1-2004

Size Matters: Size-based Scheduling for MPEG-4 over Wireless Channels

Rahul Mangharam
University of Pennsylvania, rahulm@seas.upenn.edu

Mustafa Demirhan
Rutgers University - New Brunswick/Piscataway

Ragunathan Rajkumar
Carnegie Mellon University

Dipankar Raychaudhuri
Rutgers University - New Brunswick/Piscataway

Suggested Citation:
R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri, Size matters: Size-based scheduling for MPEG-4 over wireless channels. SPIE & ACM Proceedings in Multimedia Computing and Networking. Vol. 3020 pp. 110-122. San Jose, CA, January 2004.

Size Matters: Size-based Scheduling for MPEG-4 over Wireless Channels

Abstract

For bursty traffic with a large peak-to-average ratio and a stochastic channel, is it possible to minimize the response time of every flow while maximizing the effective channel utilization and maintain fairness? This is the question we address in this paper. In wireless networks with a single shared channel, channel arbitration is a core issue for flows with throughput and timeliness requirements on the uplink and peer-to-peer links where the instantaneous demand is not known. This paper presents a link layer frame scheduling algorithm for delay-sensitive variable bit rate traffic, such as high-rate multimedia (MPEG-4), over a wireless channel. We evaluate our scheduling algorithm over two Medium Access Control (MAC) architectures and compare it to four scheduling strategies that cover a range of classes: TDMA, proportional share algorithms, real-time scheduling algorithms, and size-based scheduling algorithms. Detailed simulation results, with full-length MPEG-4 movie traces over a fading wireless channel, show that Fair-Shortest Remaining Processing Time (Fair-SRPT) outperforms other algorithms in terms of QoS performance, channel utilization efficiency and response time under all utilization levels and channel error rates. Our Fair-SRPT scheme avoids the classical SRPT problems of preferring small jobs by using normalization to mean reservations. An attractive feature of the proposed approach is that it can be implemented with no modifications to the IEEE 802.11e and IEEE 802.15.3 high-rate personal area network standards.

Keywords

Scheduling Algorithms for Embedded Wireless Networks

Disciplines

Electrical and Computer Engineering | Engineering

Comments

Suggested Citation:

R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri, Size matters: Size-based scheduling for MPEG-4 over wireless channels. SPIE & ACM Proceedings in Multimedia Computing and Networking. Vol. 3020 pp. 110-122. San Jose, CA, January 2004.

http://dx.doi.org/10.1117/12.538820

Size Matters:
Size-based Scheduling for MPEG-4 over Wireless Channels

Rahul Mangharam[+], Mustafa Demirhan[*], Ragunathan Rajkumar[+], Dipankar Raychaudhuri[*]

[+]Real-time and Multimedia Systems Laboratory
Carnegie Mellon University
Pittsburgh, PA
{rahulm, raj}@ece.cmu.edu

[*]Wireless Information Network Laboratory
Rutgers University
Piscataway, NJ
{demirhan, ray}@winlab.rutgers.edu

Abstract –
For bursty traffic with a large peak-to-average ratio and a stochastic channel, is it possible to minimize the response time of every flow while maximizing the effective channel utilization and maintain fairness? This is the question we address in this paper. In wireless networks with a single shared channel, channel arbitration is a core issue for flows with throughput and timeliness requirements on the uplink and peer-to-peer links where the instantaneous demand is not known. This paper presents a link layer frame scheduling algorithm for delay-sensitive variable bit rate traffic, such as high-rate multimedia (MPEG-4), over a wireless channel. We evaluate our scheduling algorithm over two Medium Access Control (MAC) architectures and compare it to four scheduling strategies that cover a range of classes: TDMA, proportional share algorithms, real-time scheduling algorithms, and size-based scheduling algorithms. Detailed simulation results, with full-length MPEG-4 movie traces over a fading wireless channel, show that Fair-Shortest Remaining Processing Time (Fair-SRPT) outperforms other algorithms in terms of QoS performance, channel utilization efficiency and response time under all utilization levels and channel error rates. Our Fair-SRPT scheme avoids the classical SRPT problems of preferring small jobs by using normalization to mean reservations. An attractive feature of the proposed approach is that it can be implemented with no modifications to the IEEE 802.11e and IEEE 802.15.3 high-rate personal area network standards.

Keywords: Scheduling algorithms, shortest remaining processing time, wireless medium access control, QoS, link layer protocol, MPEG-4.

I. INTRODUCTION

A. Motivation for Wireless Link Layer Scheduling

The wireless link is considered a bottlenecked resource due to the difficulty in effectively allocating the shared resource to provide service guarantees and its relative low data rate. In networks with a shared wireless channel, link arbitration is a core issue for flows with Quality of Service (QoS) requirements for the uplink and peer-to-peer links. QoS is defined as the ability of the system to maintain timeliness guarantees for frames delivered over a shared link. Whether the wireless link is at the edge of the network or between multiple hops in an ad hoc network, the common link resource allocation challenges are due to the stochastic character of the channel, the network being interference-dominated and the bursty nature of multimedia traffic.

In this paper we propose a simple and efficient link layer frame scheduling algorithm to deliver timeliness guarantees for variable bit rate (VBR) traffic in general, and multimedia MPEG-4 traffic in particular, in a wireless network with a centralized medium access controller (MAC). The centralized controller or access point (AP) enjoys privileged access to the channel and is responsible for allocating medium access opportunities to every associated flow. The scheduling algorithm in the AP arbitrates which flow accesses the medium when, for how long and on which logical/physical channels. The uplink and peer-to-peer links pose a harder problem than the downlink as the flows' instantaneous throughput and delay requirements are not known by the AP. Our solution for uplink and peer-to-peer flows is therefore applicable to down link flows.

Our approach is to schedule frames at the link layer only, independent of the details of the application layer and wireless channel. While there has been extensive theoretical research on wireless channel estimation between a node pair by tracking the channel using feedback [20] between the sender and receiver, using channel side information [21], or opportunistic scheduling with channel estimation [22], these techniques have practical limitations. For multimedia traffic with arbitrary frame rates, the fidelity of the feedback diminishes when the frame interval is large (30ms) compared to the channel variations. The gains of opportunistic scheduling are limited by the stringent latency requirements of MPEG-4 traffic resulting in a smaller time scale over which the users with a good or bad channel have to be scheduled in. Furthermore, "channel averaging" or "water pouring" techniques using transmission rate and power adaptation to maximize network throughput require complex and high cost decoder designs [21], do not work effectively under a delay bound [22], require frequent two-way packet exchange and need a large number of users to extract effective gains from multi-user diversity [20]. Similar arguments may be made for the AP's knowledge of the instantaneous per-flow throughput/delay requirements for uplink and peer-to-peer links given the large dynamic range of first and second order statistics for different rate encodings of MPEG-4 traffic [6]. We therefore choose to demonstrate the practical performance of our scheduling scheme over a fading channel but not use channel estimation or application aware properties as inputs to the scheduler. A key insight of this paper is that the proposed algorithm, Fair-SRPT, lends itself naturally to perform well under fading channel conditions and bursty traffic.

The focus of frame scheduling is on the particular cases of frames sent from a node to its associated AP (uplink) and also from one node directly to another and not via the commonly associated AP (peer-to-peer), as in Fig. 1. As all nodes are assumed to be within the transmission range of the AP, the AP is required to schedule peer-to-peer communication so that other nodes may not be scheduled concurrently. To demonstrate the mechanisms and performance of our channel arbitration scheme, we focus our study to QoS support for high-rate MPEG-4 traffic for real-time multimedia applications such as teleconferencing, interactive gaming, and digital television over a wireless link. MPEG-4 provides efficient video coding for a range of bit rates and quality levels. Unfortunately, the efficient servicing of high-rate MPEG-4 multimedia streams is hard due to the large peak-to-average ratio (from 3 to over 20) of the frame sizes that must be delivered across the network link by a specified deadline. Furthermore, as the content is created at run-time and all flows may not originate from a common video server, we may not have the privilege of a large delay buffer. Each frame, therefore, has a deadline specified at the time of encoding and must be delivered to the receiver for decoding before that time. We also consider cases where this constraint may be relaxed.

B. Problem Statement

Our system model is as follows. We assume a shared fading wireless link with a single polling server and multiple nodes, each with bursty delay-sensitive traffic. We would like to:

- Minimize the number of frames that miss their deadlines and minimize the mean waiting time of frames.
- Ensure high effective channel utilization.
- Enforce flow isolation and fair resource distribution.

Given a set of flows $\mathbf{F} = \{F_1, F_2, F_3, \dots F_n\}$ where each flow, $F_i(C_i, T_i)$, is described in terms of its mean application-layer frame size, C_i, from an arbitrary frame size distribution, and frame inter-arrival, T_i. A flow, F_i, consists of a sequence of frames $J_{i,j}$, where $r_{i,j}$ denotes the arrival time of the j[th] job (i.e. frame) of flow F_i. We assume T_i is the minimum frame inter-arrival time between successive

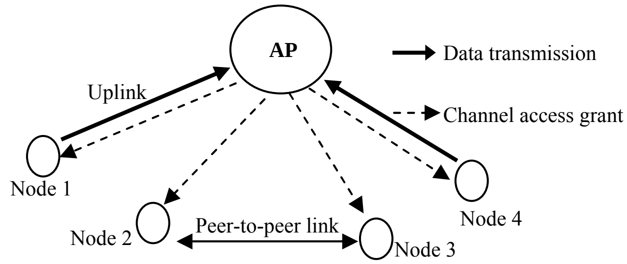


Figure 1. Centrally controlled LAN/PAN topology illustrating uplink and peer-to-peer

frames such that $r_{i,j+1} \geq r_{i,j} + T_i$. The system must deliver a frame by its absolute deadline, $d_{i,j}$, where for each frame $J_{i,j}$, $d_{i,j} = r_{i,j} + T_i$. We denote the finishing time, $f_{i,j}$, of a frame $J_{i,j}$, to be the interval between the arrival time, $r_{i,j}$, and the time at which the last symbol of the frame has been successfully delivered at the receiver.

The objective of the scheduler is to maximize $F_i(f_{i,j} \leq d_{i,j})\ \forall\ i \in \mathbf{F},\ j \in F_i$ and minimize the mean response time $R_{i,j}$ of a job $J_{i,j}$, where $TX(J_{i,j})$ is the transmission time of the j[th] frame of flow i:

$$R_{i,j} = max\{0, f_{i,j} - TX(J_{i,j}) - r_{i,j}\}$$

As a secondary goal, flow isolation ensures that an overload in one flow does not jeopardize the schedulability or adversely affect the resource reserve of other flows. Fair resource distribution for a flow guarantees that only if the current resource demand (current frame size) is not greater than the resource reserve, the flow will be allocated the resources demanded. Otherwise, the scheduling algorithm determines the resources allocated to the flow.

C. Organization

The rest of the paper is organized as follows. Section two introduces the network model in terms of the topology, traffic, channel model and MACs for channel access. A description of the proposed scheduling scheme and four other algorithms are provided in similar terms in Section three. Section four presents and discusses the simulation results. Section five covers related work followed by our conclusions.

II. Network Model

In this section, we present our model of the network.

A. MAC Protocols and Network Topology

The MAC is the lower half of the link layer responsible for reliable frame delivery across the link by means of acquiring exclusive access to the shared channel. Channel access over wireless links is performed by two general mechanisms: contention mode where all nodes in the network are peers and must compete with every other node and contention-free mode where a node is designated as the polling coordinator or AP and polls clients to give them channel access grants. In our analysis, we do not consider MAC service in the contention mode as it has been shown

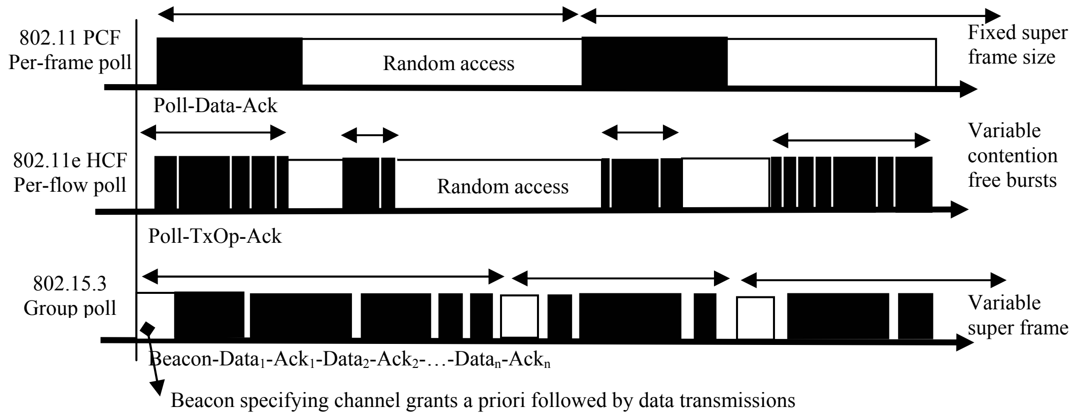


Figure 2. Contention and contention free channel access mechanisms

to not support tight QoS guarantees and distributed service differentiation suffers from unpredictable delays and unfairness [1], [2]. In our study, we employ two types of contention free modes to provide channel access grants: per-flow polling and group polling.

As in Fig. 2, the IEEE 802.11 [3] specification supports per-frame polling where a channel access grant must be sent for every uplink frame. In order to reduce the overhead due to polling, the IEEE 802.11e [4] specification enables the AP to grant nodes channel access for a fixed duration of time or transmit opportunity (TxOp) for a contention free burst consisting of one or more frames. Unlike 802.11, the 802.11e Hybrid Coordination Function supports a variable length super frame consisting of adjacent contention and contention-free periods making it adaptable to varying channel demands. The IEEE 802.15.3 [5] specification for high-rate personal area networks concatenates the channel access grants in a beacon that is broadcasted at the start of every super frame. The beacon specifies the source-destination pairs and the start and duration of their respective TxOps.

Our simulation model implements the essential functions of the 802.15.3 and 802.11e specifications with beaconing, polling, TxOp assignment, uplink, downlink and peer-to-peer frame exchange, fragmentation, frame retransmission and variable super frame sizing. All nodes in the network are immobile and can hear and interfere with each other.

The scheduling algorithms are implemented within the access point as shown in Figure 3. Based on the status of the client, in terms of its current queue size and current frame deadline that are included in the MAC header, each scheduling algorithm periodically computes and broadcasts a channel access grant table as per the MAC specifications. For this paper, we present performance results only for 802.15.3 MAC. The results for the 802.11e MAC are similar though with a slight decrease in the effective channel utilization due to the additional per-flow polling overhead [19]. At the start of every super frame, the scheduler broadcasts the TxOp grants for the flows and therefore executes scheduling decisions once every super frame. The super frame size is variable such that if the scheduling algorithm serviced all eligible flows before the maximum super frame duration, the AP polls every client in round-robin fashion to query their queue size.

B. Traffic

We consider both constant bit rate (CBR) and VBR delay-sensitive traffic. VBR traffic consists of MPEG-4 flows. We primarily use full-length videos [6] with a frame rate of 30 frames per second. In addition, a TES-based MPEG-4 traffic generator [7] which generates traffic that has the same first and second order statistics as an original MPEG-4 trace is used. All fragmentation is done at the link layer and if a frame is not completely delivered to the receiver by its deadline, it is dropped. All applications employ UDP over IP.

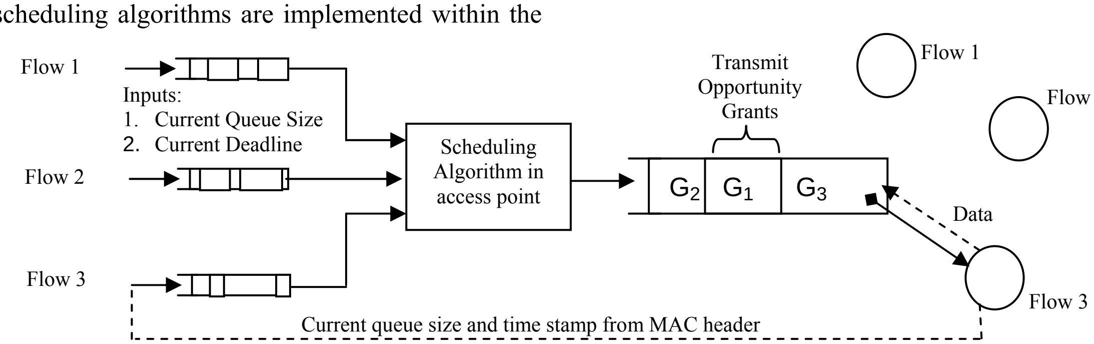
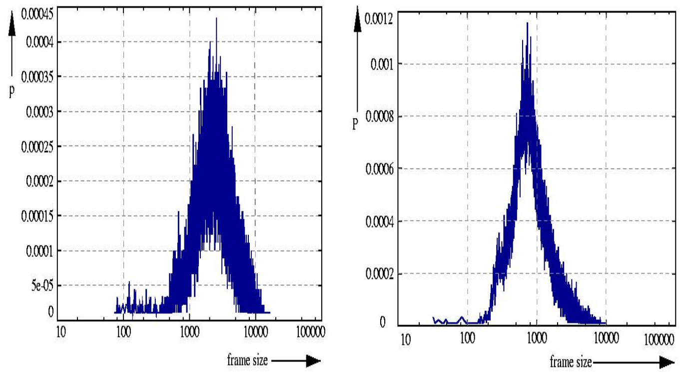


Figure 3. Access point with scheduling algorithm assigning transmit opportunities to associated flows



Frame statistics 1 Hour sample	"Jurassic Park"	"The Firm"
Compression ratio	9.92	25.93
Number of frames	89998	89998
Mean frame size	3.8 KB	1.5 KB
Variance of frame size	5.1e+06	1.3e+06
Minimum frame size	72 Bytes	32 Bytes
Maximum frame size	16745 Bytes	10204 Bytes
Mean bit rate	770 Kbps	2.9 Kbps
Peak bit rate	3.3 Mbps	2 Mbps
Peak/Mean bit rate	**4.37**	**6.96**

Figure 4. Example of frame size distributions of actual MPEG-4 movie traces. Source [6]

MPEG-4 [6] provides efficient and scalable video coding. An MPEG encoder generates three types of frames: Intra-coded (*I*), Predictive (*P*), and Bi-directional (*B*) frames. In general, I frames contain the bulk of the audio and video data and are larger then *P* frames, which in turn are larger than *B* frames. When compressing a video sequence, typical MPEG encoders use a pre-defined group of pictures (GOP), such as *I-B-B-P-B-B-P-B-B-P-B-B,* as used in our video traces. However, as our scheduling algorithms do not use application-layer information, flows can have different GOP patterns.

B. Wireless Channel Model

To model a slow fading Rayleigh channel, we use a two state Markov process for block errors with memory. In [8], it is shown that the first and second order statistics of a block fading process at the frame level are well approximated by means of a binary Markov process, which corresponds to a Gilbert channel model [9]. The state transition probabilities *P* and *p* and burst error probability *h*, as shown in Figure 5, of our wireless LAN channel model are based on the burst error length and the error-free interval distributions derived in [23]. It has been shown to closely match the first and second order statistical distribution of bit errors of a waveform simulation considering 802.11 link variables including the channel, noise, modem, coding, equalization, etc.

In the simulation model, all frame errors are assumed to be symmetric (same distribution and rate) on both directions of traffic flow. The mean error rate derived is a link layer frame error rate after the application of physical layer error resilience techniques such as error correction, source coding, equalization and BPSK modulation. We consider the channel to be in a state of outage if the frame arrives at the link layer with at least a single bit error.

III. Shared Resource Scheduling

In order to effectively describe and contrast different VBR scheduling schemes, we adopt a general bandwidth-preserving server model as outlined in [10]. Each flow is defined by a server, which, based on its scheduling scheme, decides when and for how long the flow may be serviced.

As each frame of a given flow arrives periodically, it is partially defined by a periodic server (p_s, e_s), where p_s is the server *period* and e_s is its execution time or execution *budget*. The ratio $u_s = e_s/p_s$ is the server's *utilization*. A server is *backlogged* whenever its queue is nonempty and there is a job waiting to be executed by the server. The server is *eligible* for execution when it is backlogged and has a nonzero budget. When a server executes a job, its budget is consumed at the rate of one per unit time. The budget is *exhausted* when it becomes zero. When the budget is incremented by the scheduler, up to a maximum of e_s, it is said to be *replenished* and the instance is the *replenishment time*.

Different kinds of scheduling policies are distinguished by their consumption and replenishment rules. The scheduler manages the consumption of the server budget and decides if and when to suspend a server when its budget is exhausted or it becomes idle. Once the server becomes eligible (due to replenishment of the budget or if it becomes

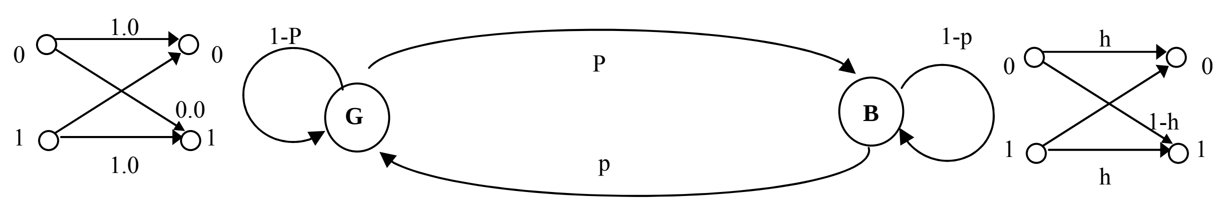


Figure 5. A 2-state Gilbert Markov channel model with a good state (G) and bad state or error burst state (B). The probability of error is zero in G and 1-h in B. For suitably small values of the transition probabilities p = Prob(B↔G) and P = Prob(G↔B), the states B and G tend to persist and the model simulates burst errors.

backlogged again and is not exhausted), the scheduler puts the server back in the ready queue. We may interpret Figure 3 as a scheduler within an AP that manages one server per flow. The scheduling policy determines when a server is eligible and the scheduler decides in which order to service the eligible flows from the ready queue based on the backlogged frame's utilization. A scheduling policy is work-conserving if it is eligible whenever it is backlogged. In other words, if the system is idle and a server is backlogged, its budget is replenished so it may be put on the ready queue by the scheduler.

We now describe our frame scheduling policy and compare it with one non-work-conserving scheme and three aggressive work-conserving algorithms. It is important to note that the primary difference between the performances of the scheduling policies is the manner in which they reclaim idle capacity from the flows whose current data rate demand is less than their reserved utilization. We define these flows as *under-loaded* flows. Equally important, as shown in figure 6, is the method in which the idle capacity is allocated among the active flows whose current data rate demand is greater than their reserved utilization. We define these flows as *overloaded* flows. The total instantaneous idle capacity of the network is the sum of the unreserved system capacity and the instantaneous unused capacity by under-loaded flows.

A. Size-based Scheduling Algorithms and our scheme, Fair-SRPT

The idea behind scheduling tasks by discriminating them based on their processing time is well known from queuing theory [11]. The focus of this paper is on the application and performance of the Shortest Remaining Processing Time (SRPT) scheduling policy for multimedia traffic over wireless networks. Our adaptation, Fair-SRPT, for the service of heterogeneous flows with different frame size distributions, is inspired by the sojourn time distributions and optimality proofs for the queue M/G/1 derived by Schrage and Miller in 1965 [12]. By minimizing the number of outstanding requests in a system, Little's Law [11] supports the fact that SRPT minimizes the aggregate mean response time of the system. Schrage shows this is true for preemptive systems with infinite and limited number of priority levels [12]. There has, however, been considerable debate that SRPT favors smaller jobs at the cost of servicing larger jobs [13], [14], [15]. We do observe that SRPT is unfair with heterogeneous flows that have different frame size distributions or different mean data rates and biases flows with smaller mean data rates (as they generally have smaller frames).

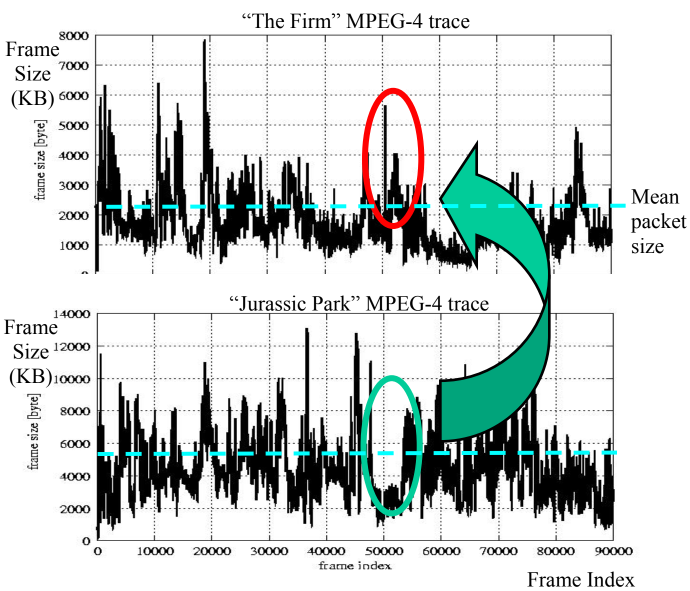


Figure 6. Example of idle capacity reclamation from under loaded flow and idle capacity allocation to an overloaded flow

Our adaptation, Fair-SRPT, remedies this problem by first normalizing all backlogged queues by their mean resource reservation and then servicing them in increasing order of their normalized queue sizes. In every super frame, we ensure that at least the mean reservation of all flows may be serviced. Furthermore, unlike "fair" sharing algorithms, a service guarantee is provided only if the current resource demand (i.e. current frame size) is not greater than the reserved resource (e.g. mean frame size). We, therefore, do not provide any guarantee at all if the current resource demand is greater than the reserve. Using this "all-or-nothing" policy, Fair-SRPT reduces the mean response time in an equitable manner by maximally servicing fully deliverable frames only. All frames are eligible for service and are not removed from the ready queue until they are completely serviced or their deadline has passed. In terms of consumption (**C**) and replenishment (**R**) rules of a bandwidth-conserving server:

- **C1** A Fair-SRPT server consumes its budget only when it executes
- **R1** Initially, the flow's reserve, $e_s = 0$.
- **R2** When a new frame arrives with execution time e to an empty queue, Normalize (e, e_s) and enqueue it into the ready queue in the order of increasing size.
- **R3** When the server successfully delivers the current frame or the deadline has passed, the job is removed from its queue. If the server is idle, replenish the budget to e_s.

For our implementation we set e_s to the mean frame size as only a single frame was buffered per-flow in the node's queue. We normalized the current queue size by using the ratio e/e_s or by $(e - e_s)/\sigma$, where σ is the variance of the frame size distribution. We did not see any significant performance difference between the two normalization methods. For flows with variable frame inter-arrival times, we ensure flow isolation by normalizing the current frame size by the residual budget, e_s', and decrementing e_s' by e. If $e_s' = 0$, we hold the frame until it is eligible.

B. Time Division Multiple Access (TDMA)

Under this non-work conserving policy, each flow gets a fixed proportion of the total available bandwidth. For example, in a network with *n* flows of equal average data rates, each flow is statically allocated 1/*n* of the channel time in a round-robin fashion. The per-flow bandwidth allocation was ensured to always be greater than or equal to the flow's average data rate. Thus, at any utilization level, the channel was always fully and proportionally allocated to all flows. If a flow's current frame was completely delivered before the end of its allotted slot, the residual TxOp is left idle. We do not use any queue size or frame deadline information as the bandwidth is proportionally and statically allocated irrespective of runtime utilization indicators.

While it may be well known that static resource reservations are not suitable for VBR traffic, we use TDMA as a base case for performance comparison to show that QoS cannot be delivered just by over-allocation of resources even at low utilization levels. We compare the performance of Fair-SRPT with that of TDMA to illustrate the importance of maintaining per-flow queue size information for idle capacity reclamation. Furthermore, we show the low sensitivity of QoS performance to the resource reservation e_s by demonstrating proportional static over-allocation of unreserved resources is inefficient and ineffective for flows with large peak-to-average frame data rates.

C. Proportional Share Algorithms (PSA)

Approximations of the Generalized Processor Sharing [16] algorithm such as Weighted Fair-Queuing have been widely employed in packet-switched networks. PSA is designed to ensure fairness among multiple servers. We compare the performance of Fair-SRPT with PSA to illustrate the detrimental effects of allocating the instantaneous idle system capacity proportionally among all overloaded flows (max-min fairness) regardless of the degree of their overload with respect to their resource reserves. In general, PSA strives to satisfy all overloaded flows by proportionally distributing the available resources while Fair-SRPT allocates the resources in the order of the most feasible job first.

A PSA server is work-conserving as its budget is replenished when it first becomes backlogged after being idle. For a backlogged queue, the budget is replenished after a job completes. Each job is assigned a finish number which is the number of the super frame in which the server budget (u_s) would be exhausted if the backlogged servers were serviced according to GPS. In terms of consumption (**C**) and replenishment (**R**) rules of a bandwidth-conserving server:

C1 Initially the server budget and finish numbers are set to zero.

C2 When the first job arrives with execution time *e*, its finish number is set to e/u_s. The system utilization, U_b, is incremented by u_s.

R1 When a job arrives at time *t* at an idle queue,

(a) Its finish number is set to $(t - t_{-1})/\ U_b$, where t_{-1} is the previous instance when the server's finish number was updated. U_b is incremented by u_s.

(b) Set $t_{-1} = t$ and U_b is incremented by u_s.

R2 When a job completes, if the server is still backlogged, its finish number is incremented by R1. If the server is idle, the finish number is incremented by $(t - t_{-1})/U_b$ and U_b is decremented by u_s.

In summary, the server distributes all resources (i.e. both reserved and idle) proportionally among all flows. We show that PSA is fair with respect to resource allocation but not in terms of QoS performance.

D. Real-Time Aperiodic Server Algorithms

Classical real-time systems guarantee timing requirements by reserving all the resources it needs for a flow's worst-case utilization. If such a guarantee cannot be made, the flow is rejected. Real-time scheduling algorithms may be classified into two categories: Fixed Priority Scheduling and Dynamic Priority Scheduling [17]. Under the former, all jobs (or frames) of the same flow are given the same priority that is proportional to the frequency of execution of the flow. During overload conditions, when the current job execution time is more than the reserve e_s, fixed-priority scheduling such as the Rate-Monotonic algorithm, schedules the overload portion in the background, delaying its completion time to an unpredictable time. Furthermore, as the worst-case least upper bound of link utilization is limited to ~69%, fixed priority scheduling is unsuitable for bursty VBR traffic over bandwidth-constrained wireless links.

Dynamic priority scheduling algorithms such as Earliest Deadline First (EDF) assign the job's priority at runtime based on its arrival time and period relative to other flows. During overload, however, EDF requires a policing scheme to maintain flow isolation to prevent overloaded flows from jeopardizing other flows' performance. Several aperiodic server extensions such as Dynamic Sporadic Server, Total Bandwidth Server and Constant Utilization Server [10], have been proposed to extend EDF to efficiently service flows with variable execution times and inter-arrival times. The Constant Bandwidth Server (CBS) [18] is a generalization of EDF and is better than other server mechanisms in that it does not require an estimate of the worst-case execution time of a flow. A CBS for a flow is defined by its utilization which is the ratio of the execution time of an average-sized frame and the mean inter-arrival time between frames. CBS maintains the following consumption and replenishment rules where U_s is the server utilization (e_s/t_s), e_s is the maximum server budget, e_s' is the residual current budget, $d_{s,k}$ is the current deadline:

The server's execution budget is consumed at a rate of one per unit time when:

C1 The server is executing.

Replenishment rules for CBS:

R1 Initially, $e_s = 0$ and deadline, $d_{s,0} = 0$.

R2 When a new frame arrives at time $r_{i,j}$ to an empty queue,
if $(e_s' \geq (d_{s,k} - r_{i,j}))*U_s$
then $d_{s,k+1} = d_{s,k} + p_s$ and $e_s' = e_s$
else job is served with last server deadline $d_{s,k}$ and current budget e_s'

R3 The job is removed from its queue when the current frame is successfully delivered or the deadline has passed.

(a) If the server is backlogged, the server admits the next job and the deadline is determined by R2.

(b) If the server is idle, set $d_{s,k} = t + p$, where t is the current time, and replenish the budget to $e_s' = e_s$

R4 If the server budget $e_s' = 0$, it is recharged to e_s and the new server deadline is determined by $d_{s,k+1} = d_{s,k} + p_s$.

Rule R2 ensures that the resource allocation to the server is never greater than the server utilization bound U_s. R4 ensures that there are no finite intervals when the server's budget is zero. By deferring the deadline during an overload, the server is still eligible and can reclaim idle capacity fairly and efficiently according to EDF while maintaining flow isolation.

We compare the performance of Fair-SRPT with CBS to show the detrimental effects of allocating the idle time fairly among all overloaded flows according to EDF (R2). We demonstrate the ill effects of deferring the deadline based on historic resource consumption (R2 and R3a). Furthermore, we show that being "size-aware" is more useful than a "deadline-aware" scheduling scheme for VBR traffic with large peak-to-average data rate requirements.

IV. Performance Analysis

A. Assumptions and Simulation Model

The 802.11e HCF and 802.15.3 MAC were implemented within ns2 network simulator. Simulations were executed for real MPEG-4 movie traces such as *Jurassic Park*, *Star Wars IV*, *The Firm* and *Silence of the Lambs* of one hour durations and with average data rates varying from 4Mbps to 18Mbps and peak-to-average data rates between 4 and 12. All scheduling decisions were made at the start of a super frame for 802.15.3 or at the start of a per-flow TxOp for 802.11e. The super frame size was fixed to a maximum of 8ms for both MACs as this was found to be the sweet spot between service latency and protocol overhead and delivered the best overall results for frames with 30ms inter-arrival times. For all scheduling algorithms, the super frame was set to $\Sigma e_{s,i}$ ($0 \leq i \leq n$) for n flows to guarantee that the reserved resource was always available. MAC/PHY overheads are listed in Table 1 as per the 802.11a [4] and 802.15.3 specifications.

Table 1. MAC and PHY parameters used in simulation models.

MAC & PHY Characteristics	802.15.3 100Mbps	54Mbps 802.11e
Slot Time	5μs	9μs
TxOp and Poll Guard Time	10μs	16μs
First TxOp Gap	100μs	-
Short Inter-Frame Space (SIFS)	10μs	16μs
DCF Inter-Frame Space (DIFS)	20μs	26μs
PHY Header Length	15μs	20μs
MAC Header Length	16μs	16μs
Max. MAC Fragment Length	2048 bytes	2048 bytes

The only application information used by the link schedulers is the mean and variance of the frame size distribution. As the frame inter-arrival time was 30ms for all flows, we observed that the start time of the flows affected the performance. The mean results presented are averaged over 90,000 frames each and over three worst-case start time separations (0ms, 9ms, and 17ms for a 8ms super frame) for each flow. The protocol overhead for 802.15.3 is approximately 24% of the flow's average data rate.

B. Network QoS

We use three metrics to effectively evaluate and contrast the per-flow performance with each scheduling algorithm:

- The job failure rate (JFR) or deadline miss rate which is the ratio of frames that were not successfully received by the receiver within the frame deadline to the total number of frames sent from the application.
- The size of the successfully delivered frame
- The mean response time of frame delivery: the interval elapsed between the moment of arrival and the instant the first symbol of the frame is received at the receiver. The response time does not include the time for transmission of the frame and is therefore the time spent in the sender's queue waiting to be serviced.

We ran a range of experiments across all channel utilization levels and a practical range of frame error rates due to the fading channel. These metrics provide insight into the significant performance impact by the subtle distinctions in idle capacity allocation to overloaded flows across the different schemes.

1. Performance over error-free channel

We first evaluate the performance of each scheduling algorithm over an error-free 100Mbps channel to isolate the impact of bursty VBR traffic. Figure 7 shows the variation of the job failure rate as the channel utilization, in terms of the number of 4Mbps uplink MPEG-4 flows, increases. All the flows are of the same video file and the results are averaged over four fixed start time intervals between each flow. We observe that Fair-SRPT outperforms TDMA over all utilization levels by a factor of 20 to 5. For moderate to high loads between 70%-100% channel utilization, Fair-SRPT outperforms PSA and CBS by 100% to 350%. We

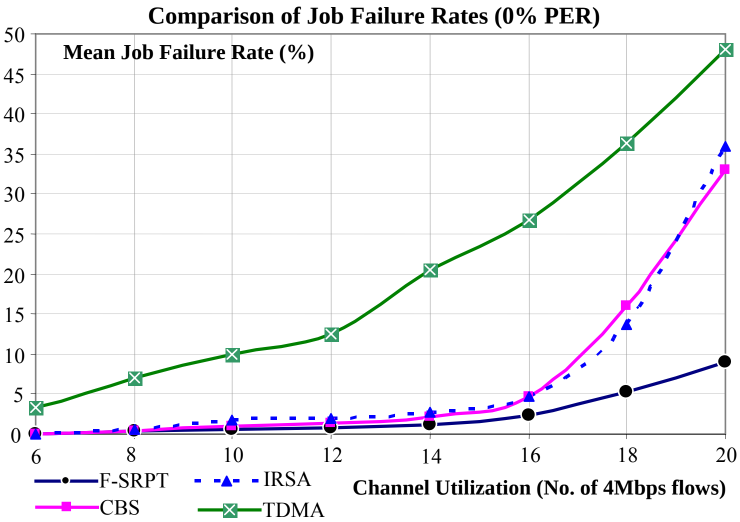


Figure 7. JFR comparison of Fair-SRPT with TDMA, PSA & CBS

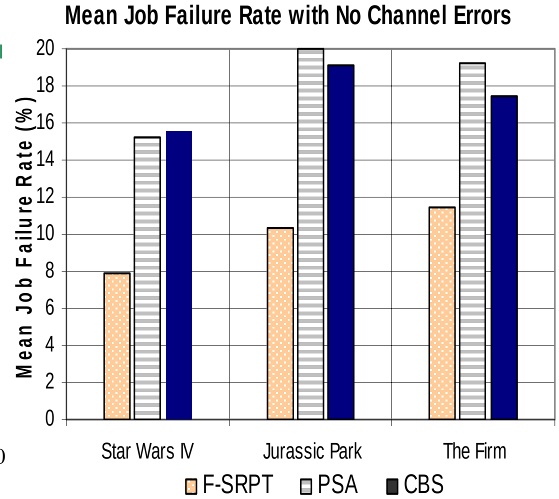


Figure 8. JFR for 14Mbps full-length MPEG-4 traces

observe that under all utilization levels, the JFR of Fair-SRPT does not cross the 10% threshold.

In Figure 8, we view the performance of three 14Mbps full-length movies with different frame size distributions. For all flows, the JFR for PSA and CBS is almost twice as much as the JFR for Fair-SRPT.

2. *Why Fair-SRPT outperforms other algorithms*

In order to understand the significant performance improvements obtained by Fair-SRPT, consider the simple example in Table 2. Given a 100Mbps channel and five 20Mbps VBR flows, we make average data rate reservations for each flow. The third column shows the system in a state of overload where the first flow currently requires less resource than its reserve while the remaining four flows require more. The last column shows the job failure rate of the system. TDMA is able to satisfy only the first flow as it distributes the resources equally among all flows regardless of their current demand. On the other hand, Fair-SRPT is able to satisfy all but one flow. This instance shows the need for the scheduling algorithm to be aw

are and adaptive to runtime resource indicators such as frame size or frame deadline.

Both CBS and PSA reclaim the 4 units of resource not used by the first flow and divide it equally among the overloaded flows (CBS does this by deferring the deadlines of each of the overloaded flows by one period and then scheduling them using EDF). We observe that only two of the five flows are completely delivered with CBS and PSA. This example illustrates that while idle capacity reclamation is important, effective idle capacity allocation among the overloaded flows is vital.

We see this effect using a real MPEG-4 trace in Figure 9. Three 8Mbps traces were run for 15 seconds (500 frames) over the same 100Mbps channel first using TDMA and then Fair-SRPT. We recorded the size of frames successfully delivered by their deadlines for one of the flows and observe Fair-SRPT delivers significantly more large size frames in the 35KB-80KB range. It is important to note here that the mean channel utilization was approximately 30% and 33Mbps was reserved for each flow (four times more than its mean data rate) with TDMA while Fair-SRPT reserved 8Mbps for each flow. *We observe that the performance is rather insensitive to the size of reserve budget (es) given the large peak-to-average data rate.* This fact leads to two conclusions: first, over-allocation of resources for VBR traffic is not very effective or efficient even for systems under low to moderate utilization levels; Second, idle capacity reclaimed instantaneously from under-loaded flows is more important than just using the unreserved system capacity.

Figure 10 provides some intuition to illustrate the

Flow Number	Reserved Data rate	Instantaneous Data rate	Current Allocation				Current Job Success Rate	
			TDMA	CBS	PSA	F-SRPT		
1	20	16	16	16	16	16	**TDMA**	**1/5**
2	20	21	20	21*	21	21	**CBS**	**2/5**
3	20	26	20	21*	21	26	**PSA**	**2/5**
4	20	30	20	21*	21	30	**F-SRPT**	**4/5**
5	20	32	20	21*	21	7		

Table 2. A comparison of idle capacity reclamation and allocation in an overloaded system. * with extended deadline

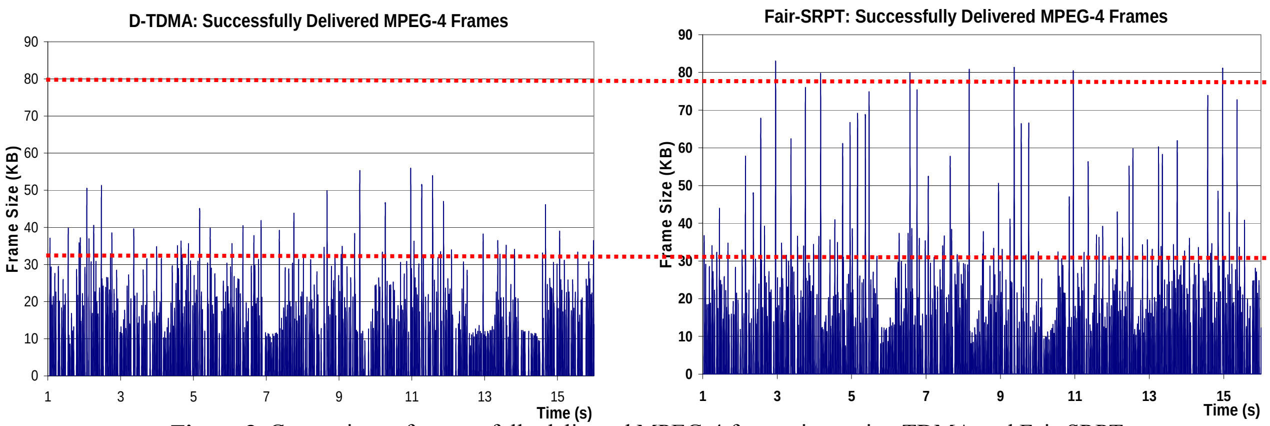


Figure 9. Comparison of successfully delivered MPEG-4 frame sizes using TDMA and Fair-SRPT

relative impact of idle capacity reclamation and allocation. The region delineated with the dashed line shows the benefit of idle capacity recovered by CBS and PSA from under-loaded flows. As with CBS, PSA attempts to fairly process all overloaded frames including the largest frames that would be eventually dropped. These partial and infeasible delivery attempts which are made in order to fairly distribute resources at all times is done at the cost of occupying channel time when some moderately overloaded flows could have successfully been delivered if it had been given the privileged use of the instantaneous idle capacity. Thus, in overload, by attempting to deliver large frames (that are eventually dropped) and maintain temporal fairness, PSA and CBS penalize the QoS of both frames within the overloaded flow and of other flows too. On the other hand, Fair-SRPT allocates idle capacity starting with the least overloaded flows (where overloads have been normalized by the average or reserved data rate). Thus, Fair-SRPT is able to maximize the number of successful frames delivered represented in the region outlined by the solid line. Fair-SRPT is still unable to service the frames in the 98-100 percentile range of frame sizes as these peak size frames often cannot be theoretically delivered across the channel since they require bandwidth in excess of the available channel capacity. *Fair-SRPT naturally determines the largest feasible set of flows based on the current load and maximum assignable idle capacity.*

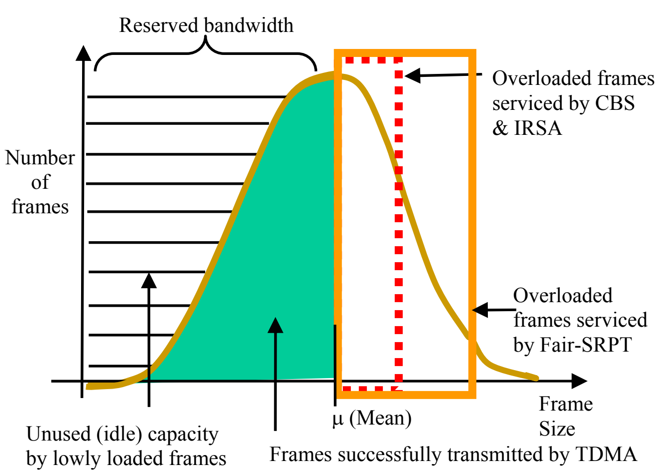


Figure 10. Range of successfully serviced frame sizes

3. Performance over a fading channel

We compare, over a realistic range of fading channel error conditions, the JFR achieved by Fair-SRPT, CBS and PSA for a moderately loaded channel with four 16Mbps MPEG-4 hour-length movie traces. In Figure 11, we observe that Fair-SRPT consistently outperforms PSA and CBS under all mean frame error rates. All values are averaged over three different start time offsets. By varying the start times we prevent PSA and EDF from behaving like round robin. The JFR suffered by CBS and PSA is consistently higher by almost 100%. Secondly, the absolute difference by which Fair-SRPT outperforms PSA and CBS increases with the error rate and link utilization.

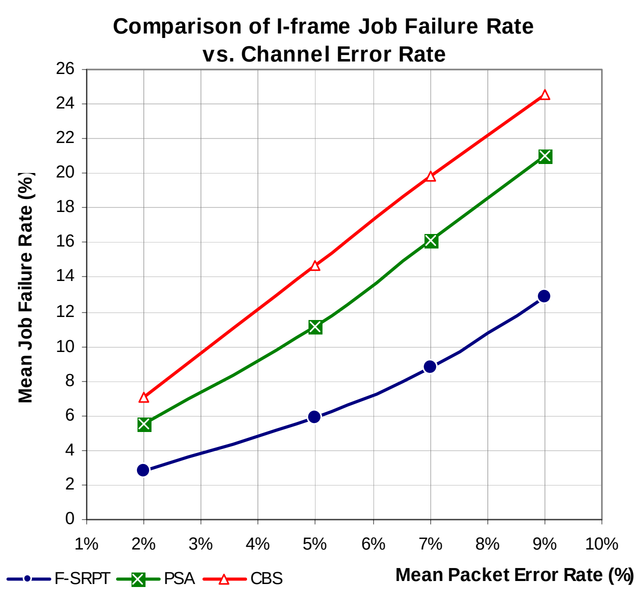


Figure 11. Comparison of JFR for a moderately loaded link over a range of mean frame error rates.

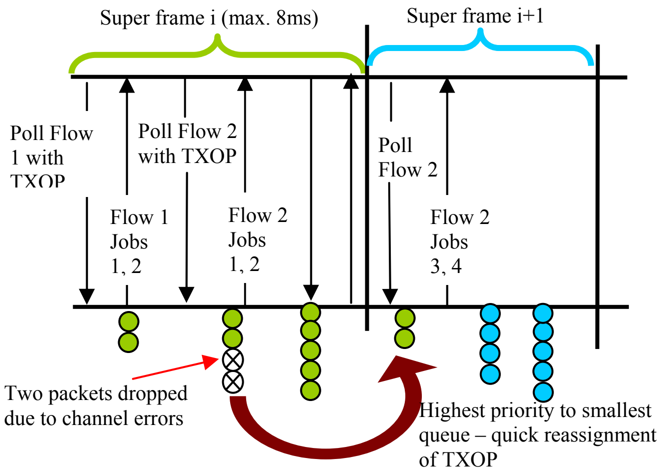


Figure 12. Fair-SRPT naturally assigns the highest priority to frame retransmits.

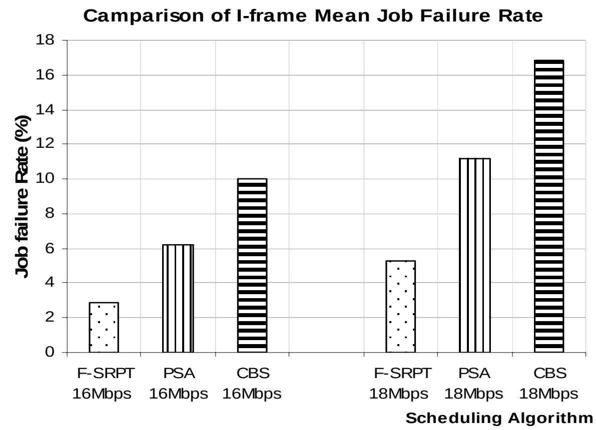


Figure 13. I-Frame JFR for hour-length movies

4. Why does Fair-SRPT outperform other algorithms over a fading channel?

For channels with high frame error rates, Fair-SRPT outperforms CBS and PS because retransmits are naturally given the higher priority within the current super frame and across super frames. For example, in Figure 11, of three flows in the network, Fair-SRPT services the least overloaded flow first. If the second flow in the first super frame suffers two frame drops due to channel errors, the channel is first assigned to that flow in the next super frame as its queue size will be the smallest. This is almost always true as the currently transmitting flow's queue is the smallest and retransmits will be favorably serviced in the following super frame. On the other hand, PSA would first allocate resources to flow 1 and then to flow 2 in the next super frame in order to maintain fairness. CBS would further defer the flow's deadline and lower its relative priority. Thus, Fair-SRPT reduces the priority inversion caused by delaying service to dropped frames.

B. User-perceived QoS

In order to evaluate the user perceived QoS, we study the impact on JFR for different frame types and the burstiness of the job failures. Flows with similar mean JFR but with different mean job failure burst sizes can have very different impact on the user experience. In general, I-frames are the most important MPEG-4 frames that contain the bulk of the video frame data [6]. The *P* frames affect only the subsequent frame and the *B* frames do not affect other frames. Both the *P* and *B* frames require the *I* frame at the beginning of the group of pictures (GOP). If an *I* frame is dropped, the subsequent eleven P and B frames are rendered useless. Therefore, in this section, we assess the impact of channel utilization and errors on I-frames.

In Figure 13, in two separate tests, we measured the mean JFR for I-frames for four 16Mbps and four 18Mbps flows (~70% and ~85% utilization) to evaluate the effect of channel utilization on user perceived QoS. We see that at both utilizations levels the I-frame JFR of Fair-SRPT is less than half that of PSA and less than third that of CBS.

In figure 14, we measured the number of I-frames dropped within 10-second intervals over one hour of MPEG-4 video. The mean link utilization was ~70% with a 7% mean frame error rate. We observe that the channel outages for CBS and PSA are significantly larger than that of Fair-SRPT. CBS has an approximately 15 I-frame outage in a 10 second interval, while PSA suffers and average of 7 frames and Fair-SRPT about 3 frames. As each I-frame drop affects 360ms of displayed content, with an average of 15 I-frames, over 5.4 seconds of content is not displayed during every 10-second interval. This is effectively a 54% user perceived JFR and essentially renders a severely unsatisfactory user experience. On the other hand, an

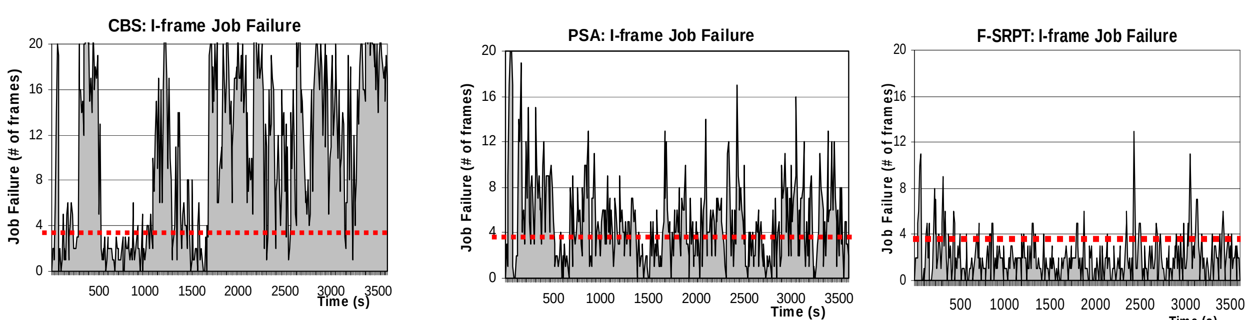


Figure 14. Number of I-frames dropped in a 10 sec interval with a 7% mean channel error rate.

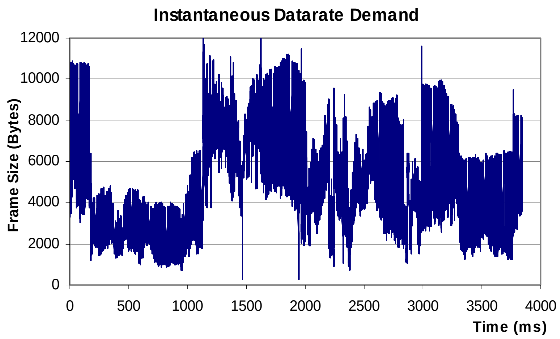


Figure 15. Snapshot of 16Mbps MPEG-4 video

average of 4 I-frame loss results in a 1.32 second (13.2%), which is reasonable given a 7% mean frame error rate.

1. Why Fair-SRPT outperforms other algorithms

In order to understand the reason for the relatively low user perceived QoS of CBS we look at a snapshot, in figure 15, of the actual frame sizes dispatched by the MPEG-4 application. In general the flow enters bursts of overload and under load that span several hundred milliseconds. This may be due to the rapid motion and change of brightness during action scenes of the movie. When a flow enters an overflow burst under CBS, its deadline is deferred so as to ensure the utilization granted to the flow is never greater than the reserved utilization. As the burst length spans several consecutive frames, the subsequent frames are adversely affected by previous overloads and suffer server deadlines well beyond their frame deadlines. This "memory" effect resulting in starvation and large outage durations is similar to the Total Bandwidth Server and the Virtual Clock Algorithm [10].

C. Flow Isolation

In order to permit well-behaved CBR flows to coexist with MPEG-4 flows that are constantly oscillating between overloaded and under-loaded states about their mean data rate, it is essential to maintain isolation between the two traffic categories. In our implementation, CBR flows were always guaranteed their required data rate in every super frame. Figure 16 shows that while the instantaneous throughput of MPEG-4 flows oscillate, the CBR flows are provided a constant throughput. We therefore do not execute any idle capacity reclamation schemes for CBR traffic and essentially ignore the queue size information from these flows.

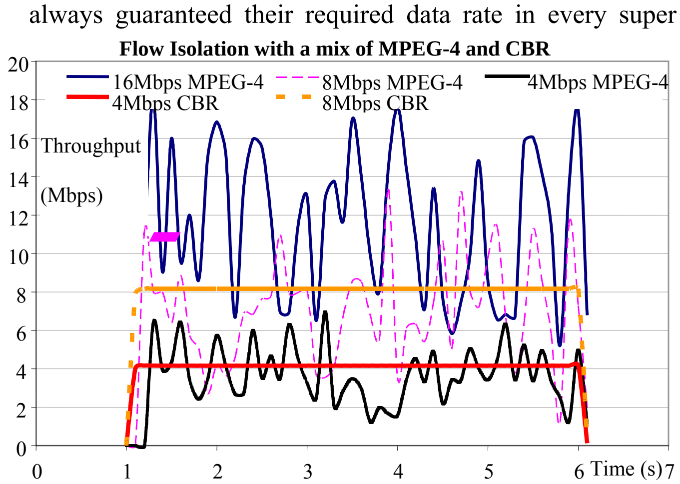


Figure 16. Flow isolation with a mix of MPEG-4 and CBR flows.

D. Performance with heterogeneous flows

In this section, we consider flows with different frame intervals and therefore different frame deadlines. We evaluate the performance of two scenarios: Four flows of the same movie but with frame intervals of 20ms, 40ms, 60ms and 80ms with mean data rates of 4Mbps, 6Mbps, 8Mbps and 16Mbps respectively. The peak-to-average of the data rate was approximately 7.6. For all three scheduling schemes, the utilization budget of each flow was set to its mean data rate and the period was set to its frame interval. In Table 3, we observe that the performance of Fair-SRPT is more evenly distributed across all flows and with a lower overall average JFR. On the other hand for CBS and PSA, the JFR for flows with smaller periods is significantly higher than that of flows with larger periods. In order to maintain resource allocation fairness across each super frame, both CBS and PSA periodically preempt the service of flows with shorter deadlines to service flows with larger deadlines.

In the second scenario, we consider flows with different frame intervals (10ms, 20ms, 40ms and 80ms) but with the same average data rate of 16Mbps. For an hour-long MPEG-4 movie trace, the JFR distribution for Fair-SRPT was more evenly distributed with a mean overall JFR of 12.1%. The JFR for PSA and CBS were 14.9% and 16.3%. We notice a higher overall JFR as the network load is higher (80% of maximum link capacity) and there is more priority inversion in all three scheduling schemes due to the larger frame sizes in flows with larger frame intervals.

It is important to note that fairness with regard to resource allocation does not necessarily lead to fairness in performance. Fair-SRPT determines the largest feasible set of flows that can be successfully delivered and is fair with respect to resource allocation only for flows that are not overloaded. For overloaded flows, the objective of Fair-SRPT is to minimize the overall JFR.

Flow and frame period	**Fair-SRPT**	**PSA**	**CBS**
Flow 1 (80ms, 4Mbps)	4.5	0	0
Flow 2 (60ms, 6Mbps)	3.9	1.4	0.8
Flow 2 (40ms, 8Mbps)	4.4	8.6	8.2
Flow 2 (20ms, 16Mbps)	1.8	7.4	7.4
Average JFR	**3.6**	**4.3**	**4.1**

Table 3. Mean job failure rates of flows with different frame intervals and mean data rates.

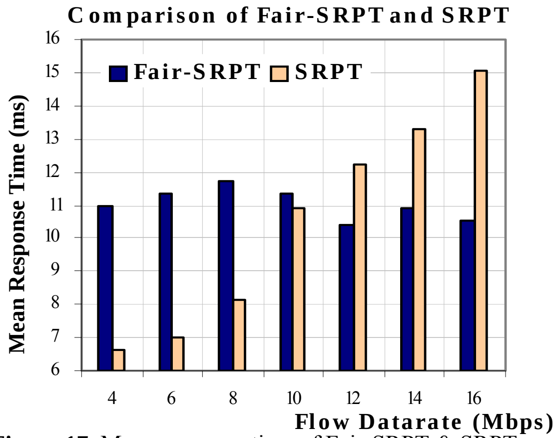


Figure 17. Mean response time of Fair-SRPT & SRPT

E. Fairness

In order to show the fairness of Fair-SRPT, we compare the mean frame response time with SRPT across a range of data rates. The channel was loaded with flows whose average data rate ranged from 4Mbps to 16Mbps and the total link utilization was ~95%. In Figure 17, we observe that all flows scheduled by Fair-SRPT experienced similar response times. Scheduling MPEG-4 traffic with SRPT, on the other hand, resulted in flows with lower average data rates to enjoy a response time almost 2.5 times lower than flows with the highest data rates. Thus, SRPT favors flows with smaller average data rates at the expense of service to higher data rate flows. This illustrates that SRPT is not fair for service of MPEG-4 traffic and the normalization of overloads incorporated in Fair-SRPT effectively enforces fairness.

V. CONCLUSION

In this paper, we have proposed two main contributions to scheduling VBR delay-sensitive traffic over wireless channels. First, we identify two key properties that must be satisfied by any scheduler of a variable capacity channel dealing with bursty traffic:

(a) The idle capacity reclamation from all under-loaded flows is more important than the unreserved capacity of the system. Greedy idle capacity allocation schemes that maximize the number of fully satisfied flows outperform schemes that fairly allocate idle capacity among all active flows.

(b) By being fair with respect to the reserved capacity and being instantaneously "unfair" in distributing the idle system capacity, network QoS and user-perceived QoS are enhanced significantly.

Our second contribution is a simple link-layer frame scheduling scheme for existing and proposed wireless MAC specifications, which ignores details of the application layer and the channel. We demonstrate its performance using full-length MPEG-4 movies over a slow fading wireless channel. Our algorithm, Fair-SRPT, avoids the traditional pitfalls of SRPT schemes and outperforms static allocation schemes such as TDMA by a factor of 5 to 20 and outperforms proportional share allocation (PSA) and CBS (a generalization of EDF) by 150-350% for link utilization between 40%-100%. Over a fading wireless channel, Fair-SRPT delivers twice as many *I*-frames than PSA and CBS enhancing user-perceived QoS by a factor of 5. We show that flow isolation is maintained and the complexity is similar to that of PSA. The only requirement for implementing the proposed scheduling scheme is the queue size of a flow, which is available in the IEEE 802.11e QoS Control Field and IEEE 802.15.3 Last Fragment Number field. Future work will include adapting Fair-SRPT to maintain QoS while minimizing the node's power consumption.

REFERENCES

1. A. Lindgren, A. Almquist, O. Schelén. "Evaluation of Quality of Service schemes for IEEE 802.11 wireless LANs," *LCN 2001*.
2. M. A. Visser, M. E. Azrki. "Voice and data transmission over 802.11 wireless network," *Proc. of PIMRC'95*, pp. 648-652, 1995.
3. The IEEE Std 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999 edition.
4. 802.11e. Draft Supplement IEEE LAN/MAN Specific Requirements - Part 11: Wireless MAC and physical layer specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), Draft 3.0, May 2002
5. 802.15.3 Draft Standard IEEE LAN/MAN - Part 15: Wireless MAC and PHY Specifications for High Rate Wireless PAN
6. F. Fitzek, M. Reisslein. "MPEG-4 & H.263 Video Traces for Network Performance Evaluation," *IEEE Network*, 15: 40-54, 2001.
7. B. Melamed. "Modeling Compressed Full-Motion Video," *Winter Simulation Conference 1997,* 1368-1374.
8. M. Zorzi, R.R. Rao, L.B. Milstein, "A Markov model for block errors on fading channels," *PIMRC'96*, pp. 1074--78, Oct. 1996.
9. E.N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.* 39, pp. 1253-1256, 1960.
10. J. W. S Liu. "Real Time Systems," *Prentice Hall*, 2000
11. L. Kleinrock, "Queuing Systems, volume II, Computer Applications," *John Wiley & Sons*, 1976.
12. L. Schrage "Proof of the optimality of the shortest remaining processing time discipline," *Operations. Research*, 678-690, 1968
13. N. Bansal, M. Harchol-Balter. "Analysis of SRPT Scheduling: Investigating Unfairness," *Proceedings of ACM Sigmetrics 2001.*
14. M. Bender, et al.. "Flow and stretch metrics for scheduling continuous job streams," *ACM-SIAM SODA*, pages 270--279, 1998.
15. A.S. Tanenbaum, "Modern Operating Systems," Prentice Hall, Upper Saddle River, NJ. 2001.
16. A. K. Parekh, R. G. Gallager. "A generalized processor sharing approach to flow control in integrated services networks: the

single-node case," *IEEE/ACM Transactions on Networking,* 1(3): 344-357. June 1993.

17. C. L. Liu , J. W. Layland. "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of the ACM*, 20(1):46--61, 1979.

18. L. Abeni & G.C. Buttazzo. "Integrating multimedia applications in hard real-time systems" In *Proc. IEEE RTSS*, Dec 1998.

19. R. Mangharam, "1-Byte Wireless QoS Solution," *Masters Thesis, Carnegie Mellon University*, 2002

20. R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multi-user communications," in *Proc. IEEE International Conference on Communications (ICC'95)*, Seattle, USA, June 1995.

21. A. J. Goldsmith and P. P. Varaiya. "Capacity of fading channels with channel side information," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1986.

22. P. Viswanath, D. N. C. Tse and R. Laroia. "Opportunistic beam-forming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp 1277-1294, June 2002.

23. J. J. Lemmon, "Wireless Link Statistical Bit Error Model," *NTIA Report 02-394*, June 2002.