February 2009

UILU-ENG-09-2202
DC-240

ADMISSION CONTROL AND SCHEDULING FOR QoS GUARANTEES FOR VARIABLE-BIT-RATE APPLICATIONS ON WIRELESS CHANNELS

I-Hong Hou and P. R. Kumar

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 2009	

4. TITLE AND SUBTITLE
Admission Control and Scheduling for QoS Guarantees for Variable-Bit-Rate Applications on Wireless Channels

5. FUNDING NUMBERS
AFOSR W911NF-08-1-0238, AFOSR W-911-NF-0710287, NSF ECCS-0701604, NSF CNS-07-21992, NSF CNS-0626584, and NSF CNS-05-19535

6. AUTHOR(S)
I-Hong Hou and P. R. Kumar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801-2307

8. PERFORMING RGANIZATION REPORT NUMBER
UILU-ENG-09-2202
DC-240

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR, 875 N. Randolph St., Arlington VA 22203
NSF, 4201 Wilson Blvd, Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy, or decision, unless so designated by other documentation

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Providing differentiated Quality of Service (QoS) over unreliable wireless channels is an important challenge for supporting several future applications. We analyze a model that has been proposed to describe the QoS requirements by four criteria: traffic pattern, channel reliability, delay bound, and throughput bound. We study this mathematical model and extend it to handle variable bit rate applications. We then obtain a sharp characterization of schedulability vis-à-vis latencies and timely throughput. Our results extend the results so that they are general enough to be applied on a wide range of wireless applications, including MPEG Variable-Bit-Rate (VBR) video streaming, VoIP with differentiated quality, and wireless sensor networks (WSN).

Two major issues concerning QoS over wireless are admission control and scheduling. Based on the model incorporating the QoS criteria, we analytically derive a necessary and sufficient condition for a set of variable bit-rate clients to be feasible. Admission control is reduced to evaluating the necessary and sufficient condition. We further analyze two scheduling policies that have been proposed, and show that they are both optimal in the sense that they can fulfill every set of clients that is feasible by some scheduling algorithms. The policies are easily implemented on the IEEE 802.11 standard. Simulation results under various settings support the theoretical study.

14. SUBJECT TERMS
QoS, wireless channels, admission control, scheduling

15. NUMBER OF PAGES
10

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Admission Control and Scheduling for QoS Guarantees for Variable-Bit-Rate Applications on Wireless Channels

I-Hong Hou
Department of Computer Science
University of Illinois
Urbana, IL, 61801, USA
ihou2@illinois.edu

P. R. Kumar
CSL and Department of ECE
University of Illinois
Urbana, IL 61801, USA
prkumar@illinois.edu

Abstract

Providing differentiated Quality of Service (QoS) over unreliable wireless channels is an important challenge for supporting several future applications. We analyze a model that has been proposed to describe the QoS requirements by four criteria: traffic pattern, channel reliability, delay bound, and throughput bound. We study this mathematical model and extend it to handle variable bit rate applications. We then obtain a sharp characterization of schedulability vis-a-vis latencies and timely throughput. Our results extend the results so that they are general enough to be applied on a wide range of wireless applications, including MPEG Variable-Bit-Rate (VBR) video streaming, VoIP with differentiated quality, and wireless sensor networks (WSN).

Two major issues concerning QoS over wireless are admission control and scheduling. Based on the model incorporating the QoS criteria, we analytically derive a necessary and sufficient condition for a set of variable bit-rate clients to be feasible. Admission control is reduced to evaluating the necessary and sufficient condition. We further analyze two scheduling policies that have been proposed, and show that they are both optimal in the sense that they can fulfill every set of clients that is feasible by some scheduling algorithms. The policies are easily implemented on the IEEE 802.11 standard. Simulation results under various settings support the theoretical study.

1 Introduction

Digital wireless communication technology has contributed to the development of several application fields, such as Wireless Local Area Networks (WLAN) and Wireless Sensor Networks (WSN). Among other research issues, we anticipate that providing QoS support will be of increasing interest due to the increasing demands of delay-sensitive data traffic. Applications that require QoS support include video streaming, VoIP, and realtime surveillance.

Essentially, QoS consists of providing guarantees on both delay and throughput for each flow in the system. Two important issues arise when providing QoS support. One is to determine whether the requirements of a set of clients exceed the capacity of the wireless network. The other is, given that the requirements of the set of clients can all be satisfied by the network, to find a scheduling policy that actually does so.

In this paper, we extend a theoretical study on supporting QoS that allows us to jointly deal with fundamental difficulties. Our extension allows us to deal with the different traffic arrival patterns generated by different flows, which has not been addressed in previous work. Clients may have different demands either due to the different applications they are running, or price they are willing to pay. Even within the same flow, the generated traffic may not be periodic. Applications like MPEG video streaming may generate traffic with variable bit rate (VBR). Thus, mechanisms based on static resource allocation cannot handle this kind of traffic. We consider the different throughput requirements of the clients. Finally, a realistic theory of QoS must take into account the unreliable nature of wireless networks. Wireless transmissions are vulnerable to fading and shadowing effects, resulting in different qualities for each link.

We begin by providing a mathematical framework for QoS support based on an earlier work [5]. The wireless network is described by an abstract client-server model. The model incorporates all the aforementioned criteria: delay bounds, differentiated throughput bounds, various traffic patterns with probabilistic packet arrival, and heterogeneous channel reliability. This abstract model can capture the realistic characteristics of a number of wireless applications. Based on this model, we first derive an extension of a necessary condition for a set of clients to be feasible. We also analyze two earlier proposed dynamic scheduling policies. We analytically prove that both proposed policies can fulfill every set of clients that satisfy the necessary condition. Thus, we not only show that the two policies are optimal but also establish that the necessary condition is indeed sufficient to flows with different arrival patterns. Based on this finding, admission control is reduced to evaluating the necessary and sufficient condition. In summary, we jointly address both admission control and scheduling under the described model, extending it to handle flows with different

traffic generating patterns so that it is applicable to several scenarios of substantial interests.

In addition to the theoretical study, implementation issues are also discussed. We demonstrate that it is easy to implement the policies under the current IEEE 802.11 mechanism. Simulation results for both VoIP traffic and VBR video streaming are shown. The results suggest that the proposed policies are optimal in that they fulfill every feasible set of clients. They also confirm the accuracy of the necessary and sufficient condition.

The rest of the paper is organized as follows. Section 2 summarizes some existing work on providing QoS for VBR traffic. In Section 3, we describe the abstract client-server model with QoS criteria. Section 4 demonstrates how this model can be applied to a variety of wireless applications. In Section 5, we derive a necessary condition for a set of clients with differing traffic generation patterns to be feasible. Two proposed dynamic scheduling policies are proved to be optimal even in a more general setting in Section 7. In Section 8, we discuss how to implement the policies under the IEEE 802.11 mechanisms. Simulation results are shown in Section 9. Finally, Section 10 concludes this paper.

2 Related Work

Providing QoS for wireless multimedia applications is gaining extensive research interest. Stockhammer, Jenkac, and Kuhn [14] have studied the minimum initial delay and the minimum required buffer size for video streaming. Their study considers the case where there is only one wireless client in the system. Kang and Zakhor [7] have focused on improving the quality of video streaming by giving priorities to packets according to the content of the video. Li and Schaar [9] have proposed an adaptive algorithm for tuning the MAC retry limit for layered coded video. These works all lack provable performance bounds. Wongthavarawat and Ganz [15] have studied the scheduling problem in IEEE 802.16. Their result is bound to IEEE 802.16 and not applicable to other MAC mechanisms. Raghunathan et al [11] and Shakkottai and Srikant [13] have derived theoretical results on minimizing the total number of expired packets in a system. Their results, however, cannot provide differentiated QoS for each user. Our work extends the recently proposed method of Hou, Borkar, and Kumar [5] which deals with admission control and scheduling for the restrictive case where all clients generate traffic periodically with the same period. Their results are not applicable to the more complicated traffic patterns. He et al [4] and Zhou et al [16] have considered providing QoS in WSNs. Their studies focus more on implementation issues rather than theoretical results. Fattah and Leung [3] have summarized other existing scheduling algorithms.

3 A Model For QoS With Probabilistic Arrivals

We first describe an abstract client-server system with QoS requirements that generalizes a model that has been proposed in [5]. We will show that the more general model captures the characteristics of a variety of wireless applications of substantial interests, such as MPEG VBR video streaming, VoIP with differentiated quality and wireless sensor networks in Section 4.

Consider a system with N clients, numbered as $\{1,2,\ldots,N\}$, and one server. Clients generate *jobs* for the server to accomplish. We assume that time is slotted. During each time slot, the server can attempt exactly one job. We further assume the time slots are grouped into *intervals*, with each interval containing τ time slots. Jobs can only be generated in the beginning of an interval and must be finished within that interval. Unfinished jobs are discarded at the end of an interval. Thus, a delay bound of τ time slots is imposed on all jobs.

In addition to the delay bound, the QoS requirements can be further classified in three respects: traffic pattern, reliability, and throughput. To reflect different traffic patterns for various applications, we do not restrict attention only to clients that generate one job during each interval as in [5]. Rather, clients generate jobs according to a probability mass function $R : 2^{\{1,2,\ldots,N\}} \to (0,1)$. For a given subset of clients $S \subseteq \{1,2,\ldots,N\}$, the probability that exactly every client in S generates a job in an interval is $R(S)$. Notice that under this generic description, we do not assume clients generate jobs independently. Neither do we assume that jobs generated in an interval are independent from those in other intervals. This extension allows us to provide QoS to several applications noted above.

As in [5], we assume wireless channels are unreliable. When the server attempts to transmit a job to client n, the job gets delivered with probability p_n, which is called the *reliability for client n* . If the attempted job is not delivered, it stays in the system, and the server can further attempt it before the end of that interval. Finally, each client n requires a long-term average throughput of q_n delivered jobs per interval. Since, on average, client n generates $\sum_{S:n\in S} R(S)$ jobs per interval, the throughput requirement is equivalent to a delivery ratio requirement of $\frac{q_n}{\sum_{S:n\in S} R(S)}$. That is, the fraction of discarded jobs for client n cannot exceed $1 - \frac{q_n}{\sum_{S:n\in S} R(S)}$.

In each time slot, the server can either choose to attempt to transmit a job in the system or to stay idle. The choice that the server takes is based on a *scheduling policy*:

DEFINITION 1. *Let H_t be the set of all possible histories of the system up to time slot t. A* ***scheduling policy*** *is a function $\eta : H_t \to \{1,2,\ldots,N,\phi\}$ with the interpretation that, at time slot $t+1$, the server attempts to transmit the job from client n if $\eta(h_t) = n$ or idles if*

$\eta(h_t) = \phi$, *where* $h_t \in H_t$ *is the actual history that the system has experienced.*

The goal of a scheduling policy is to meet the demands of all clients. With an unreliable system, the performance of a policy is not deterministic. Thus, a more careful specification of the requirement for performance is required, as proposed in [5]:

DEFINITION 2. *A set of clients is said to be* ***fulfilled*** *by a scheduling policy* η *if the long-term average throughput of each client n is at least* q_n *jobs per interval with probability 1. That is,*

$$\liminf_{K\to\infty} \sum_{k=1}^{K} \frac{1}{K} 1(\text{a job for client } n \text{ is accomplished in the } k^{th} \text{ interval}) \geq q_n, \text{ with probability 1,}$$

for every client n, where $1(\cdot)$ *is the indicator function.*

Before obtaining such a policy that can fulfill a particular set of clients, one needs to determine whether the set of clients is *feasible*:

DEFINITION 3. *A set of clients is said to be* ***feasible*** *if there exists a scheduling policy* η *that fulfills it.*

Finally, we aim to design an *optimal scheduling policy*:

DEFINITION 4. *An* ***optimal scheduling policy*** *is a policy that fulfills every feasible set of clients.*

4 Examples of Applications

In this section, we describe several delay-sensitive wireless applications that can be described by the extended model introduced in the previous section.

4.1 Video Streaming

In this scenario, each client is a wireless user, such as a laptop or a PDA, and the server is an access point (AP). Each user subscribes to a video stream and requests som quality from the AP. When packets, which correspond to the aforementioned *jobs*, for clients arrive at the AP, the AP can *attempt* them by transmitting a packet to the corresponding client. Upon receiving a packet, the user must reply with an ACK. Thus, the length of a *time slot* is the time required to transmit both a data packet and an ACK. An attempt is considered successful only if the AP receives the ACK from the client. Wireless links are known to be unreliable and vary in quality from client to client. Thus, the parameter p_n captures the heterogeneous link qualities for different clients with the interpretation that whenever the server sends a packet to client n, it receives an ACK with probability p_n.

The MPEG coding algorithm is widely used to generate real-time video traffic. MPEG, along with other video coders, generates VBR traffic to maintain a fixed video quality. Depending on the context, MPEG alternates between three coding modes (I,P,B) that require different numbers of bits per frame. Given a fixed packet size, the three coding modes generate packets at different rates. Thus, the durations between packet arrivals vary throughout the whole video. In terms of our model, this means the packet arrivals in each interval is a random variable, where a higher bit rate implies a higher arrival probability. Thus, the traffic pattern of video streaming can be well-captured by our model.

Finally, while packet loss is inevitable, each user may require a certain delivery ratio bound, which can be converted into a throughput bound, and captured by the parameter q_n. The different values of q_n for each user also reflect that the requested video qualities vary for users.

4.2 VoIP Traffic

Like in the previous example, we have a set of wireless users and an AP, which serves as the server. Each user requests a VoIP traffic flow from the AP. The major difference between a VoIP traffic and and a video stream is that VoIP traffic involves both uplink traffic and downlink traffic. To this end, we create two clients for each user, one for uplink traffic and one for downlink traffic. When the AP attempts a downlink client, n, it sends a data packet for the corresponding user and waits for an ACK. The attempt is considered successful if the AP receives an ACK, which happens with probability p_n. On the other hand, when the AP attempts an uplink client, m, it first sends out a small request message to the corresponding user. The designated user, upon receiving the request message, replies with a data packet. The attempt is considered successful, with probability p_m, if both the request message and the data packet are delivered. The length of a time slot is set large enough to accommodate both an attempt for the downlink client and an attempt for the uplink client.

In this paper, we consider audio codecs that generate constant bit rate (CBR) traffic, such as ITU-T standards G.711 and G.718. Thus, clients generate jobs periodically, with smaller period corresponding to higher bit rate. The job generation time for clients may be offset. For example, consider a set of three clients $\{1,2,3\}$, where clients 1 and 2 have period 2, while client 3 has period 3. Client 1 generates jobs at intervals $1,3,\ldots$, client 2 generates jobs at intervals $2,4,\ldots$, and client 3 generates jobs at intervals $1,4,\ldots$. Thus, we have $R(\{1,3\}) = R(\{2,3\}) = 1/2$. This example demonstrates how our model captures the traffic pattern of VoIP traffic.

4.3 Real Time Surveillance

We now address the scenario of a wireless sensor network for real time surveillance. There are two levels of devices in the network: multiple simple sensor nodes, which corresponds to clients, and a more powerful base station, or data aggregator, that plays the role of the server. To avoid packet collisions between sensor nodes, we adopt a server-centric scheme. When the server attempts a job from a particular client, it polls the corresponding sensor node for data. The attempt is successful, with probability p_n, if a data packet is received by the server. One example of such a setting is a Body Sensor Network (BSN) [10]. In BSN, we aim to record a his-

togram of physiological data for medical use. Timely packet delivery is required in cases of emergency events.

Since sensor nodes monitor a variety of events, their readings are of differing importance. For example, in the context of a BSN, there may be sensor nodes for monitoring heart activity, blood pressure, and body temperature. Thus, we assume each client generates jobs periodically, with the differing frequencies of job generation reflecting the importance of the corresponding data. As described in the previous case, such a traffic pattern also fits into our model.

5 The Necessary Condition for Feasibility

In this section, we extend a necessary condition in [5] for a set of clients to be feasible to the more general model with variable traffic arrival patterns. Intuitively, the more often the server attempts jobs for a client, the higher the throughput that the client gets. This observation is described more formally in the following lemma:

LEMMA 1. *The long-term average throughput of a client n is at least q_n jobs per interval if and only if the server, on average, attempts jobs from that client $w_n = \frac{q_n}{p_n}$ times per interval.*

We will hereby refer to w_n as the *implied attempt rate for client n*. Thus, a set of clients is fulfilled if and only if the average attempts per interval for jobs from each client is higher than its implied attempt rate.

Since the length of an interval is τ time slots and the server can attempt jobs at most once in each time slot, the following necessary condition can be obtained:

LEMMA 2. *A set of N clients is feasible only if $\sum_{n=1}^{N} w_n \leq \tau$.*

This necessary condition turns out, however, to be not sufficient. Since undelivered jobs are discarded at the end of each interval, the server can only attempt jobs that are generated in the current interval. It is possible that, at some time slot of an interval, all jobs are accomplished and the server is forced to stay idle. While the number of idle time slots depends on the scheduling policy, we show its probability distribution is the same for a particular set of policies.

DEFINITION 5. *A scheduling policy is said to be **work conserving** if the server never idles whenever there is any undelivered job at the server.*

LEMMA 3. *The probability distribution of the amount of idle time slots in an interval is the same for all work conserving scheduling policies.*

PROOF. Let γ_n be the random variable denoting the number of attempts the server needs to make for a job from client n before delivering it. γ_n has the geometric distribution with parameter p_n, that is, $Prob\{\gamma_n = t\} = p_n(1-p_n)^{t-1}$ for all positive integers t. Further, assume that a subset S of clients generates jobs in an interval. Let $L_{S,\eta}$ be the random variable indicating the number of idle time slots in such an interval under scheduling policy η. We have:

$$L_{S,\eta} = \begin{cases} \tau - \sum_{n\in S}\gamma_n, & \text{if } \sum_{n\in S}\gamma_n < \tau, \\ 0, & \text{otherwise,} \end{cases}$$

for all work conserving policies. Thus, the probability distribution of $L_{S,\eta}$ is the same for all work conserving policies. □

We will hereby define $L_S := L_{S,\eta}$, where η is any work conserving policy.

The following observation shows that we can always construct a work conserving policy, from any policy, by modifying it so that it performs at least as good as the original policy.

LEMMA 4. *Let η be a scheduling policy that fulfills some sets of clients. Then there exists a work conserving policy η' that fulfills the same set of clients.*

PROOF. The policy η can be modified into a work conserving one by attempting any unaccomplished job whenever η idles. This modification cannot reduce the number of undelivered jobs for any client and thus would fulfill any set of clients that η fulfills. □

Based on this lemma, we can therefore limit our discussion to work conserving policies throughout the rest of the paper. Suppose a subset S of clients generates jobs in an interval, then Lemma 3 implies the expected number of idle time slots in that interval is $E[L_S]$. Since such an interval occurs with probability $R(S)$, the average number of idle time slots in an interval is $\sum_S R(S)E[L_S]$, and the server can, on average, therefore make $\tau - \sum_S R(S)E[L_S]$ attempts in an interval. This observation leads to the following refined necessary condition:

$$\sum_{n=1}^{N} w_n \leq \tau - \sum_{S} R(S)E[L_S]. \tag{1}$$

However, we can go even further by considering all subsets of the set of all clients $\{1,2,\ldots,N\}$. For any subset $S' \subseteq \{1,2,\ldots,N\}$, let

$$\begin{aligned} I_{S'} &:= \sum_{S} R(S)E[\max\{0, \tau - \sum_{n\in S\cap S'} \gamma_n\}] \\ &= \sum_{S} R(S)E[L_{S\cap S'}]. \end{aligned}$$

This is the average number of time slots spent idling in an interval, if S' were the set of all clients. Clearly, if a set of clients is feasible, all subsets of it are also feasible. Hence, we can further refine the necessary condition (1):

LEMMA 5. *A set of clients is feasible only if $\sum_{n\in S} w_n \leq \tau - I_S$ holds for every subset S.*

It may seem that the condition for a strict subset S of $\{1,2,\ldots,N\}$ is redundant, and that we only need to evaluate the condition for all clients. However, the following example shows that merely evaluating the condition (1) is not sufficient.

EXAMPLE 1. Consider a system with interval length $\tau = 3$, and two clients. Each client generates one job in every interval, that is, $R(\{1,2\}) = 1$. The raliabilities for both clients are $p_1 = p_2 = 0.5$. Client 1 requires a throughput of $q_1 = 0.876$, while the throughput requirement of client 2 is $q_2 = 0.45$.

Now, we have:

$$\begin{aligned} w_1 &= 1.76, \\ w_2 &= 0.9, \\ I_{\{1\}} = I_{\{2\}} &= 1.25, \\ I_{\{1,2\}} &= 0.25. \end{aligned}$$

If we evaluate the condition for the subset of $S = \{1\}$, we find $w_1 = 1.76 > 1.75 = \tau - I_{\{1\}}$. This indicates that the set of clients is not feasible. However, if we evaluate the condition for all clients $\{1,2\}$, we have $w_1 + w_2 = 2.66 < 2.75 = \tau - I_{\{1,2\}}$. Thus, this example suggests that merely evaluating the condition for all clients is not sufficient. □

Surprisingly, we will show that the necessary condition stated in Lemma 5 is indeed sufficient in Section 7.

6 Scheduling Policies

In a previous work, Hou, Borkar, and Kumar [5] have proposed and shown that two index type of policies are both optimal when clients generate a job in each interval. (In terms of our model, this means $R(\{1,2,\ldots,N\}) = 1$, supposing there are N clients.) Both policies are *most debt first policies*. In the beginning of each interval, the server computes the *debts* owed to each client and assigns priority accordingly, clients with higher debts getting higher priority. In any time slot during this interval, the server attempts the job from the client with the highest priority among those who have an unaccomplished job. The only difference between the two policies is their differencing definitions on debt.

The first policy, the *most time-based debt first policy*, attempts jobs from each client at least as often as its implied attempt rate:

DEFINITION 6. *Let $u_n(t)$ denote the number of attempts that the server makes for jobs from client n up to time slot t. The* ***time-based debt*** *for client n is defined to be $w_n t - u_n(t)$. The policy that assigns priorities according to the time-based debts is called the* ***most time-based debt first policy.***

The time-based debt reflects how much a client is lagging behind its implied attempt rate. By giving a client with large debt high priority, the client is, on average, granted more attempts during the interval. Thus, that client will have a better chance to catch up with its implied attempt rate.

The next policy adopts a more direct approach by tracking how many jobs the server actually delivers for a client:

DEFINITION 7. *Let $c_n(t)$ denote the number of jobs for client n accomplished by the server up to time slot t. The* ***weighted-delivery debt*** *for client n is defined to be $[q_n t - c_n(t)]/p_n$. The policy that assigns priorities according to the weighted-delivery debts is called the* ***most weighted-delivery debt first policy.***

Note that $q_n t - c_n(t)$ is the number of jobs that the server should accomplish for client n before meeting its demands. Moreover, since the server accomplishes a job for client n with probability p_n every time it makes an attempt, the weighted-delivery debt can be thought of as the number of attempts that the server owes the client. Thus comes the definition of the most weighted-delivery debt first policy.

7 Proofs of Optimality

In this section, we prove that these policies are also optimal for any arbitrary traffic pattern, since, as we have noted in Section 1, many applications actually require variable bit rates.

Our proof is also based on Blackwell's approachability theorem [1].

Consider a single player game with payoff function M, whose value is a probability distribution in the Euclidean N-dimensional space depending on the action taken by the player. Suppose, under some policy, the player takes action $a(i)$ and gets a payoff $v(i)$, which is an N-dimensional vector, in each round i. Blackwell studied the long-term average payoff the player gets, that is, $\lim_{j\to\infty} \sum_{i=1}^{j} v(a(i))/j$, and introduced the concept of *approachability*:

DEFINITION 8. *Let $A \subseteq \mathbb{R}^N$ be any set in the N-dimensional space. Consider a policy η, which incurs payoffs $v(a(1)), v(a(2)), \ldots$. Let δ_j be the distance between the point $\sum_{i=1}^{j} v(a(i))/j$. We shall say A is approachable under policy η, if for every $\varepsilon > 0$ there is a j_0 such that,*

$$Prob\{\delta_j \geq \varepsilon \text{ for some } j \geq j_0\} \leq \varepsilon.$$

In other words, the distance between the point of the long-term average payoff and the set A converges to 0 with probability 1.

Blackwell derived a sufficient condition for approachability:

THEOREM 1. *Let $A \subseteq \mathbb{R}^N$ be any closed set in the N-dimensional space. Let η be a policy whose action depends solely on the average payoff to date, $x_j = \sum_{i=1}^{j-1} v(a(i))/j$. Thus, we can express $a(j)$ by $a'(x_j)$. Then A is approachable under η if the following statement holds:*

If $x_j \notin A$, let y be the closest point in A to x_j and H be the hyperplane passing through y and perpendicular to the line segment $x_j y$. A is approachable under η if H separates x_j and the expected payoff of round j, that is, $E[v(a'(x_j))]$.

Based on this fundamental theorem, we prove that both most debt first policies are optimal. Since a feasible set of clients must satisfy the necessary condition in Lemma 5, we only need to prove that the two policies fulfill every set of clients that satisfy the necessary condition.

THEOREM 2. *The most time-based debt first policy is optimal.*

PROOF. We first translate the model into a single player game. A round in the game corresponds to an interval in the model. The player is the server. The action the player can take is in choosing the priorities of clients, with the interpretation that an unaccomplished job for a client is attempted only after all jobs from clients with higher priorities are accomplished. The payoff the player gets is the net change of the time-based debt owed to each client, which is thus an N-dimensional vector. To be more precise, the payoff the player gets is $v = [v_1, v_2, \ldots, v_N]$, where v_n equals w_n minus the number of times the server attempts the job from client n.

By Lemma 1, the demand of a client n is met if the server attempts its jobs at least $w_n = q_n/p_n$ times per interval on average, or equivalently, the client has a non-positive time-based debt. Thus, to establish the optimality of the most time-based debt first policy, we only need to show that the set $A := \{z = [z_1, z_2, \ldots, z_N] | z_n \leq 0, \forall n\}$ is approachable under this policy.

Suppose that at the beginning of some interval, the average payoff is $x = [x_1, x_2, \ldots, x_N]$. If $x \in A$, no action violates approachability by Theorem 1. If $x \notin A$, at least one of $x_1, x_2, \ldots, x_N$ is strictly positive, and we can reorder the clients so that $x_1 \geq x_2 \geq \cdots \geq x_m > 0 \geq x_{m+1} \cdots \geq x_N$. The closest point in A to x is $y = [0, 0, \ldots, 0, x_{m+1}, x_{m+2}, \ldots, x_N]$. The hyperplane passing through y and perpendicular to the line segment xy is $H := \{z | h(z) := \sum_{n=1}^{m} x_n z_n = 0\}$.

Let $\bar{x}$ be the payoff of this round according to the most time-based debt first policy. Also, let $\bar{w}_n$ be the number of times the sever attempts the job from client n in the interval. We can express $\bar{x}$ as $\bar{x} = [w_1 - \bar{w}_1, w_2 - \bar{w}_2, \ldots, w_N - \bar{w}_N]$.

Since $h(x) = \sum_{n=1}^{m} x_n^2 > 0$, in order to show H separates x and $E[\bar{x}]$, it suffices to show that $h(\bar{x}) \leq 0$. We have:

$$\begin{aligned} h(\bar{x}) &= \sum_{n=1}^{m} x_n (w_n - \bar{w}_n) \\ &= \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} w_k - \sum_{k=1}^{n} \bar{w}_k] \\ &\quad + x_m (\sum_{k=1}^{m} w_k - \sum_{k=1}^{m} \bar{w}_k). \end{aligned}$$

Next we evaluate the value of $\sum_{k=1}^{n} \bar{w}_k$ for each n. First assume that a subset S of clients generate jobs at the beginning of an interval. By the most time-based debt first policy, the server will give priority according to the ordering $1, 2, \ldots, N$. Hence, $\sum_{k=1}^{n} \bar{w}_k$ is the number of attempts the server makes if there are only jobs for a subset $S_n = \{1, 2, \ldots, n\} \cap S$ of clients.

In other words, $\sum_{k=1}^{n} \bar{w}_k = \tau - L_{S_n}$, where L_{S_n} is the random variable indicating the number of time slots that remain idle in an interval when only jobs from clients in the subset S_n are present. Thus we have $E[\sum_{k=1}^{n} \bar{w}_k | S] = \tau - E[L_{S_n}]$. Taking the expected value over all S yields:

$$\begin{aligned} E[\sum_{k=1}^{n} \bar{w}_k] &= E[E[\sum_{k=1}^{n} \bar{w}_k | S]] \\ &= \sum_{S} R(S) E[\sum_{k=1}^{n} \bar{w}_k | S] \\ &= \tau - \sum_{S} R(S) E[L_{S \cap \{1,2,\ldots,n\}}] \\ &= \tau - I_{\{1,2,\ldots,n\}}. \end{aligned}$$

Now, according to the necessary condition stated in Lemma 5, we have $\sum_{k=1}^{n} w_k \leq \tau - I_{\{1,2,\ldots,n\}} = \sum_{k=1}^{n} \bar{w}_k$, for all n. Further, $x_1 \geq x_2 \geq \cdots \geq x_m > 0$. Thus, $E[h(\bar{x}) \leq 0]$, and A is approachable under the most time-based debt first policy by Theorem 1, which also implies that the most time-based debt first policy is optimal. □

THEOREM 3. *The most weighted-delivery debt first policy is also optimal.*

PROOF. Like in the previous proof, we also need to translate this policy into one for the single player game. Again, a round in the game corresponds to an interval in our model. The action a player, which is the server, can take is to decide the priorities of clients. However, in this case, the payoff the player gets is the net change of the weighted-delivery debt. In other words, the payoff is an N-dimensional vector $v = [v_1, v_2, \ldots, v_N]$, where $v_n = (q_n - 1)/p_n$ if the server accomplishes a job for client n in the interval, or $v_n = q_n/p_n$ if not. The throughput of a client n is at least q_n jobs per interval if it has a non-positive weighted-delivery debt. Thus, we can prove that the most weighted-delivery debt is optimal by showing that the set $A := \{z = [z_1, z_2, \ldots, z_N] | z_n \leq 0, \forall n\}$ is approachable.

Let $x = [x_1, x_2, \ldots, x_N]$ be the average payoff at the beginning of an interval. Again, we only need to evaluate the performance of the most weighted-delivery debt first policy under the case $x \notin A$. We can reorder the clients so that $x_1 \geq x_2 \geq \cdots \geq x_m > 0 \geq x_{m+1} \geq \cdots \geq x_N$. The closest point in A to x is $y = [0, 0, \ldots, 0, x_{m+1}, x_{m+2}, \ldots, x_N]$. The hyperplane passing through y and perpendicular to the line segment xy is $H := \{z | h(z) := \sum_{n=1}^{m} x_n z_n = 0\}$.

Let π_n be the indicator function that the server accomplishes a job from client n, which is a random variable. The payoff of this interval is $\bar{x} = [(q_1 - \pi_1)/p_1, (q_2 - \pi_2)/p_2, \ldots, (q_N - \pi_N)/p_N]$.

By Theorem 1, the set A is approachable if H separates x and $E[\bar{x}]$. Since $h(x) = \sum_{n=1}^{m} x_n^2 > 0$, we only need to show $E[h(\bar{x})] \leq 0$ to complete the proof. We

have:

$$
\begin{aligned}
h(\bar{x}) &= \sum_{n=1}^{m} x_n \frac{q_n - \pi_n}{p_n} \\
&= \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} \frac{q_k}{p_k} - \sum_{k=1}^{n} \frac{\pi_k}{p_k})] \\
&+ x_m(\sum_{k=1}^{m} \frac{q_k}{p_k} - \sum_{k=1}^{m} \frac{\pi_k}{p_k}) \\
&= \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} w_k - \sum_{k=1}^{n} \frac{\pi_k}{p_k})] \\
&+ x_m(\sum_{k=1}^{m} w_k - \sum_{k=1}^{m} \frac{\pi_k}{p_k}) \text{ (since } w_k = \frac{q_k}{p_k}).
\end{aligned}
$$

Since $x_1 \geq x_2 \geq \cdots \geq x_m > 0$, it suffices to show $\sum_{k=1}^{n} w_k \leq E[\sum_{k=1}^{n} \frac{\pi_k}{p_k}]$, for every n. Recall that the necessary condition stated in Lemma 5 requires $\sum_{k=1}^{n} w_k \leq \tau - I_{\{1,2,\ldots,N\}}$ for every n, to be feasible. Thus, we only need to show $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k}] = \tau - I_{\{1,2,\ldots,N\}}$ to establish optimality. Further, we have $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k}] = \sum_S R(S) E[\sum_{k=1}^{n} \frac{\pi_k}{p_k} | S]$ and $I_{\{1,2,\ldots,n\}} = \sum_S R(S) E[L_{S \cap \{1,2,\ldots,n\}}]$. The proof is hence complete by showing that $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k} | S] = \tau - E[L_{S \cap \{1,2,\ldots,n\}}]$ for every S and n, which is done in Lemma 6 below. □

LEMMA 6. *Under the priority order* $\{1,2,\ldots,N\}$, $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k} | S] = \tau - E[L_{S \cap \{1,2,\ldots,n\}}]$, *for* $n = 1,2,\ldots,N$, *and all* $S \subseteq \{1,2,\ldots,N\}$.

PROOF. Suppose client m doesn't generate a job in the interval, that is $m \notin S$. Then the server cannot make any attempt for client m, and we have $E[\pi_m | S] = 0$. Also, since $m \notin S$, $m \notin (S \cap \{1,2,\ldots,n\})$, and client m plays no role in deciding the value of $E[L_{S \cap \{1,2,\ldots,n\}}]$. Thus, we can delete every client that is not in S and reorder the remaining clients. Equivalently, we only need to prove $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k} | S] = \tau - E[L_{\{1,2,\ldots,n\}}]$, for $S \supseteq \{1,2,\ldots,n\}$.

We prove this by induction. First consider the case $n = 1$. Since client 1 has the highest priority, its job is accomplished unless the server fails in all the τ attempts. Thus,

$$
\begin{aligned}
E[\frac{\pi_1}{p_1} | S] &= \frac{Prob\{\text{the job of client 1 is accomplished}\}}{p_1} \\
&= \frac{1 - (1 - p_1)^{\tau}}{p_1}.
\end{aligned}
$$

On the other hand, we also have:

$$
\begin{aligned}
E[L_{\{1\}}] &= \sum_{t=1}^{\tau-1} Prob\{\text{the job from client 1 is accomplished in at most } \tau - t \text{ attempts}\} \\
&= \sum_{t=1}^{\tau-1} (1 - (1 - p_1)^{\tau - t}) \\
&= \tau - \frac{1 - (1 - p_1)^{\tau}}{p_1}.
\end{aligned}
$$

This gives us $E[\frac{\pi_1}{p_1} | S] = \tau - E[L_{\{1\}}]$ and the lemma holds for the case $n = 1$.

Assume that $E[\sum_{k=1}^{n} \frac{\pi_k}{p_k} | S] = \tau - E[L_{\{1,2,\ldots,n\}}]$ holds for all $n \leq m$. Consider the case $n = m + 1$. Since the client $m+1$ has the lowest priority among clients $\{1,2,\ldots,m+1\}$, its job is attempted only after all jobs from client 1 through client m are accomplished. Since there are $L_{\{1,2,\ldots,m\}}$ time slots left after the server accomplishes jobs from the first m clients, we have:

$$
\begin{aligned}
&E[\pi_{m+1} | S, L_{\{1,2,\ldots,m\}} = \sigma] \\
&= Prob\{\text{the job of client } m+1 \text{ is accomplished in } \tau - \sigma \text{ attempts}\} \\
&= 1 - (1 - p_{m+1})^{\tau - \sigma}.
\end{aligned}
$$

On the other hand, since $L_{\{1,2,\ldots,m\}} - L_{\{1,2,\ldots,m+1\}}$ is the number of attempts that the server makes for a job from client $m+1$, we also have:

$$
\begin{aligned}
&E[L_{\{1,2,\ldots,m\}} - L_{\{1,2,\ldots,m+1\}} | L_{\{1,2,\ldots,m\}} = \sigma] \\
&= \sum_{t=\sigma+1}^{\tau} Prob\{\text{the server makes at least } t - \sigma \text{ attempts for the job from client } m+1\} \\
&= \sum_{t=\sigma+1}^{\tau} (1 - p_{m+1})^{t - \sigma - 1} \\
&= \frac{1 - (1 - p_{m+1})^{\tau - \sigma}}{p_{m+1}} = E[\frac{\pi_{m+1}}{p_{m+1}} | S, L_{\{1,2,\ldots,m\}} = \sigma],
\end{aligned}
$$

for all σ. Thus, $E[\frac{\pi_{m+1}}{p_{m+1}} | S] = E[L_{\{1,2,\ldots,m\}} - L_{\{1,2,\ldots,m+1\}}]$. Finally, we have:

$$
\begin{aligned}
&E[\sum_{k=1}^{m+1} \frac{\pi_k}{p_k} | S] \\
&= E[\sum_{k=1}^{m} \frac{\pi_k}{p_k} | S] + E[\frac{\pi_{m+1}}{p_{m+1}} | S] \\
&= \tau - E[L_{\{1,2,\ldots,m\}}] + E[L_{\{1,2,\ldots,m\}} - L_{\{1,2,\ldots,m+1\}}] \\
&= \tau - E[L_{\{1,2,\ldots,m+1\}}].
\end{aligned}
$$

By induction, the lemma holds for all n. □

A final remark is that, since both policies fulfill every set of clients that satisfy the necessary condition in Lemma 5, this condition is also sufficient for feasibility.

THEOREM 4. *A set of clients is feasible if and only if* $\sum_{n \in S} w_n \leq \tau - I_S$ *holds for every subset S.*

8 Implementation on IEEE 802.11

While our generic model can be applied to a variety of wireless applications, we believe that the WLANs can particularly benefit most from our study due to their wide deployment, and increasing demands of QoS support. In this section, we show that the proposed policies can be easily implemented in the current IEEE 802.11 mechanisms.

The IEEE 802.11 defines two transmission modes for Medium Access Control (MAC) [6]: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The two mechanisms can coexist by dividing a *superframe*, which corresponds to the interval in our model, into a PCF contention-free period (CFP) followed by a DCF contention period (CP). The PCF mechanism is server-centric, and thus suitable for implementing the proposed policies.

In the PCF mode, when the server schedules a downlink transmission, it sends out the data to a client after sensing the channel being idle for a period of PIFS. The client, after receiving the packet, waits a period of SIFS and then replies with an ACK. When the server schedules an uplink transmission, it sends out a CF-POLL packet, which contains information about which client is scheduled to transmit, after the channel is idle for a period of PIFS. The designated client replies with a data packet, or a NULL packet if it does not have any data to send, after waiting for a period of SIFS upon receiving the CF-POLL packet. On the other hands, any node that operates in the DCF mode must wait for the channel being idle for at least a period of DIFS before transmitting a packet. The value of DIFS is set larger than both the values of SIFS and PIFS. Thus, the server is granted the highest priority when it operates in the PCF mode.

9 Simulation Results

We have implemented both most debt first policies, namely, the most time-based debt first policy and the most weighted-delivery debt first policy, on ns-2 under the IEEE 802.11 PCF mechanism. We evaluate the performance of these two policies under two scenarios, one for the MPEG video streaming traffic and one for the VoIP traffic. We compare the most debt first policies with the naive approach of using the IEEE 802.11 DCF standard and a *random priority policy* that also operates in the PCF mode. The random priority policy works like the most debt policies, only that the server assigns priorities to clients randomly at the beginning of each interval. We define a metric, *throughput insufficiency*, to reflect the difference of the desired throughput and the actual throughput. To be more specific, let $d_n(t)$ be the actual average throughput of client n at some time slot t. The throughput insufficiency of client n is $q_n - d_n(t)$ if $q_n > d_n(t)$ or 0 otherwise. The throughput insufficiency of the system is the sum of the throughput insufficiency over all clients.

9.1 VoIP Traffic

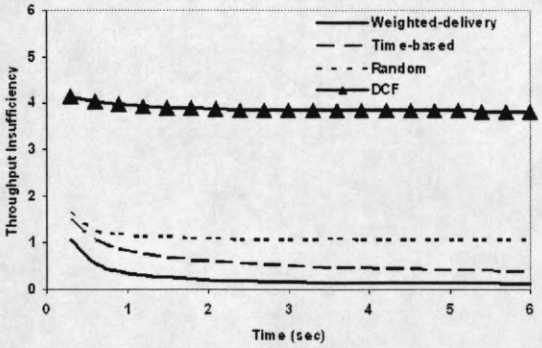
We follow the ITU-T G.729.1 [12] codec, which generate traffic with bit rates ranging from 8 kbits/s to 32 kbits/s, in simulating VoIP traffic. We assume an interval length of 20 *ms* and 160 Bytes VoIP packet. IEEE 802.11b is used as the underlying MAC protocol. Related parameters are described in Table 1. Under this setting, the transmission times for both the uplink traffic, consisting of a CF-POLL packet and a data packet, and the downlink traffic, consisting of a data packet and an ACK, are slightly less than 610 μs, allowing 32 time slots in an interval.

Table 1: Simulation Setup For VoIP

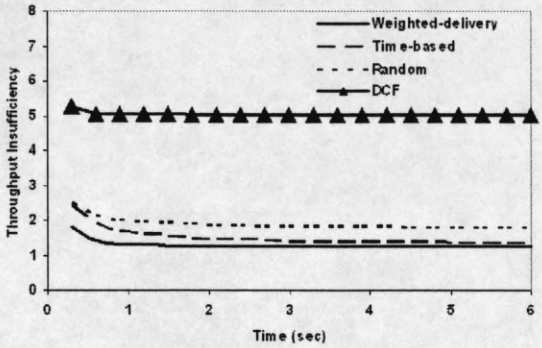
Interval	20 *ms*
Payload size per packet	160 Bytes
Transmission data rate	11 Mb/s
SIFS	10 μs
PIFS	30 μs
DIFS	40 μs

We consider two groups of clients, group A and group B. Each client in group A generates packets periodically with period 60 *ms*, resulting in a 21.3 kbits/s flow, and requires 99% delivery ratio. Clients in group B also generate packets periodically but with period 40 *ms*, which corresponds to 32 kbits/s flows, and require 80% delivery ratio. The starting times of clients in each group are separated evenly. To be more specific, we can further divide the two groups into subgroups A_1, A_2, A_3, B_1, and B_2. Clients in subgroup A_i generate packets at the beginning of intervals $i, i+3, i+6, \ldots$, while clients in subgroup B_j generate packets at the beginning of intervals $j, j+2, j+4, \ldots$. Evaluating the necessary and sufficient condition in Theorem 4 suggests that a set of 6 clients in each of the subgroup A_i and 5 clients in each of B_j is feasible while a set of 6 clients in each of A_i and 6 clients in each of B_j is not.

Figure 1a shows the simulation results for the aforementioned feasible set of clients on the four tested policies, namely, the two most debt first policies, the random policy, and the DCF mechanism. The throughput insufficiencies of both most debt first policies converge to 0 over time, showing that they fulfill this set of clients. However, the most weighted-delivery debt first policy converges much faster than the most time-based debt first policy. This is because the weighted-delivery debt reflects the actual throughput a client is having, and thus is a more direct and precise measure than the time-based debt. While the most time-based debt first policy may be easier to implement, the most weighted-delivery debt first policy should be preferred when tight performance is important. The other two policies both fail to fulfill this set of clients, indicating that they



(a) Performance of a feasible set



(b) Performance of an infeasible set

Figure 1: Throughput insufficiency for VoIP traffic

cannot be optimal. The random policy, though also operating in the contention-free PCF mode, fails to fulfill the set of clients because it does not consider the differentiated requirements of the clients. The DCF mechanism has the worst performance among the four since it suffers greatly from contentions and collisions.

To verify the correctness of the necessary and sufficient condition in Theorem 4, we also run simulation on the predicted infeasible set of clients composed of 6 clients in each subgroup A_i and 6 clients in each subgroup B_j. The results are shown in Figure 1b. All the four tested policies of course fail to fulfill this set of clients, since it is indeed infeasible. Further, it can be noted that although the two most debt first policies do not fulfill this set of clients, they still incur less throughput insufficiency than the other two policies. This result shows that the proposed policies perform well in comparison to some other policies even when dealing with an infeasible set of clients.

9.2 MPEG Video Streaming

We consider the case where a number of wireless users request MPEG video stream traffic with various requirements from the AP. Since video stream requires much larger bandwidth than VoIP traffic, we assume the system uses the higher data rate IEEE 802.11a. Some related parameters are shown in Table 2. Under this setting, the transmission time for a data packet and an ACK is roughly 650 μs, allowing 9 time slots in an interval.

Table 2: Simulation Setup For Video Streaming

Interval	6 ms
Payload size per packet	1500 Bytes
Transmission data rate	54 Mb/s
SIFS	16 μs
PIFS	25 μs
DIFS	34 μs

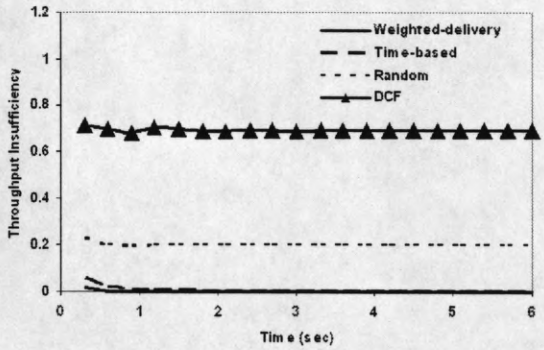
Some previous works [8] [2] model the MPEG VBR traffic by a Markov chain consisting of three activity states. Traffic with different bit rates is generated in each of the three states. Martin et al [2] hence derived statistical results for the movie "The Graduate". We adopt their model for the MPEG VBR traffic and transfer the statistical result into the packet arrival probability in each interval under our setting, which is summarized in Table 3.

Table 3: MPEG Traffic Pattern

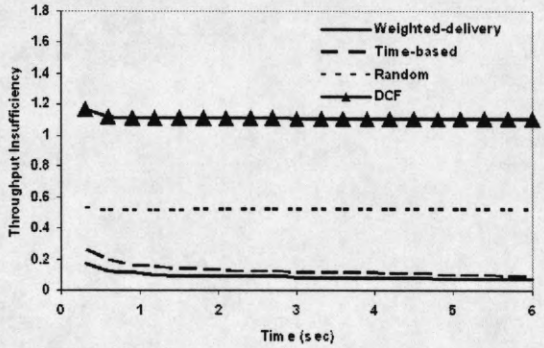
Activity	Great	High	Regular
Arrival probability	1	0.8	0.75

We assume there are two groups of clients, A and B. Clients in group A require high quality video that generates packet according to Table 3 and a 90% delivery ratio for each packet, resulting in a requested throughput of 0.765 packet per interval. Clients in group B only require low quality video that generates packets only 80% as often as those in group A and demand a 60% delivery ratio, or a throughput requirement of 0.408 packet per interval. The channel reliability of the n^{th} client in each group is $(60+n)\%$. By evaluating the necessary and sufficient condition in Theorem 4, we predict that a set of 4 group A clients and 4 group B clients is feasible, while a set of 5 group A clients and 4 group B clients is not.

Figure 2a shows the simulation results on the feasible set of clients composed by 4 group A clients and 4 group B clients. Like in the case of VoIP traffic, the throughput insufficiency of both the two most debt policies converge to zero over time, and the two policies therefore fulfill this set of clients. Also, the most weighted-delivery debt first policy converges faster than the most time-based debt first policy. The random policy and DCF, on the other hand, fail to fulfill this set of clients. In the case of video streaming, the AP is the only wireless device that generates traffic. Thus, there should be no contention and collision in the system. Still, the CSMA/CA mechanism used by the DCF forces the AP to backoff a random time before each transmission. This overhead results in



(a) Performance of a feasible set



(b) Performance of an infeasible set

Figure 2: Throughput insufficiency for video streaming

much higher throughput insufficiency for DCF, than that for the random policy.

Simulations on the infeasible set consisting of 5 group *A* clients and 4 group *B* clients are also conducted, and the results are shown in Figure 2b. All the four tested policies of course fail to fulfill this set of clients. These results also demonstrate that our model can be applied to a wide range of applications. Finally, the two most debt first policies have the least throughput insufficiency among the four tested policies, showing that they offer good performance even for an infeasible set of clients.

10 Conclusions

We have analytically addressed the problem of providing QoS support for heterogeneous VBR traffic flows over the unreliable wireless channels. We study an extension of a proposed mathematical model that incorporates delay bounds, throughput bounds, traffic patterns, and channel reliabilities. This extended model turns out to adequately capture the characteristics of a variety of wireless applications, including video streaming, VoIP, and BSN. Based on the model, we have derived a necessary and sufficient condition for a set of clients to be feasible. Admission control is thus reduced to evaluate the necessary and sufficient condition. We have also studied the scheduling problem, and studied two scheduling policies. We prove that these two policies are both optimal in the sense that they fulfill every feasible set of clients. In addition to theoretical study, we have also addressed implementation issues under IEEE 802.11 and implemented the two scheduling policies in ns-2. Simulation results have confirmed our theoretical studies and shown that the proposed scheduling policies outperform other tested policies under a variety of settings.

11 References

[1] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific J. Math*, 6(1), 1956.

[2] I. V. Martin F., J.J. Alins-Delgado, M. Aguilar-Igartua, and J. Mata-Diaz. Modelling an adaptive-rate video-streaming service using Markov-rewards models. In *Proc. of QSHINE 2004*.

[3] H. Fattah and C. Leung. An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wireless Communications*, October 2002.

[4] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. Speed: a stateless protocol for real-time communication in sensor networks. In *ICDCS 2003*.

[5] I-H. Hou, V. Borkar, and P. R. Kumar. A theory of QoS for wireless. *UIUC Technical Report*, (UILU-ENG-08-2215 DC-239), Sept. 2008.

[6] IEEE. Wireless LAN medium access control (MAC) and physical (PHY) specifications. 1999.

[7] S. H. Kang and A. Zakhor. Packet scheduling algorithm for wireless video streaming. In *PV 2002*.

[8] L.J. De la Cruz and J. Mata. Performance of dynamic resources allocation with QoS guarantees for MPEG VBR video traffic transmission over ATM networks. In *Proc. of GLOBECOM 1999*.

[9] Q. Li and M. van der Schaar. Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation. *IEEE Trans. on Multimedia*, 6(2), 2004.

[10] B. Lo, S. Thiemjarus, R. King, and G-Z. Yang. Body sensor network – a wireless sensor platform for pervasive healthcare monitoring. In *PERVASIVE 2005*.

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

[12] S. Ragot, B. Kovesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Garner, S. Schandl, H. Taddei, Y. Gao, E. Shlomot, H. Ehara, K. Yoshida, T. Vaillancourt, R. Salami, M.S. Lee, and D.Y. Kim. ITU-T G.729.1: an 8 –32 kbit/s scalable coder interoperable with G.729 for wideband telephony and voice over IP. In *ICASSP 2007*.

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

[14] T. Stockhammer, H. Jenkac, and G. Kuhn. Streaming video over variable bit-rate wireless channels. *IEEE Trans. on Multimedia*, 6(2), April 2004.

[15] K. Wongthavarawat and A. Ganz. Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. *Int. J. Commun. Syst.*, 2003.

[16] G. Zhou, J. Lu, C-Y. Wan, M. D. Yarvis, and J. A. Stankovic. BodyQoS: adaptive and radio-agnostic QoS for body sensor networks. In *Proc. of INFOCOM 2008*.