture where the NSC is located between the CDN and the clients and proxies (e.g. as a split TCP proxy) all the session traffic to the clients.

In the case when the NSC also acts as a regulator it adaptively adjusts the data transfers to the sessions according to the rate assignments computed by the scheduling algorithm. Alternatively, the rate regulations can be implemented on the device client side component (CSC). In this case the NSC would not necessarily have to be in the data path but will only adaptively select the data rates for each session with the actual rate enforcement implemented by the CSC. With the CSC the NSC can get better visibility into the session state (buffer state), client mobility and even channel quality. In addition it can also provide a more scalable distributed implementation of the algorithm including rate adjustment enforcement. However the downside is that it may require additional changes to existing client side media streaming application for these new functionalities. The system architecture with both NSC and CSC components is depicted in Figure 3.

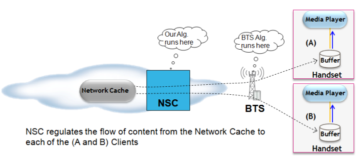

Fig. 2. As a proxy

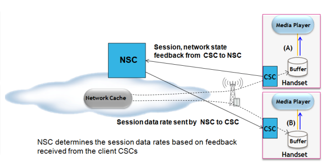

Fig. 3. On the control path

IV. Our Scheme

We start with some background about the radio channel. Figure 4 depicts the variations in the radio power received by a mobile as a function of time and thus also represents the variations of the received data rate since the data rate is directly related to the received signal to noise ratio. This variations has three components: a distance dependent constant path loss, a very fast variation called fast fading and another slower variation called shadow fading. The fast fading happens at very short time granularity (milliseconds). However the variation in shadow fading happens at much longer intervals. Shadow fading is known to have a lognormal distribution [8] with an autocorrelation function that decays exponentially with distance [5]. In particular the correlation between two points x meter apart is given by $e^{-\alpha x}$ where $\alpha = 1/20$ for environments intermediate between urban and suburban microcellular and higher for indoor users [10]. Moreover most video streaming is by relatively stationary users in indoor environments (where the de-correlation distance is of the order 1m-2m [10]). This implies that the link quality variations due to shadow fading for video streaming manifests at order of seconds or more.

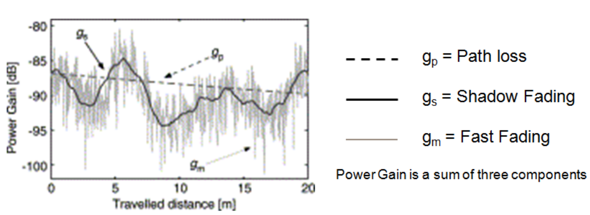

Fig. 4. Radio Link Power Gain as a function of time

We use the following notations in this section. We denote discrete time slots by $t_0, t_1, \ldots$. Each time slot is τ seconds long where τ is selected to be the order of a few seconds (for our tests we assumed $\tau = 5$ seconds). The choice of τ is driven by the need to effectively capture the variations in the channel due to slow fading, while "averaging" out the effects of fast fading. In each time slot the channel impairments (and hence the data rates) due to both Path Loss and fading can be assumed to be relatively static particularly for typical video streaming environments (stationary, indoor environment).

Consider a cell C and a time slot t_j. Let there be k streaming users $S = \{U_i, 1 \leq i \leq k\}$ in cell C at interval t_j. Let $r_i(j)$ denote the "individual" average rate of user U_i in time slot t_j. Averaging over the time slot is done to average out the impact of fast fading. By "individual" rate we mean the rate that the user would get if it was the only user in the cell. Thus with k users the rate $r_i(j)$ would be k times the observed rate for user U_i in time slot t_j. Let R_i denote the average rate for user U_i computed using the rates $r_i(j)$ over an appropriately selected sliding window of time slots (for our tests we used a sliding window of 12 slots or 1 minute). We describe later how the computation of $r_i(j)$ and R_i can be performed. Let $e_i(j) = \frac{r_i(j)}{R_i}$ be the ratio of the instantaneous to average rate for user U_i in time slot t_j. Let $B_i(j)$ denote the buffer occupancy for user U_i at the beginning of time slot t_j. $B_i(j)$ is measured in units of time slots (of size τ). This means that the playback for user U_i can continue for $B_i(j) * \tau$ seconds from its buffer alone (without any additional data transfer from the network).

A. Algorithm

In our algorithm the dynamic scheduling of data transfers is performed once every time slot $t_0, t_1, \ldots$. The basic scheduling step of the algorithm is very simple:

At time slot t_j serve user U_i at rate:

$$f_i(j) r_i(j)$$

where

$$f_i(j) = \frac{e_i(j)}{\sum_i^k e_i(j)}$$

Intuitively, $f_i(j)$ fraction of the "maximum cell" capacity is assigned to user U_i in time slot t_j, where $f_i(j)$ is directly proportional to the parameter $e_i(j) = \frac{r_i(j)}{R_i}$, or the ratio of the instantaneous rate to the average rate of user U_i. $e_i(j)$ represents how much more efficient user U_i's channel is in time slot t_j compared to her average channel quality (since rate is proportional to the channel quality). Since this efficiency is mainly because of the variability in the slow fading channel the parameter $e_i(j)$ represents the efficiency of the shadow fading channel in this slot for the user. In addition the normalization of the instantaneous rate $r_i(j)$ by the average rate R_i helps compare the efficiency of the channels for users who may be in very different radio conditions (e.g. users close to the BTS versus users at the edge of the cell) and hence may be getting very different rates.

By allocating more resources to users with higher link efficiency our scheme increases the overall network efficiency. However we need to make sure, particularly for video streaming, that these gains are not obtained at the expense of some users who end up getting lower performance. Also, the algorithm should not under-serve the users who are at the edge of the cell or do not have a good connection to the base station. We show that this is not the case with our scheme. In particular we now show that our algorithm can provide rate guarantees to all k users even over short time intervals such that their effective rate can only increase. In fact we show something much stronger. We show that the rate guarantees of our algorithm hold not just for the shadow fading model but hold more generally for an adversarial model where an adversary may select the rates of the user. Thus, the algorithms rate guarantees holds for any size interval and any user rate distributions. We also show that in our scheme no user is under-served and in fact each user gets strictly better rate compared to a proportional fair allocation.

Consider a set of n time slots $t_m, t_{m+1}, \ldots t_{m+n-1}$ for some m and n. Let R_i^* denote the average rate of user U_i in these n slots. That is $R_i^* = \sum_{j=m}^{m+n-1} r_i(j)/n$. Let $e_i^*(j) = \frac{r_i(j)}{R_i^*}$ be the users channel efficiency measures. Then:

Theorem 1: Over the n time slots, the total rate $T(i)$ for user U_i, when scheduled by the algorithm, exceeds $\sum_{j=m}^{m+n-1} r_i(j)/k$.

Proof: Note that for all i:

$$\sum_{j=m}^{m+n-1} e_i^*(j) = \sum_{j=m}^{m+n-1} \frac{r_i(j)}{R_i^*} = n$$

Therefore:

$$\sum_{j=m}^{m+n-1} (e_i^*(j) - \frac{\sum_{i=1}^{k} e_i^*(j)}{k}) =$$

$$n - \frac{\sum_{i=1}^{k} \sum_{j=m}^{m+n-1} e_i^*(j)}{k} = n - \frac{nk}{k} = 0$$

Let among these time slots the set S denote those time slots for which $e_i^*(j)$ is no more than the average value $\frac{\sum_{i=1}^{k} e_i^*(j)}{k}$. Thus $S = \{j | e_i^*(j) - \frac{\sum_{i=1}^{k} e_i^*(j)}{k} \leq 0\}$. Let the set $\bar{S}$ contain the remaining n time slots. Note that since users channels are independent, each of the sets S and $\bar{S}$ must be non-empty. Since, $\sum_{j=m}^{m+n-1} (e_i^*(j) - \frac{\sum_{i=1}^{k} e_i^*(j)}{k}) = 0$ it follows:

$$\sum_{j \in S} (\frac{\sum_{i=1}^{k} e_i^*(j)}{k} - e_i^*(j)) = \sum_{j \in \bar{S}} (e_i^*(j) - \frac{\sum_{i=1}^{k} e_i^*(j)}{k})$$

Note that since:

$$\forall_{j \in S} \frac{e_i^*(j)}{\frac{\sum_{i=1}^{k} e_i^*(j)}{k}} \leq 1 \quad \text{and} \quad \forall_{j \in \bar{S}} \frac{e_i^*(j)}{\frac{\sum_{i=1}^{k} e_i^*(j)}{k}} > 1$$

it follows:

$$\sum_{j \in S} \frac{e_i^*(j)}{\frac{\sum_{i=1}^{k} e_i^*(j)}{k}} (\frac{\sum_{i=1}^{k} e_i^*(j)}{k} - e_i^*(j)) <$$

$$\sum_{j \in \bar{S}} \frac{e_i^*(j)}{\frac{\sum_{i=1}^{k} e_i^*(j)}{k}} (e_i^*(j) - \frac{\sum_{i=1}^{k} e_i^*(j)}{k})$$

By rearranging we get:

$$\sum_{j=m}^{m+n-1} \frac{e_i^*(j)}{\sum_{i=1}^{k} e_i^*(j)} e_i^*(j) > \frac{\sum_{i=1}^{k} e_i^*(j)}{k}$$

Multiplying both side by R_i^* and substituting for $r_i(j)$ and $f_i(j)$ we get:

$$T(i) = \sum_{j=m}^{m+n-1} f_i(j) r_i(j) > \sum_{j=m}^{m+n-1} r_i(j)/k$$

thus establishing the bound.

■

Note that $\sum_{j=m}^{m+n-1} r_i(j)/k$ is also the rate for user U_i for a proportional fair allocation in which each user gets the same share of each time slot. Thus with our algorithm each user gets strictly better rate compared to when each user independently tries to transfer data at the fastest possible rate.

In our algorithm a user U_is average rate R_i^* is estimated from the rate values observed in the past. In particular we average over a sliding window of n slots to estimate R_i^*. Thus, $R_i^* = R_i^*(1 - 1/n) + r_i(j)(1/n)$, as in [11], is the averaging mechanism to compute R_i^* at time slot t_j. Here n is selected to be large enough (between $10 - 20$ slots) so that the observed rate average holds steady for the next n time slots as well. The algorithm as described requires the values $r_i(j)$ before doing the scheduling for the time slot. In order to compute these values, the algorithm starts out by allowing all users to independently proceed with their data transfers at the beginning of time slot t_j. The rates of the users are observed to compute $r_i(j)$ and then using these values the algorithm accordingly computes the fractions and regulates the flows for the users. With this approach it follows that the algorithm need not operate in discrete time slots but rather the rates R_i^* and $r_i(j)$ could be continuously tracked and the schedule dynamically adjusted to reflect the changes in these rates. However, for ease of exposition we will continue to assume that the scheduling is performed in discrete time slots.

One way to increase the network efficiency is to bias the scheduling even more towards the users with higher values for $e_i(j)$. So far in the algorithm the time slot fractions $f_i(j)$ were allocated in proportion to the $e_i(j)$ values. Now we consider assigning the time slot fraction $f_i(j)$ in proportion to $e_i^l(j)$ for some parameter $l \geq 1$. Thus we set:

$$f_i(j) = \frac{e_i^l(j)}{\sum_i^k e_i^l(j)}$$

Note that when $l = 1$ we get the basic algorithm. Also note that as l is made very large the algorithm will assign majority of the time slot to the users with the largest $e_i(j)$ values. And in the limit only the user with the largest $e_i(j)$ value will be scheduled in the time slot. Thus in the limit the algorithm will derive the highest network efficiency. This also follows from our analytical results (in Section V) where we show that the efficiency of this algorithm is close to optimal even in the worst case (Theorem 4) and it has significant gains under Shadow Fading (Theorem 2).

One potential downside of larger l is that the rate guarantees (see Theorem 1) may not necessarily hold for short time intervals. Such guarantees may also not hold for a few initial slots at the beginning (when video streaming is started up). Also a few initial slots may be needed to build up a stable estimate for R_i^*. Thus, an initial buffer may be needed before starting the playback to prevent buffer underflow at startup. Subsequently, given the rate guarantees of the algorithm, this buffer will be kept filled thus avoiding the risk of interruptions in playback. From our simulations (Section VII-B) we find that the initial buffer do not have to be very large. All our simulations performed well with initial buffers of no more than a minute and there were no significant degradation even at much lower buffer sizes. We therefore further enhance the basic algorithm to also consider the users buffer occupancy $B_i(j)$ in making the scheduling decision. In particular once user U_is buffer occupancy $B_i(j)$ exceeds a high watermark threshold θ_H the algorithm stops assigning it any portion of the time slot until its buffer occupancy falls below the low watermark θ_L. In other words the algorithm takes away the share of the time slot from the high performing users and assigns it to the low performing users. This has two benefits. One, this lets the algorithm operate at a much higher network efficiency and second it helps deal with the unfairness of the existing link aware schemes particularly for stationary users [17]. In our simulations we set θ_H (θ_L) to two (one) times the initial buffer size.

The high level pseudo-code for the algorithm is shown below for a particular user U_i. First content is sent to build up the users initial buffer. After that the playback starts and content is only sent when the user is "Active" (with low buffer occupancy) in the time slot ($A_i = true$). The amount of content transferred is determined by the time slot fraction allocations performed by the algorithm. Here n is the sliding window size in time-slots. The A_i values are updated at the end of every time slot according to U_i's buffer size.

Algorithm. Scheduling of Video Streaming to U_i

```
A_i ← true      //User is active
for Initial Buffer Filling do
    R_i ← Average rate for U_i
end for
for j = 0, ... do
    1...k ← "Active" users at time-slot t_j
    if A_i then      //User is active
        r_i(j) ← Data rate of U_i in time-slot t_j
        e_i(j) ← r_i(j) / R_i
        f_i(j) ← e_i^l(j) / Σ_i^k e_i^l(j)
        Allocate f_i(j) fraction of time-slot t_j to U_i.
        R_i ← R_i(1 − 1/n) + r_i(j)(1/n)
    end if
    if B_i(j) ≥ θ_H then
        A_i ← false
    else if B_i(j) < θ_L then
        A_i ← true
    end if
end for
```

□

V. Analytical Results

We compare our algorithms to a standard proportional fair algorithm. We show gains in aggregate throughput (and per user throughput), in proportion to the number of users. First, we analyze the algorithm for $l \to \infty$. In this case the algorithm assigns the entire time slot to the user with the largest value of $e_i(j)$ and hence achieves the highest possible network efficiency. We call this the greedy scheduler. We show that for this algorithm the aggregate throughput increases increases by as much as $O(\sqrt{\log k})$ for k streaming users. Next we show that even for the basic algorithm ($l = 1$), which has the least network efficiency, the aggregate throughput increases as a function of k. This therefore establishes the result for all l. Although shown for aggregate gains these two results apply equally to per user gains as well. Next we establish that our algorithms are also the best possible. In particular we show that the worst case (for an adversarial model) our greedy scheduler performs within 2 times of any optimal solution in terms of aggregate gains.

Theorem 2: The aggregate throughput of k streaming users (within a cell) increases by $c\sqrt{\log k - \log\log k}$ for some constant c with the greedy scheduler.

Proof: We start out by considering the case where all of the k users have identical average SNR during each scheduling interval. The variations in SNR are due to shadow fading since the effects of fast fading get averaged out in larger time slots. The SNR of the i-th user in the scheduling time slots is distributed as $L \cdot Y_i$ where L is its average SNR and Y_i is a random variable with a lognormal distribution and the random variable $X_i = \log Y_i$ has a normal distribution with mean 0 [8].

The data rate R in bits per second that can be achieved in an AWGN channel is related to SNR by $R = B\log_2(1+SNR)$ where B is the channel bandwidth in Hertz [16]. In a purely TDMA proportional fair system each user gets an average rate of $\frac{1}{k}B\log_2(1+L)$ for a total system rate of $B\log_2(1+L)$. With a greedy scheduler that always schedules the best user (highest rate or SNR) in each scheduling interval the data rate achieved for the system is distributed as $R = B\log_2(1+LZ)$, where $Z = \max_i Y_i$. The expected rate of the system is therefore $\mathbb{E}[R] = B\,\mathbb{E}[\log_2(1+LZ)]$. We show below that $\mathbb{E}[\log_2(1+LZ)]$ can be significantly higher than $\log_2(1+L)$ thus establishing the performance bound. In the following we will no longer specify the base of the logarithms since that only changes the results by a constant factor. By Markov's inequality (For a non-negative random variable X the expectation $\mathbb{E}[X] \geq \gamma P(X \geq \gamma)$, where $P()$ denotes the probability distribution function):

$$\mathbb{E}[\log(1+LZ)] \geq \log(1+L\gamma)P(\log(1+LZ) \geq \log(1+L\gamma))$$

or

$$\mathbb{E}[\log(1+LZ)] \geq \log(1+L\gamma)P(Z \geq \gamma) \tag{1}$$

Now

$$P(Z \geq \gamma) = P(\max_i Y_i \geq \gamma) =$$

$$P(\max_i \log Y_i \geq \log\gamma) = P(\max_i X_i \geq \log\gamma)$$

Since each of the X_i is identically distributed we can omit the index i:

$$P(Z \geq \gamma) = 1-(P(X < \log\gamma))^k = 1-(1-P(X \geq \log\gamma))^k \tag{2}$$

Since X has a normal distribution with mean 0 we have:

$$P(X \geq t) = \frac{1}{\sigma\sqrt{2\pi}}\int_t^\infty e^{-\frac{1}{2}(\frac{x}{\sigma})^2}\,\mathrm{d}x = \frac{1}{\sqrt{\pi}}\int_{t/(\sigma\sqrt{2})}^\infty e^{-z^2}\,\mathrm{d}z$$

The last equality follows from the substitution $x = \sigma z\sqrt{2}$. But $\frac{1}{t+\sqrt{t^2+2}} < e^{t^2}\int_t^\infty e^{-x^2}\,\mathrm{d}x$ [1]. Therefore it follows:

$$P(X \geq t) > \sigma\sqrt{\frac{2}{\pi}}\frac{1}{(t+\sqrt{t^2+4\sigma^2})e^{\frac{1}{2}(\frac{t}{\sigma})^2}} \tag{3}$$

In the following c stands for any constant value. Since c is only used in functions that grow with k the actual values of c do not matter.

It follows that as long as t is $c\sqrt{\log k - \log\log k}$ the right side of the Equation 3 is at least $c\sqrt{\log k}/k$. Thus for $\log\gamma = c\sqrt{\log k - \log\log k}$ we have from Equation 2:

$$P(Z \geq \gamma) > 1 - (1 - c\sqrt{\ln k}/k)^k = 1 - (1/e)^{c\sqrt{\log k}} \tag{4}$$

thus showing that $P(Z \geq \gamma)$ approaches 1 for $\log\gamma = O(\sqrt{\log k - \log\log k})$ for large values of k. It therefore follows from Equation 1 that for large k the expected rate $\mathbb{E}[\log(1+LZ)]$ grows as $\log(1+L\gamma)$ where $\log\gamma = c\sqrt{\log k - \log\log k}$. In other words the greedy scheduler increases the spectral efficiency of the network from $\log_2(1+L)$ to $\log_2(1+L) + \Omega(\sqrt{\log k - \log\log k})$ thus establishing the bound.

Now consider the more general case of non-identical average SNR L_i for each user i. Since each user gets served equally by the proportional fair scheduler each user i gets an average rate of $\frac{1}{k}B\log(1+L_i)$ for a total system rate of $\frac{1}{k}B\sum_i\log(1+L_i)$. The greedy scheduler selects the user which attains the highest gain over its average SNR. In other words the user picked is one for which $Z = \max_i Y_i$ is attained. The expected overall rate of the system can therefore be computed by conditioning on the user that gets picked at any given scheduling interval. Let us consider user j. The conditional probability that this user gets picked is $1/k$ since every user is equally likely to be picked (since the random variables Y_i are identically distributed). The expected spectral efficiency when user j is picked is $\mathbb{E}[\log(1+L_jZ)]$ which by the analysis above exceeds $\log(1+L_j)$ by $\Omega(\sqrt{\log k - \log\log k})$. Since this is applicable for any user j and since each user is equally likely to be picked by the scheduler we get that that the system efficiency $\frac{1}{k}\sum_i\mathbb{E}[\log(1+L_iZ)]$ increases by $\Omega(\sqrt{\log k - \log\log k})$.

■

Theorem 3: Compared to the naive algorithm, the algorithm defined in Section IV-A delivers additional aggregate throughput that increases in proportion to the number of users k.

Proof: We only provide a proof sketch. Let the mean channel bandwidth be denoted by μ. Thus, $R_i = \mu$ for all users i. Let $\alpha_1, \alpha_2 \ldots$ denote the ratio of instantaneous rate to the average channel bandwidth. Thus for user i in time slot j the instantaneous rate may be given by $r_i(j) = \alpha_r\mu$ and $e_i(j) = \frac{r_i(j)}{R_i} = \alpha_r$. Consider a set of distinct values for $\alpha_1, \alpha_2 \ldots \alpha_k$ and consider a set E (of size k!) of events in which the k users instantaneous channel is determined by each permutation of these parameters. For example an event in E may correspond to the permutation $1, 2, 3, \ldots k$ where user i's parameter is α_i. Let us consider a user i. Let us consider all such events in E in which this users parameter is α_r. In each of these events user i is assigned a rate of $\frac{\alpha_r}{\sum_k \alpha_k}(\alpha_r\mu)$. Note that this is applicable for all parameters $\alpha_r, 1 \leq r \leq k$. Also note that every event in E has the same probability since the users channels are identically distributed. In addition note that the number of events in which user i's parameter is α_r are the same for every r. Thus, it follows that the average rate that user i gets for all such events in E is $\mu\frac{1}{k}\frac{\sum_r \alpha_r^2}{\sum_r \alpha_r}$. We compare this to the average rate that user i would get when using the naive algorithm where the channel is shared proportionally by the users. Note that this is $\frac{\mu\alpha_r}{k}$ at any event in E for which the users parameter is α_r. Thus, it follows that the average rate that user i gets, with the naive algorithm, among all such events in E is $\mu\frac{1}{k}\sum_r \alpha_r$. Thus the average rate gain for user i, with our algorithm over the naive algorithm, for the events in E is:

$$\frac{\mu}{k}\frac{\sum_r \alpha_r^2}{\sum_r \alpha_r} - \frac{\mu}{k}\frac{\sum_r \alpha_r}{k} \geq \frac{\mu}{k^2}\frac{\sum_{m\neq n}(\alpha_m - \alpha_n)^2}{\sum_r \alpha_r} \tag{5}$$

It therefore follows that there is a positive rate gain with our algorithm and since this applies to all choices of the values for α_r we have shown that overall there is a positive rate gain. We now quantify the magnitude of this gain in terms of k.

Let X_r be a random variable which is 1 if $0.5 \leq \alpha_r \leq 1.5$ and 0 otherwise. Let the probability of X_r being 1 be p. Note that p depends on the underlying channel probability distribution. Let $X = \sum_r X_r$. Note that the expected value of X is kp. Using Chernoff bounds [3] it follows for any δ_1:

$$P(X \geq (1+\delta_1)kp) \leq e^{-\delta_1^2 kp/3}$$

Thus with probability at least $1 - e^{-\delta_1^2 kp/3}$ there are k' values of α_r outside the range $(0.5, 1.5)$, where $k' \geq k_1 = k(1-p-\delta_1 p)$.

As before we define a random variable Y_r which is 1 if $\alpha_r > 1.5$ and 0 otherwise and let $Y = \sum_r Y_r$ for k_1 such Y_r. Let the probability of Y_r being 1 be p' (p' depends on the channel distribution). Since there are k_1 such r the expected value of Y is k_1p'. Let δ_2 be a parameter. From Chernoff bound we get

$$P((1-\delta_2)k_1p' \leq Y \leq (1+\delta_2)k_1p') \geq$$
$$1 - (e^{-\delta_2^2 k_1 p'/3} + e^{-\delta_2^2 k_1 p'/2})$$

Combining these results we get that the probability of the event in which at least $K = (1-\delta_2)k_1p' = k(1-p-\delta_1p)(1-\delta_2)p'$ of the α_r values satisfy $\alpha_r > 1.5$ and at least the same number satisfy $\alpha_r < 0.5$ (since $k_1 - (1+\delta_2)k_1p' \geq (1-\delta_2)k_1p'$) is at least $P(k) =$

$$(1 - (e^{-\delta_2^2 k_1p'/3} + e^{-\delta_2^2 k_1 p'/2}))(1 - e^{-\delta_1^2 kp/3}) \tag{6}$$

The $\alpha_r, 1 \leq r \leq k$ values for the above defined event can be partitioned into 3 sets: S_1, S_2 an S_3, based on whether $\alpha_r < 0.5$ or $\alpha_r > 1.5$ or otherwise respectively. Note that the cardinality of both S_1 and S_2 is at least K. For this event we have $\frac{\sum_{m\neq n}(\alpha_m-\alpha_n)^2}{\sum_r \alpha_r} \geq \frac{\sum_{\alpha_m\in S_2}\sum_{\alpha_n\in S_2}(\alpha_m-\alpha_n)^2}{1.5k+\sum_{\alpha_r\in S_2}(\alpha_r-0.5)}$ and therefore is at least

$$\frac{K\sum_{\alpha_m\in S_2}(\alpha_m-0.5)^2}{1.5k+\sum_{\alpha_r\in S_2}(\alpha_r-0.5)} \geq \frac{K}{1.5k/K+1} \geq \frac{K}{1.5c+1}$$

for some constant c. Here we used the facts that $\alpha_r - 0.5 \geq 1, \ \forall \alpha_r \in S_2$ and that the ratio k/K is a constant independent of k. Combining with the lower bound on the probability of this event $P(k)$ (Equation 6) along with the Equation 5 it follows that the expected rate gain for a user i over the naive algorithm is at least $\frac{\mu}{k}P(k)c/(1.5c+1)$. This implies that the aggregate rate gain with our algorithm for all k users is $\mu P(k)c'$ for some constant c'.

■

Next we show that the greedy schedulers performance is close to the optimum even in the worst case. Specifically we show that it is able to transfer at least half the amount of content from the network compared to any optimal and even offline scheme for such a transfer that may even operate with full knowledge of the future in terms of the clients channel quality variations and the congestion in the network. The result also holds independent of the order in which streaming requests may arrive in the system and in particular it holds at any given time when the comparison is made.

Theorem 4: The greedy online scheduler is a 2 approximation algorithm.

Proof: Let us denote by a demand a clients request for the transfer of data. Consider the set of demands served by an optimal scheme at time t and compare it with the set of demands served by the greedy scheduler by the same time t. Let $A_1, A_2, \ldots, A_n$ be the set of demands which are not served by the greedy scheduler but are served by the optimal scheme by time t. Note that it can be the case that a demand is partially served by the greedy scheduler or the optimal scheme. However this is merely a technicality since we can think of each streaming request to be made up of multiple demands so that each of the A_i can be thought of as a full demand. Consider a demand A_i and let t_i be the time interval in which it was served by the optimal scheme. Note that the greedy scheduler also had the opportunity to serve demand A_i at time interval t_i but decided to serve some other demands in that time interval. However, since the greedy scheduler is locally optimal it must have scheduled those demands in time interval t_i which could transfer data more efficiently than the demand A_i (ie at a higher rate). Hence the greedy scheduler must have transferred more data than the optimal scheme during each of the intervals t_i. In other words the total amount of data transferred by the greedy scheduler by time t exceeds the size of all demands A_i (ie $\sum_i^n |A_i|$). Note also that compared to the data transferred by the optimal by time t, the greedy scheduler had only $\sum_i^n |A_i|$ data left to transfer by time t. Hence the greedy scheduler has transferred at least half the data by time t as compared to the optimal scheme thus establishing the result.

■

VI. Numerical Results

Here we present the result of testing our algorithm on a simulated shadow fading channel. In Section VII we present our simulation results using LTE-Sim [14]. The channel for each user is independently generated. Let Y_i be the SNR of user U_i due to shadow fading. Then $10\log_{10} Y_i$ is Gaussian distributed with mean 0 and standard deviation 8 dB [8]. The overall SNR of user U_i, with all propagation losses, is assumed to be Y_i/K for a constant K. The channels are assumed to be AWGN with user U_i's rate $r_i(j)$ in time slot t_j given by $r_i(j) = B\log_2(1 + Y_i/K)$. The parameter K is selected to be a constant value for which the spectral efficiency $\log_2(1 + Y_i/K)$ is close to 1. In our tests we found the best fit in the range $1 \leq K \leq 3$. We set B the bandwidth of the channel at 3MHz. In all these tests the scheduling was only link aware (all users stayed Active all the time regardless of their buffer occupancy).

A. Streaming Capacity Gains

We compared the number of streaming sessions that can be supported with our scheme versus the progressive download (PD) streaming scheme. We fixed the maximum number of simultaneously streaming users at 20. We computed the maximum video rate (M) that can be supported, for each one of the 20 users without buffer overflow, using our scheme. We assumed a half an hour long video (same video watched by all users). We assumed a startup buffer of size 1 minute of the video. We then tried the same setup for progressive download and computed the maximum number of users (out of the 20 users) that could simultaneously stream this video at rate M without buffer overflow. Let N be this number. Then $20/N$ is the streaming capacity gain of our scheme. We show the results for the gain for three values of K in Figure 5. On the X axis is shown the percentage gain and on the Y axis we show the cumulative probability distribution (CDF) for the gains. It can be seen that the gains occasionally exceed 100% with average gain around 65%. It can also be seen that the gains are likely to be higher at higher values of K (ie for users with lower SNR values). This implies that the algorithm has better performance for users at the edge of the cell since they have much lower average SNR. This is also confirmed by our simulation results which are presented in the next section. For this test we used the value $l = 1$. The results for the case $l = 2$ are shown in Figure 6. It can be seen that the gains are somewhat less for this case. However, as we show below the advantage of using higher values of l are the significant increase in the network efficiency.

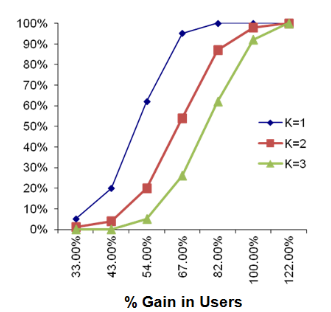

Fig. 5. $l = 1$

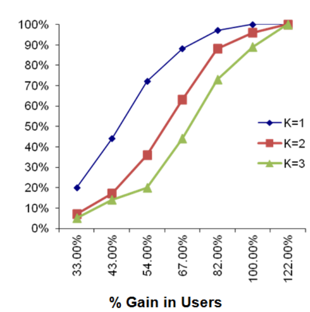

Fig. 6. $l = 2$

B. Throughput Gains

We compared the total throughput of the streaming sessions with our scheme versus the PD streaming scheme. We fixed the value of K between 1 and 3 and we fixed the number of users k between 10 and 17. In each case we ran multiple tests to compute the total rate obtained by each user using our scheme. For each test we took the minimum value for the total rate (say T_1) among all users. We ran the same test for the PD scheme (using the same $r_i(j)$ values) and computed minimum value for the total rate (say T_2) among all users. Then the rate gain is computed as the ratio T_1/T_2. The results for $l = 1$ are shown in Figure 7. As can be seen from the Figure the minimum rate gain is at least 60% and can go as high as 90% for $K = 3$.

The results for $l = 2$ are shown in Figure 8. The throughput gains are significantly more compared to the case $l = 1$ implying much higher network efficiency at higher values of l. Thus the choice for l helps trade-off between per user efficiency (Streaming capacity) gain versus the network efficiency (Throughput) gain.

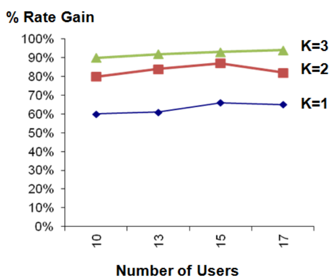

Fig. 7. $l = 1$

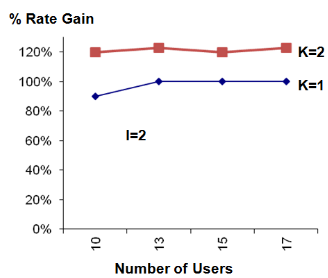

Fig. 8. $l = 2$

VII. Simulation Results

We now present the results of applying the algorithm on a simulated LTE network created using LTE-Sim [14]. We choose urban macro-cellular cells of 3 km radius. We used a random placement of video streaming stationary users within a cell.

A. Throughput Gains

Our first observation is that Slow Fading can cause significant rate variability around the mean for the users at the edge of the cell (Figure 9). This is an ideal situation for our algorithm and as a result the performance of such users is significantly enhanced by the algorithm. The increase in aggregate throughput is depicted in Figure 10. Here the simulation was carried out with $k = 15$ streaming users in a cell. We sorted these users in the non-increasing order of their average channel quality as measured by their average download throughput (without applying the algorithm). Let this ordering be $U_1, U_2, \ldots U_k$. In this ordering users close to the cell are on the left while users at the edge of the cell are towards the right. For each i we computed the ratio g_i of increase in aggregate throughput for all users $U_i, U_{i+1} \ldots U_n$. These throughput gain ratios $g_1, g_2 \ldots g_k$ are shown in Figure 10. It can be seen that at higher values of i, which mainly corresponds to users at the edge of the cell, there can be more than tripling of the throughput implying strong gains for such users.

B. Streaming Capacity Gains

We compared the number of streaming sessions that can be supported with our scheme versus the progressive download (PD) streaming scheme. The setting is the same as in

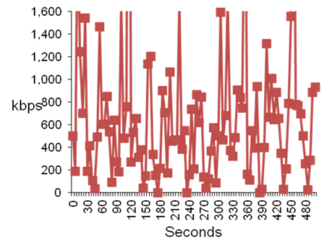

Fig. 9. Rate variations at cell edge

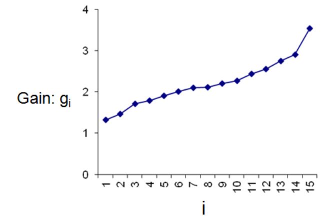

Fig. 10. Throughput gain ratios

the Section VI-A except that the simulation is done for 15 simultaneous users. We monitored the minimum video rate (M) that can be supported without buffer overflow for any user using both schemes. Note that M is determined by the performance of the worst user in the cell. The results are shown in Figure 11. As can be seen that with our scheme even at $l = 1$ the minimum supported video rate is almost double that of the PD scheme. The rate goes up even further as the value of l is increased. We also carried out the simulation when buffer awareness was included in the algorithm. The results are denoted by $l = 1, WB$ and $l = 3, WB$ for the case when $l = 1$ and $l = 3$ respectively. As can be seen that making the algorithm buffer aware provides further enhancements. Figure 12 shows how the minimum video rate is impacted by the startup buffer size both for our scheme (denoted by Alg) and the PD scheme. This is for the case $l = 1$. As expected the achieved rate is lower at smaller buffer size (measured in seconds). However, the performance gain is close to twice even at lower values of initial buffer size. Next, we show in Figure 13 the performance gains as the number k of streaming users is increased. The gain is the ratio of the minimum video rate (M) that can be supported in our scheme versus that in the PD scheme. In this case we used $l = 1$ with a 30 second initial buffer. The performance gain increases significantly at 20 or more users. This is because of the greater likelihood of having many users at the edge of the cell thus providing the rate diversity that is needed for the improved performance.

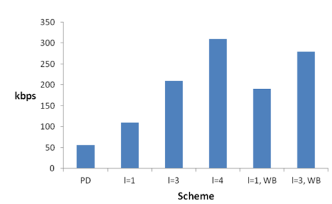

Fig. 11. Min. Video Rate (M) possible with each scheme

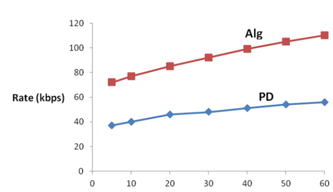

Fig. 12. Min. Video Rate (M) for different startup buffer sizes

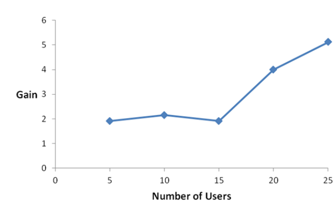

Fig. 13. Gain for different number of streaming users

VIII. Related Work

Optimizing delivery of video and content over mobile networks is an active area of research. Prior work has focused on the challenging problem of enhancing quality of experience for UDP (RTP) based real-time video streaming protocols (e.g. RTSP) over time varying and congestion prone wireless channels. This includes solutions [9], [22] for optimum rate selection at the video server or based on feedback from the video client on the end-to-end delay or the loss rate. More recently TCP/http based streaming protocols [13] that address many of the quality and scalability issues have gained popularity as well have received attention from the research community. Many bit rate selection schemes have been described, some that are based on bandwidth estimations at the server (multipath based [19]) or through client feedback [7]. Wireless link quality aware rate and content adaptation schemes have been developed such as those for link and content aware dropping of video frames to maintain quality [21] or more recently for adapting the senders rate based on network condition forecasting for interactive, live video [20]. However in many of these schemes the adaptation is done independently for each streaming session. Recently [18] proposed a scheme for rate computation and adaptation for Http Adaptive Streaming video and data sessions that maximizes global system utility. For stored video, and for exploiting Shadow Fading [12] developed a link and buffer aware scheduling algorithm. In their scheme dynamically computed rate threshold determine the users data transfer rates. Their work differs from ours in many respects. Their objective is to minimize the utilization of the wireless network and they mostly assume prior knowledge of the link conditions. A majority of their work is focused on the single user case while we address the more practical case of multiple streaming users contending for a cells capacity. Also, for the multi-user case no analytic guarantees are provided that can ensure that the client buffers will stay well supplied over short time windows, a critical requirement for video delivery. In the scheme of [6] greedy resource allocation is performed within time periods called epochs based on users channel quality estimates and buffer state. Unlike our scheme the scheduling decisions are driven more by the buffer state rather than the channel quality, even among users whose buffers are reasonably well occupied. Also their analytical results hold only for a single epoch while we show performance bounds for the full duration of the streaming.

IX. Conclusion

In this paper we introduced a novel link and buffer aware opportunistic scheduling scheme for enhancing streaming capacity for stored mobile video delivery. Specifically our scheme is designed to exploit the slow wireless fading, an area that has received relatively less attention in the literature. Our scheme is particularly suited for enhancing the quality of the users at the edge of the cell for whom the impact of fading is much more pronounced. Through extensive analysis and simulation we have established that the proposed scheme is highly effective with almost double the gains for mobile video delivery. Extending these results to adaptive streaming remains a future direction for this work.

References

[1] M. Abramowitz, I. A. Stegun *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* Dover Publications (1972), Formula 7.1.13.

[2] M. Andrews, L. Qian, and A. Stolyar *Optimal utility based multi-user throughput allocation subject to throughput constraints* IEEE INFOCOM 05, 2005.

[3] T. Hagerup and C. Rub *A guided tour of Chernoff bounds* Information Processing Letters 33 (6), 1990

[4] *Cisco visual networking index: Global mobile data traffic forecast.* 2011-2016

[5] H. Claussen *Efficient modeling of channel maps with correlated shadow fading in mobile radio systems.* IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications Sept. 2005.

[6] P. Dutta, A. Seetharam, V. Arya, M. Chetlur, S. Kalyanaraman, J. Kurose On managing quality of experience of multiple video streams in wireless networks IEEE Infocom, 2012

[7] P. Fröjdh, U. Horn, M. Kampmann, A. Nohlgren, M. Westerlund, *Adaptive streaming within the 3GPP Packet-Switched Stream. Service* IEEE Network, Mar. 2006

[8] A. Goldsmith *Wireless Communications* Cambridge University Press, 2005.

[9] C.-Y. Hsu, A. Ortega, M. Khansari, *Rate control for robust video transmission over burst-error wireless channels* IEEE J. Sel. Areas Commun., vol. 17, no. 5, 1999.

[10] N. Jalden, P. Zetterberg, B. Ottersten, H. Aihua, R. Thoma *Correlation Properties of Large Scale Fading Based on Indoor Measurements* IEEE WCNC 2007.

[11] A. Jalali, R. Padovani, R. Pankaj *Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system* Proc. of IEEE Vehicular Technology Conference, Tokyo, Japan Vol. 3 (May 2000)

[12] Z. Lu, G. D. Veciana *Optimizing stored video delivery for mobile networks: The value of knowing the future* IEEE Infocom, 2013

[13] K. Ma, R. Bartos, S. Bhatia, R. Nair *Mobile Video Delivery with HTTP* IEEE Comm. Magazine, 49,4, 2011

[14] G. Piro, L. Grieco, G. Boggia, F. Capozzi, P. Camarda *Simulating lte cellular systems: an open source framework* IEEE Trans. Veh. Technol., vol. 60, no. 2, Feb, 2011

[15] Sandvine : Global Internet Phenonomena Report 1H 2012

[16] C. E. Shannon *A Mathematical Theory of Communication.* Bell Sys. Tech. Journal, 1948

[17] F. P. Tso, J. Teng, W. Jia, D. Xuan *Mobility: A Double-Edged Sword for HSPA Networks* ACM MOBIHOC, 2010.

[18] D. D. Vleeschauwer, H. Viswanathan, A. Beck, S. Benno, G. Li, R. Miller *Optimization Of HTTP Adaptive Streaming Over Mobile Cellular Networks* IEEE INFOCOM 2013

[19] B. Wang, W. Wei, Z. Guo, D. Towsley *Multipath live streaming via TCP: Scheme, performance and benefits* ACM Trans. Multimedia Comp. Comm. Appl., 5(3), 2009.

[20] K. Winstein, A. Sivaraman, and H. Balakrishnan *Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks* NSDI 2013

[21] H. Zhang and S. Rangarajan *Adaptive scheduling of streaming video over wireless networks* IEEE Int. Conf. on Mult. and Expo, 2008

[22] X. Zhu, B. Girod *Video streaming over wireless networks* Proc. European Signal Processing Conference (EUSIPCO), 2007