Scheduling of Access Points for Multiple Live Video Streams

I-Hong Hou
CESG and Department of ECE
Texas A&M University
College Station, TX 77843, USA
ihou@tamu.edu

Rahul Singh
CESG and Department of ECE
Texas A&M University
College Station, TX 77843, USA
rsingh1@tamu.edu

ABSTRACT

This paper studies the problem of serving multiple live video streams to several different clients from a single access point over unreliable wireless links, which is expected to be major a consumer of future wireless capacity. This problem involves two characteristics. On the streaming side, different video streams may generate variable-bit-rate traffic with different traffic patterns. On the network side, the wireless transmissions are unreliable, and the link qualities differ from client to client. In order to alleviate the above stochastic aspects of both video streams and link unreliability, each client typically buffers incoming packets before playing the video. The quality of the video playback subscribed to by each flow depends, among other factors, on both the delay of packets as well as their throughput.

In this paper we address how to schedule packets at the access point to satisfy the joint per-packet-delay-throughput performance measure. We test the designed policy on the traces of three movies. From our tests, it appears to outperform other policies by a large margin.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design —*Wireless communication*

Keywords

Wireless networks; video streaming; delays; deadlines

1. INTRODUCTION

Multimedia applications for video streaming have been predicted to become a dominant portion of future wireless traffic [1]. In order to provide smooth playback to end users with multimedia applications, their packets need to be delivered in a timely manner and with a sufficient throughput. Providing the per-packet delay guarantees is particularly challenging for two reasons. First, on the application side, the video streams generate variable-bit-rate (VBR) traffic that congests the network when multiple streams generate a burst of packets at the same time. Second, on the network side, wireless transmissions are inherently unreliable due to shadowing, fading, and interference.

In order to mitigate the effect of the uncertainties in traffic bit rate as well as channel reliability, packets are typically buffered at the receivers. Each receiver waits a specified amount of time buffering incoming packets before it starts playing the video. On the positive side, this approach has the benefit of greatly reducing the impact of network congestion and channel unreliability. However, buffering also increases delay as the receiver cannot start playing the video immediately. For live video streaming, such as sports events and instant news, it is important to provide smooth playback while waiting only a reasonably small amount of time before playing the video.

In this paper, we analyze a model that addresses the three characteristics of stochastic packet arrivals, unreliable wireless channel, and buffering delay. We determine a simple online scheduling policy, the Earliest Positive-Debt Deadline First (EPDF) policy, for serving the live video streams that achieves the delay-throughput capacity. This policy does not require knowledge of the exact traffic pattern of each video stream, enhancing its suitability for implementation.

We test the proposed EPDF policy on three movie traces. We compare its performance with three other well known policies in ns-2. The EPDF policy outperforms the other policies by a large margin on these three video traces. We also propose a modification to enhance short-term performance, and investigate the tradeoff between long-term and short-term performance guarantees.

2. RELATED WORK

There have been several studies on providing delay guarantees for flows that require delay guarantees over wireless networks. Liu, Wang, and Giannakis [12] have proposed a scheduler that updates link priorities dynamically to provide QoS. Dua and Bambos [2] have proposed a heuristic that jointly considers wireless channel quality and packet deadlines. Raghunathan et al [13] and Shakkottai and Srikant [15] have proposed scheduling policies that minimize the number of packets that exceed their delay bounds.

Hou el al [4] have proposed a model that jointly considers the per-packet delay bound and the per-flow throughput requirement. This model has been extended to consider variable-bit-rate traffic [5], fading wireless channels [6] [9], the mixture of real-time and non-real-time traffic [8], and

---

This material is based upon work partially supported by NSF under Contract Nos. CNS-1302182 , CPS-1239116, CPS-1232601, CPS- 1232602 and ECCS-1150944, and AFOSR under Contract No. FA 9550-13-1-0008.

MobiHoc'13, July 29–August 1, 2013, Bangalore, India.

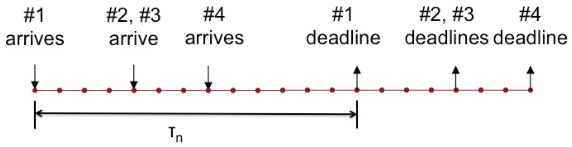


Figure 1: Packet arrivals and deadlines of a stream, with sequence numbers indicated.

multi-hop wireless transmissions [10]. However, these studies assume that all flows are synchronized and generate packets at the same time, and the results depend critically on this assumption. Moreover, they assume that all flows start playback immediately without buffering any packets, which is a critical feature of the playback process. Dutta el al [3] have studied serving video streams when receivers may buffer packets before playing them. They assume, however, that all packets are available at the server when the system starts, which is applicable to on-demand videos, but not to live videos.

3. SYSTEM MODEL

Consider a system with N wireless clients, $\{1, 2, \dots, N\}$, and one access point (AP). Each client subscribes to a live video stream through the AP.

Video streams generate variable-bit-rate traffic, and the exact number of packets generated in a time slot is influenced by not only the video encoding mechanism, but also the context of the current frame. We suppose that the number of packets generated by each flow in a time slot is described by some irreducible finite-state Markov chain, and is bounded. However, we do not assume that either the AP or the clients know this Markov chain model. We only suppose that each client n knows the long-term average number of packets generated by its subscribed video stream, which we denote by r_n. In each time slot the access point can transmit one such packet, including all overhead.

To smooth the bursty arrival process as well as channel unreliability, each client buffers incoming packets before it commences playing the video. To be more specific, a video frame that is generated at time slot t for client n will be played by the client at time slot $t+\tau_n$, with the AP informed of the value of τ_n. This is equivalent to specifying a hard per-packet delay bound of τ_n time slots for the video stream of client n. Fig. 1 illustrates an example of packet arrivals and their respective deadlines. If a packet cannot be delivered within its delay bound, it is dropped from the system. Packet drops result in glitches in the live videos, and hence need to be limited.

We model the channel reliability between the AP and client n by a probability p_n. When the AP schedules a transmission for client n, the transmission is successful with probability p_n. The AP obtains immediate feedback on whether a transmission is successfully received by ACKs. A transmission is considered to be successful if both the packet and ACK are correctly received. When a transmission fails, the AP may retransmit the packet, if the packet has not already expired, i.e., exceeded its delay bound, or it may schedule a different packet in the next time slot.

For enforcing a minimum quality for each video, each client n requires that its throughput is at least q_n packets per time slot; that is, it requires that $\liminf_{T\to\infty} \frac{\sum_{t=1}^{T} e_n(t)}{T} \geq q_n$, where $e_n(t)$ is the indicator function that a packet for client n is delivered in time slot t.

4. EARLIEST POSITIVE-DEBT DEADLINE POLICY AND ITS OPTIMALITY

In this section we establish the optimality of a simple online scheduling policy called the Earliest Positive-Debt Deadline First (EPDF). We will show that it can support every vector of throughputs that is strictly in the capacity region.

Denote by $w_n := \frac{q_n}{p_n}$ the implied workload of client n. As noted in [4], the throughput of client n is at least q_n packets per time slot if and only if the AP, on average, schedules client n for w_n times per time slot. The EPDF policy is then based on the concept of *truncated time debt*, which differs from the debt in [4]. The choice of M below is related to the usage of a multistep negative drift of the Lyapunov function.

DEFINITION 1. *The* truncated time debt *of client n at time t, denoted by $d_n(t)$, is defined recursively by:*

$$\begin{aligned} d_n(t+1) = &[d_n(t) + M w_n 1(t+1 \equiv 1 \pmod{M}) \\ &- 1(\textit{the AP schedules } n \textit{ in time t+1})]^+, \end{aligned}$$

where $1(\cdot)$ is the indicator function.

Note that it increases by $M w_n$ every M time slots, where M is an adjustable parameter, and decreases by 1 in each time slot that the AP schedules client n for transmission. Finally, we only retain the positive part. We will hereafter say that the M time slots form a *frame*. The choice of M will be discussed in the sequel. It will be the single parameter that will be tuned by the policy to adapt to all the above uncertainties.

We now present the scheduling policy, which we call the *Earliest Positive-Debt Deadline First* (EPDF) policy: In each time slot, the AP schedules the packet with the earliest deadline from those whose associated clients have strictly positive truncated time debts, that is, $d_n(t) > 0$. Ties are broken arbitrarily. If the associated client of every packet in the system has $d_n(t) = 0$, the AP schedules the packet with the earliest deadline. The AP only idles in a time slot when there are no packets to be transmitted. We note that this policy differs from the EDF policy in that it restricts the AP from serving clients with zero truncated time debts, so as to prevent the AP from providing too much service to some clients while starving others.

The EPDF policy only needs to be tuned with an appropriate choice of M. It does not need any explicit information on the traffic pattern of each client besides the knowledge of the implied workload and the choice of M. It is a simple online scheduling policy that is readily implementable. We now show that this policy supports every vector of throughputs that is strictly in the capacity region through an appropriate choice of M.

The proof employs the Lyapunov function

$$L(k) := \textstyle\sum_{n=1}^{N} d_n(kM).$$

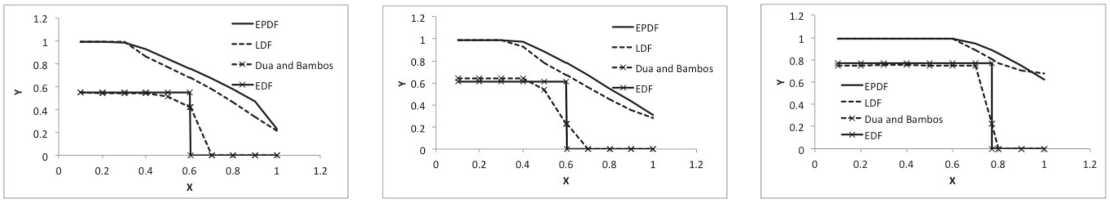
In our context the Foster-Lyapunov Theorem is used as follows:

THEOREM 1. *If, under some scheduling policy η, there exists a positive number δ, and a finite subset $\mathcal{D}_0$ of $\mathbb{R}^N$ such that:*

$$E\{L(k+1) - L(k) | [d_n(kM)]\} \leq -\delta, \textit{ if } [d_n(kM)] \notin \mathcal{D}_0,$$
$$E\{L(k+1) - L(k) | [d_n(kM)]\} < \infty, \textit{ if } [d_n(kM)] \in \mathcal{D}_0,$$

then the throughput of each client n is at least q_n.

THEOREM 2. *The EPDF policy supports every vector $[q_n]$ that is strictly in the capacity region, with properly chosen M.*



(a) Harry Potter (b) Finding Neverland (c) Transporter 2

Figure 2: Achieved throughput regions under different policies

PROOF. Due to space limitation, we only provide a sketch of the proof. Interested readers can find the complete proof in [7].

Consider a vector of throughputs $[q_n]$ that is strictly in the capacity region. There exists some $\alpha > 0$ such that the vector $[q_n + \alpha]$ is still in the capacity region. Therefore, we can assume that there exists a scheduling policy η, under which the throughput of each client n is at least $q_n + \alpha$. We then show that there exists a large enough M such that both conditions in Theorem 1 are satisfied when using η.

Next, we demonstrate that the value of $E\{L(k+1) - L(k)\}$ under the EPDF policy is smaller than that under η. Hence, the condition in Theorem 1 is also satisfied under the EPDF, and the throughput of each client n is at least q_n under the EPDF policy. ☐

In the proof of Theorem 2, the number of time slots in a frame, M, can be large, which means the truncated time debt of each client is updated infrequently. While the EPDF policy guarantees that the long-term average throughput of each client is at least as large as its requirement, a large M may lead to undesirable short-term performance. To improve short-term performance, we may want to set M to be a small number. However, the following example suggests that the EPDF policy may fail to support a vector of throughputs that is strictly in the capacity region when M is too small.

EXAMPLE 1. *Consider a system with three clients. Client 1 generates a packet in every time slot, has channel reliability* $p_1 = 1.0$*, and requires that* $\tau_1 = 1$*,* $q_1 = 0.5$*, and hence* $w_1 = 0.5$*. Client 2 generates a packet in every time slot, has channel reliability* $p_2 = 1.0$*, and requires that* $\tau_2 = 1$ *and* $q_2 = 0$*. Finally, client 3 generates a packet in time slots of the form* $4m+2$*,* $m = 0, 1, 2, \dots$*, i.e., in time slots* $\{2, 6, 10, 14, \dots\}$*. Client 3 has channel reliability* $p_3 = 0.5$*, and requires that* $\tau_3 = 2$*,* $q_3 = 3/16$*, and hence* $w_3 = 3/8 = 0.375$*.*

Suppose we set $M = 4$*. Then the EPDF policy schedules client 1 in time slots of the form* $4m + 1$ *and* $4m + 2$*, schedules client 3 in time slots of the form* $4m+3$*, and schedules client 3 in time slots of the form* $4m + 4$ *if the transmission in time slot* $4m + 3$ *fails, for all* $m = 0, 1, 2, \dots$ *It is easy to check that the EPDF policy supports the required throughput to every client.*

On the other hand, suppose we set $M = 2$*. At the beginning of time slot 1, we have* $d_1(1) = 1, d_2(1) = 0, d_3(1) = 0.75$*. The EPDF policy may schedule client 1 in time slot 1, and client 2 in time slot 2. Note that the packet of client 3 has not been generated in time slot 2, and hence cannot be scheduled in the time slot. At the beginning of time slot 3, we have* $d_1(3) = 1 - 1 + 1 = 1$*,* $d_2(3) = 0$*, and* $d_3(3) = 0.75 + 0.75 = 1.5$*. The EPDF policy then schedules client 1 in time slot 3, as its deadline is earlier than that of client 3, and schedules client 3 in time slot 4. The same schedule then repeats for all the following time slots. Hence, under the EPDF policy, client 3 is scheduled once every 4 time slots, and has a throughput of* $1/8$*, which is less than its requirement.* ☐

5. SIMULATIONS ON THREE MOVIES

We now present simulation results based on real traces of video streaming. We use the traces provided by the Video Trace Library of the Arizona State University [14,16]. We conduct our simulations on three different HD movies, namely, “Harry Potter," “Finding Neverland," and “Transporter 2."

We have implemented the EPDF policy, as well as three other policies for performance comparison, in ns-2. The three other policies are the EDF policy, which is proposed in [11], the Largest Debt First (LDF) policy proposed in [4], and a policy from Dua and Bambos [2].

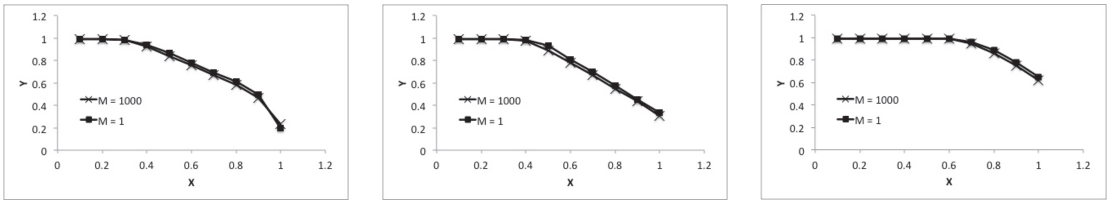
We consider IEEE 802.11a, which supports up to 54Mbps data rate, as the MAC protocol. We assume that the system has 30 clients arranged in a 6×5 grid, where adjacent clients are separated by 170 meters, and one AP that is placed at the center of the grid. We use the `Shadowing` module in ns-2 to model the unreliable wireless links. Each simulation lasts 100 seconds. The delay bounds of clients, τ_n, are assumed to be evenly distributed between 20000 – 30000 time slots.

To better illustrate the simulation results, we assume that half of the clients require a portion X of their packets to be delivered on time, while the other half of the clients require a portion Y of their packets to be delivered on time. The throughput requirements of clients are then computed accordingly. We define the *achieved throughput region* of a policy as the region composed of all (X, Y) that can be achieved by the policy. A pair (X, Y) is considered to be achieved if the resulting throughput of each client is at least 95% of its required throughput.

We first present the performance of all the four scheduling policies. Simulation results are shown in Fig. 2. It can be seen that the EPDF policy outperforms all the other three policies in all three movies. For a fixed X, the difference of achieved Y between the EPDF policy and the LDF policy can be as large as 0.13. Further, both the EDF policy and the policy in [2] result in poor performance.

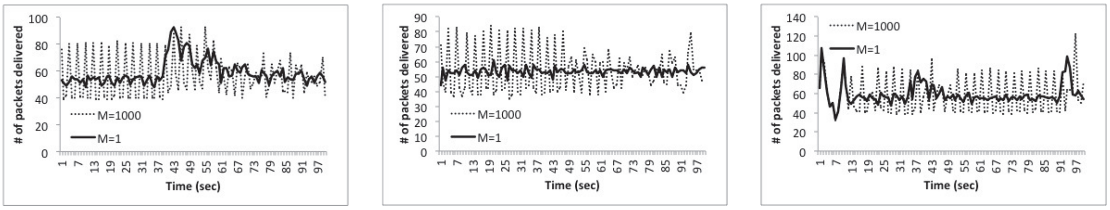
Next, we investigate the influence of frame sizes on system performance. We simulate the performance of the EPDF policy under two extreme cases: $M = 1$ and $M = 1000$. Simulation results are shown in Fig. 3. Surprisingly, the achieved throughput regions for the two cases are virtually indistinguishable. For any movie and any fixed X, the difference between the achievable Y is less than 0.03 under the two cases.

We further investigate the short-term performance under both cases of $M = 1$ and $M = 1000$. We assume that all



(a) Harry Potter (b) Finding Neverland (c) Transporter 2

Figure 3: Achieved throughput regions for different frame sizes.



(a) Harry Potter (b) Finding Neverland (c) Transporter 2

Figure 4: Short-term throughputs for different frame sizes.

clients require a portion X of their packets to be delivered on time. The value of X is set to be 0.65 for Harry Potter, 0.7 for Finding Neverland, and 0.8 for Transporter 2. For each movie, we find the client that generates the most packets during the simulation, and then plot the number of packets that this client receives in every second of the simulations.

Simulation results are shown in Fig. 4. The number of packets delivered to the client in each second varies much from second to second when $M = 1000$. On the other hand, when $M = 1$, the number of packets delivered to the client does not vary as much. Although both $M = 1000$ and $M = 1$ result in similar long-term throughputs for each client, smaller M can lead to much better short-term performance.

6. CONCLUSIONS

We have studied the problem of serving multiple live video streams to wireless clients with various delay and throughput requirements, as well as heterogeneous link reliabilities. We have proposed a simple scheduling policy, the EPDF policy, and proved that it is able to support every vector of throughputs that is strictly in the capacity region. The utility of the EPDF policy has been demonstrated through trace-based simulations, where it is seen that it appears to outperform other policies by a large margin.

7. REFERENCES

[1] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016.

[2] Dua, A., and Bambos, N. Downlink wireless packet scheduling with deadlines. *IEEE Transactions on Mobile Computing 6*, 12 (Dec. 2007).

[3] Dutta, P., Seetharam, A., Arya, V., Chetlur, M., Kalyanaraman, S., and Kurose, J. On managing quality of experience of multiple video streams in wireless networks. In *Proc. of IEEE INFOCOM* (Mar. 2012).

[4] Hou, I.-H., Borkar, V., and Kumar, P. A theory of QoS in wireless. In *Proc. of IEEE INFOCOM* (2009).

[5] Hou, I.-H., and Kumar, P. Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels. In *Proc. of ACM MobiHoc* (2009).

[6] Hou, I.-H., and Kumar, P. Scheduling heterogeneous real-time traffic over fading wireless channels. In *Proc. of IEEE INFOCOM* (2010).

[7] Hou, I.-H., and Singh, R. Capacity and scheduling of access points for multiple live video streams. arXiv:1306.2360.

[8] Jaramillo, J., and Srikant, R. Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic. In *Proc. of IEEE INFOCOM* (Mar. 2010).

[9] Jaramillo, J., Srikant, R., and Ying, L. Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints. *IEEE Journal on Selected Areas in Communications 29*, 5 (May 2011).

[10] Li, R., and Eryilmaz, A. Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multi-hop networks. In *Proc. of IEEE INFOCOM* (Apr. 2011).

[11] Liu, C. L., and Layland, J. W. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM 20*, 1 (1973).

[12] Liu, Q., Wang, X., and Giannakis, G. A cross-layer scheduling algorithm with qos support in wireless networks. *IEEE Transactions on Vehicular Technology 55*, 3 (May 2006).

[13] Raghunathan, V., Borkar, V., Cao, M., and Kumar, P. Index policies for real-time multicast scheduling for wireless broadcast systems. In *Proc. of IEEE INFOCOM* (2008).

[14] Seeling, P., and Reisslein, M. Video transport evaluation with H.264 video traces. *IEEE Communications Surveys and Tutorials, in print* (2012). Traces available at `trace.eas.asu.edu`.

[15] Shakkottai, S., and Srikant, R. Scheduling real-time traffic with deadlines over a wireless channel. *Wireless Networks 8*, 1 (2002).

[16] Van der Auwera, G., David, P., and Reisslein, M. Traffic and quality characterization of single-layer video streams encoded with the h.264/mpeg-4 advanced video coding standard and scalable video coding extension. *IEEE Transactions on Broadcasting 54*, 3 (Sep. 2008).