Admission Control Framework to Provide Guaranteed Delay in Error-prone Wireless Channel

Prasanna Chaporkar and Saswati Sarkar

Abstract

We consider a wireless system in which sessions traverse different channels with different transmission characteristics. We propose a general framework for admission control and scheduling that provides stochastic delay and packet drop guarantees in this error-prone wireless system. By "general" we mean that the scheduling policies from a large class can be plugged in this framework and admission control conditions can be obtained for different arrival processes. This enables the use of many scheduling policies that have not been considered so far for the error-prone wireless systems. Using large deviation bounds and renewal theory, we prove that once a session i is admitted, irrespective of the scheduling policy and the channel errors experienced by other sessions, i obtains its desired quality of service. The admission control algorithm uses only the individual channel statistics of sessions and not the joint statistics, and the scheduling does not require any knowledge of instantaneous channel states.

I. Introduction

The next generation wireless packet networks will have to provide delay, delay jitter and packet loss guarantees to heterogeneous real-time traffic. Emergency operations during disaster recovery and multimedia applications like on-demand video, distance learning, tele-conferencing can not be supported without these guarantees. For these applications, every packet has a deadline, and packets that are not served before their deadlines must be dropped. Real-time applications can

The authors are with the department of electrical and systems engineering at the University of Pennsylvania.

Parts of this paper have been presented at WoWMoM 2002 and VTC Fall 2003

accommodate some packet loss without noticeable degradation in the quality of service, e.g., for voice, depending on the encoding and transmission schemes used, 20% packet loss can be acceptable [1]. The tolerable loss is different for different applications. *We focus on obtaining a resource allocation scheme that ensures the delivery of the required fraction of packets before the respective deadlines for each application in a wireless system.*

Any framework that provides delay and packet loss guarantees needs (1) an *admission control mechanism* on the control path and (2) a *packet scheduling* on the data path. The admission control admits a session only if enough resources are available to limit its delay and packet loss, while the scheduling distributes the resources among the admitted sessions in accordance with their delay and packet loss requirements. We now examine the challenges in designing admission control and scheduling schemes for wireless networks. First the admission control schemes in wireline networks *can not* be easily extended for use in wireless systems. This is because location dependent, bursty channel errors in wireless systems lead to increased delays in delivering the packets to the destinations and consequently additional deadline violations and packet drops. This additional packet drop may cause unacceptable degradation in the quality of reception. Thus, the decision regarding whether to admit a new session must be different for wireline and wireless systems. For example in Figure 1, if there are no channel errors (wireline case) then the required delay guarantee can be provided to every packet of every session (Figure 1(a)). In the presence of channel errors (wireless case), the delay experienced by each packet is increased by 1 time unit (Figure 1(b)). If the packets that are delayed beyond their deadlines are dropped, then all the packets of session 3 will be dropped and the remaining packets will be served (Figure 1(c)). Thus, sessions 3 suffers 100% drop, whereas the drop rate for every session is 0 when there are no channel errors. Therefore, unlike a wireline system, a wireless system can not provide the desired QoS to the three sessions, and therefore all these sessions should not be admitted.

The packet drops can be reduced in wireless systems by admitting fewer sessions and thereby reserving additional resources to compensate for the slots in which the channels have poor transmission quality. The resource reservations must be just sufficient to reduce the packet drop below the acceptable level, as increase in reservations leads to higher session blocking. The challenge is

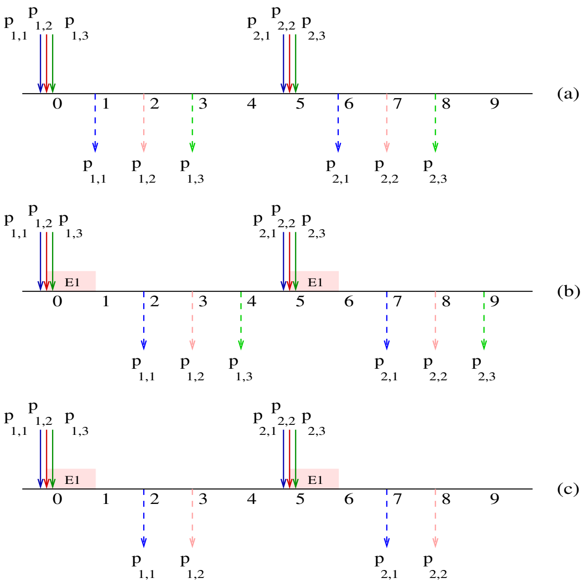

Fig. 1. A server is serving packets for three sessions, namely, 1,2 and 3. Each session has a leaky bucket constrained arrival process with bucket depth (σ) as 1 and token replenishment rate (ρ) as $1/5$. An arrival pattern that fits the above characterization is one that generates a unit length packet for each session in slots $0, 5, 10, \ldots$. A session can tolerate a maximum delay of 3 slots for each packet. Thus, all packets arriving in slot 0 have deadline 3. Let Earliest Deadline First (EDF) scheduling policy be used. Thus the finish time of each packet equals its deadline. The packet with least finish time is selected for transmission. Ties are always broken in favor of the session with the smallest index number. The arrows pointing towards and away from the axis represent packet arrivals and departures, respectively. Let $p_{k,i}$ denote the k^{th} packet from session i. $E1$ indicates that session 1 has erroneous channel in the slot. The arrival process is the same for parts (a), (b) and (c) and channel errors are the same for parts (b) and (c). Part (a) shows the service process for the wireline case (no channel errors). Part (b) shows the service process in erroneous channel if packets are not dropped even when their deadline expires. Part (c) shows the service process if a packet is dropped when its deadline expires.

to *quantify* this excess reservation, and use the quantification to design a channel statistics aware admission control algorithm.

A node may need to deliver packets of different sessions to different destinations (Figure 2). Thus, different sessions may have different sequences of channel errors. The delay experienced by a session depends on the channel errors of all sessions. For example, in Figure 1(b), session 3 experiences an additional delay of 1 slot even though it never had a channel error. This has two undesirable consequences. Firstly, the channel errors of a new session can affect the delay guarantees of the existing sessions. Thus, the joint channel state process of all the sessions has to be analyzed to determine whether the packet drops of the existing sessions remain below the required thresholds even after a new session is admitted. Channel states for various sessions are correlated, and estimating this correlation is not feasible in practice. Secondly, excessive channel errors of one session may increase the packet drops of another session. For example, in Figure 1(c), even when only session 1 has channel errors, all the packets of session 3 are dropped, while all the packets of session 1 are transmitted. Thus, if the admission control process uses an erroneous estimate of the channel statistics of a session, then other sessions may also have unacceptable packet drops.

We now outline our contribution. We consider a scenario where a single node transmits packets for different sessions using wireless channels to different destinations within its transmission range (Figure 2). Sessions arrive at the node at different times and seek admission. We develop a framework that decides whether to admit a session, given the delay, packet loss requirements and channel statistics of the session and other existing sessions. The delay and packet loss requirements and channel statistics may be different for different sessions. A session is admitted if there are enough resources to satisfy its delay and loss requirements, without disrupting the guarantees for the existing sessions; otherwise the session is blocked. Clearly, this decision depends on the particular scheduling policy used. We do not design a new scheduling policy, but design a method to obtain the admission control condition for any given scheduling policy. This enables the use of many scheduling policies that have not been considered yet for the error-prone wireless systems. Consequently, any desired differentiation of service can be obtained among different sessions. Using large deviation bounds and renewal theory, we prove that once a session is admitted, irrespective of the scheduling policy, the session's expected packet drop rate is below its acceptable threshold, and the packets that are not dropped are delivered before their deadlines. The admission control algorithm uses only the individual channel statistics of sessions and not the joint statistics, and the scheduling does not require any knowledge of instantaneous channel states. The framework can, however, use the knowledge of the instantaneous channel states, if these are known, and improve the performance further. We also prove that a session's expected packet drop rate does not exceed its acceptable threshold even if the channel statistics of other sessions have been estimated incorrectly. Thus the proposed framework is robust.

The paper is organized as follows. In Section II, we review the relevant literature and distinguish our work from the existing results. In Section III, we describe our system model and notation. In Section IV, we present a general framework for admission control, which can provide delay and drop guarantees with any scheduling. In Section V, we show how to augment any given scheduling policy for use in the general framework. In Section VI, we present the performance guarantees for the framework. We conclude in Section VII. We present the proofs in the appendix.

II. Literature Review

A prerequisite for developing a combined framework of admission control and scheduling to limit the delay and the packet drop rate of sessions in wireless systems is to quantify the delay experienced by packets and the packet drop rates for different scheduling policies, arrival parameters and channel characteristics. Most of the prior work obtain delay guarantees for sessions that do not experience channel errors [2], [3], [4], [5]. Lu *et al.* [3], [4] upper bound the delay experienced by a packet after it reaches the head of line position in presence of channel errors. But, this does not upper bound the overall delay experienced by a packet in presence of channel errors. Further, the authors assume that the scheduler knows the channel state in a slot before deciding which packet to transmit in the slot. This knowledge is difficult to acquire. Krunz *et al.* [6] quantified the expected delay for a FIFO scheduler in presence of random arrivals and channel states. They also consider a single channel, and assume that the scheduler knows the state of the channel before scheduling. Recently, Wang *et al.* [7] quantified the expected delay for static priority scheduling in a single wireless channel. Furthermore, under static priority scheduling, a packet for a session is served only when the system does not have a packet for a session with a higher priority. Packet drops of low priority sessions can be reduced without increasing that of high priority sessions if a packet from a low priority session whose deadline is close to the current time is served rather than a packet from another high priority session whose deadline is later. But, this requires a dynamic scheduling policy where the priorities of sessions can be altered depending on the deadlines and the service received so far. The framework we consider accommodates such scheduling policies. Recently, Ayyorgun *et al.* [8], [9] obtained an admission control condition and a scheduling policy to guarantee the required delay and packet loss. But, the admission control condition assumes the knowledge of the channel state in all future slots. They also consider a single wireless channel. Summarizing, the delay experienced by packets in presence of multiple sessions that traverse different channels, experience poor transmission quality, and have different scheduling policies have not been characterized yet.

Shakkotai *et al.* [10] have shown that the channel state aware Earliest Deadline First (EDF) scheduling minimizes the packet drop due to deadline expiry for many arrival processes. Tassiulas *et al.* [11] and Shakkotai *et al.* [12] have presented scheduling policies that maximize the throughput in lossy channels. Tassiulas *et al.* [11] and Goyal *et al.* [13] have presented scheduling policies that minimize the average delay in wireless systems. Tassiulas *et al.* [11], Shakkotai *et al.* [12] and Goyal *et al.* [13] assume that packets do not have deadlines. Depending on the traffic load and channel conditions, the packet drop of the EDF policy [10] and the delay of the delay-minimal policies may not remain below an acceptable level. Admission control is necessary to limit the packet drop and the delay. Tassiulas *et al.* [11] and Shakkotai *et al.* [10], [12] do not quantify the packet drop and the delay. Goyal *et al.* [13] present a computationally complex Markov decision process (MDP) formulation for computing the delay experienced by an arbitrary policy; but, solving an MDP every time a session requests admission is not realistic. Thus, the results here can not be easily extended to obtain a joint admission control and scheduling framework. Kim *et al.* [14] studied the packet drop for FIFO schedulers due to buffer overflow in wireless systems. We consider packet drop due to deadline violation.

Admission control for minimizing the dropping of calls during handovers in cellular networks has been extensively studied [15], [16], [17]. Jain *et al.* [15] provides a detailed taxonomy of these admission control schemes. The issue in this scenario is of limiting the severity, frequency and duration of overload due to mobility and hand-offs. Generally, these approaches *assume* that the amount of resources that must be reserved in a cell to provide the required QoS is known. This QoS can be Signal to Interference Ratio (SIR) [18], [19], average bandwidth, delay or packet loss

[20], [21], [22]. Given these required resources, the authors explore how much resource must be reserved in the adjacent cells to reduce the *call dropping* probability in case of hand-offs. The authors provide good heuristic to obtain desired balances between call dropping and call blocking [23], [24], [25], assuming the knowledge of resources required to guarantee the desired performance. In contrast, we concentrate on quantifying the resources required to deliver the desired QoS in presence of the channel errors. The schemes for cellular networks can use our analysis to determine the resources required in the home and adjacent cells.

Substantial amount of work has been done in the recent past to provide channel state aware scheduling in wireless scenario [26], [27], [28], [29], [30], [31]. The main objective in most of the cases was to distribute available resources fairly among the sessions [2], [3], [4], [5], [32]. When fairness is the primary criteria, a question of delay violation does not arise and hence decreasing the packet drop was not considered.

III. System Model

We consider a dynamic scenario where sessions arrive at a source node and seek to transfer information to destinations via error prone channels (Figure 2). The destinations are within the transmission range of the source. The source decides whether or not to admit sessions and schedules the packets. Thus, the packet loss is only due to channel errors and not due to collisions or interference. Time is slotted. Each packet has unit length, arrives at the beginning of a slot, and can be served in the slot in which it arrives.

Each session i presents a four tuple $(\vec{A}_i, \vec{C}_i, D_i, \mu_i)$, where $\vec{A}_i$ and $\vec{C}_i$ are a set of parameters describing i's arrival process and channel statistics respectively, and D_i and μ_i are the required delay and packet drop guarantees respectively. Now we explain each quantity.

We consider only deterministic traffic characterizations. A traffic characterization is deterministic if there exists a non-decreasing function $g(\tau)$ such that the maximum amount of data that the session can send in any duration τ is less than or equal to $g(\tau)$. For example, a $(\sigma, \rho)-$leaky bucket constrained traffic is a deterministic traffic, where $g(\tau) = \rho\tau + \sigma$, σ is the bucket depth and ρ is the token replenishment rate [33]. Session i's traffic is characterized by the function

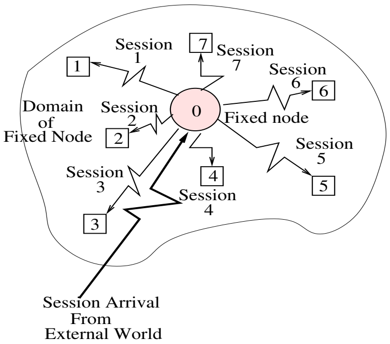

Fig. 2. Sessions from the external node arrive at a source node (node 0). Each session transfers data to one of the destination nodes (nodes 1,2,. . .,7), e.g., session 1 transmits data to node 1.

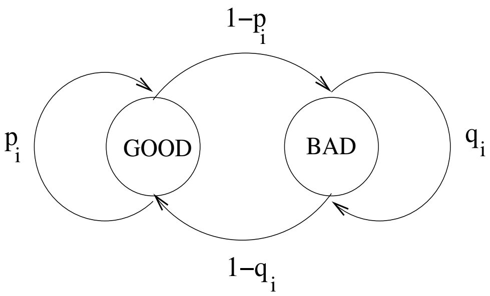

Fig. 3. Figure shows a two state Markov chain that models bursty channel errors for session i.

$g_i(\tau)$. Now, $\vec{A}_i$ is a set of parameters that characterize $g_i(\tau)$, e.g., $\vec{A}_i = \{\sigma_i, \rho_i\}$ for a leaky bucket constrained session i.

We model the channel for each session by a two state Markov Chain (MC) (Figure 3). If the channel state is *good* (*bad*) for session i, then the probability of the successful transmission for session i is 1 (0). A packet is said to be *successfully transmitted* when it is transmitted in a slot in which its session's channel has a good state. The probability of transition of session i's channel from the good (bad) state to the bad (good) state is p_i (q_i). Thus, $\vec{C}_i = \{p_i, q_i\}$. This is the classical Gilbert-Elliot (GE) model [34], [35], and has been widely used to analyze the performance of the wireless systems [6], [7], [10], [14]. Different sessions may have different channel statistics depending on their destinations, power control and channel coding schemes. The channel states of different sessions may be correlated. The source may not know the nature of such correlations.

Session i can tolerate delays of up to D_i. A packet P arriving at t should be successfully transmitted before $t + D_i$; otherwise the packet is dropped. Thus, $t + D_i$ is a *deadline* for P. Let session i transmit n_i packets in all, $n_i < \infty$. The value of n_i may not be known when the session arrives. Let $\widehat{n}_i$ denote the number of session i packets dropped. Since the channel states are random, $\widehat{n}_i$ is a random variable. The expected packet drop rate of session i, $\frac{\mathbf{E}[\widehat{n}_i]}{n_i}$ must not exceed μ_i for all possible sequences of packet arrivals of all sessions as long as the arrivals conform to the specified characterization $g_i(\tau)$. Thus,

$$\frac{\mathbf{E}[\widehat{n}_i]}{n_i} \leq \mu_i. \tag{1}$$

The expectation is taken with respect to the channel state processes of the sessions.

Let session i seek admission. The admission control algorithm determines whether (1) holds for i and other existing sessions, given the characterizations $(\vec{A}_i, \vec{C}_i, D_i, \mu_i)$ and $(\vec{A}_j, \vec{C}_j, D_j, \mu_j)$ other existing sessions j (*admission control condition*). If this is the case, then i is admitted, otherwise it is blocked.

The value of $\frac{\widehat{n}_i}{n_i}$ can be much greater than μ_i along some sample path, even if (1) holds. Now we explore whether worst case guarantees for packet drop can be provided, i.e., given a finite n_i is it possible to ensure that $\frac{\widehat{n}_i}{n_i} \leq \mu_i$ along every sample path. Assume that session i remains in the system for t_i slots. If t_i is finite, there exists a sample path of non-zero probability such that i's channel has errors in all $t_i + D_i$ slots. Hence, i's packets are all dropped. Thus, on such a sample path, $\frac{\widehat{n}_i}{n_i} = 1 > \mu_i$. *Thus, the worst case guarantees for packet drop can not be provided.* If, however, t_i is large, then the probability of such a sample path is small. Also, then n_i is large, and thus the fraction of packets dropped in most sample paths will be close to $\frac{\mathbf{E}[\widehat{n}_i]}{n_i}$. Now, t_i and n_i are large for many real-time applications, e.g., tele-conference, video on demand, etc. We therefore consider the expected fraction of packets dropped as the QoS measure.

The admission control algorithm requires knowledge of only channel statistics that can be learned through effective estimation techniques [2]. We do not however assume the knowledge of the instantaneous state of the channel at a scheduling instance. Thus, a session with *bad* channel may be served, and the transmitted packet can be received in error at the destination. If the deadline for this packet has not expired, then the packet may be re-transmitted. We assume immediate feedback from each destination about the correctness of the received packet. Finally, the algorithms we present can also use the knowledge of instantaneous channel state and decrease the packet drop rate of the sessions, if such information is available about some or all sessions.

IV. Admission Control Framework

We describe the admission control algorithm in this section. The admission control condition will depend on the precise scheduling scheme used. We first describe the class of scheduling strategies we consider. When a packet arrives, the scheduling policy assigns it a time stamp, the "*finish time*". The finish time of the packet is computed according to the specific scheduling policy, e.g., earliest deadline first (EDF) [36], first come first serve (FCFS), fair queueing policies like PGPS [37] (a.k.a. WFQ [38]), SCFQ [39], VirtualClock (VC) [40], WF^2Q [41], SFQ [42], a class of scheduling policies called Rate Proportional Servers (RPS) [43] and the policies that promise certain service-curve [44]. We discuss the computations of the finish times in Section V.

Now we outline the scheduling. We present the details in Section V. In every slot, the scheduling policy determines which session would be served in the slot. This decision depends on the least finish time of packets in the sessions' queues and the number of packets of each session that had to be re-transmitted. A session has higher priority if it has a packet with a low finish time and if it had fewer re-transmissions. The knowledge of the channel states of the sessions, if available, is also incorporated in the decision process. Once the session is selected, the source transmits the packet in the session's queue that has the least finish time. Thus, the finish time is used as a priority indicator among the sessions and the packets of the same session. A large class of dynamic scheduling policies [36] may be implemented in wireless systems by appropriately selecting the algorithm for allocating the finish times.

Now we explain the relation between the admission control, a packet's finish time and deadline. Recall that the deadline of a packet from session i is the sum of the packet's arrival time and i's acceptable delay, D_i. For a given scheduling policy, the primary goal of an admission control algorithm is to ensure that the expected fraction of packets dropped for any admitted session i is at most μ_i. This is sufficient to guarantee the required QoS. The admission control condition is different for different scheduling policies as the policies assign priority structure (by assigning finish times) in different fashion. Thus the packet transmission sequence may differ for different policies. By successfully transmitting packets before their finish times, scheduling policies attain various objectives in addition to limiting the delay, e.g., PGPS guarantees short term fairness [37], VC guarantees isolation [40], and SCED guarantees certain service curve [44]. If the scheduling strategy does not desire any property other than the acceptable delay, then it sets the finish time of each packet equal to the packet's deadline; this scheduling is referred to as earliest deadline first (EDF). We consider a class of scheduling policies that assign to each packet a finish time which is smaller than or equal to the packet's deadline. The class is large, and contains all of the well known policies mentioned above. The admission control strategy admits a session i only if in absence of channel errors each of its packets can be successfully transmitted before the packets' respective finish times*, and in presence of channel errors its packets are successfully transmitted before their finish times with a probability of at least $1 - \mu_i$. Nevertheless, channel errors may delay a packet's successful transmission beyond its finish time. The session tries to transmit a packet until its deadline, but drops a packet from the queue if it is not successfully transmitted before its deadline. This motivates the following definition.

Definition 1: Delay D_i is *guaranteeable* for session i if under every arrival stream that satisfies the session's traffic characterization, each packet of session i has a finish time less than or equal to the packet's arrival time plus D_i and is transmitted successfully before its finish time. Let $\vec{D} = (D_1, \ldots, D_j)$. Delay $\vec{D}$ is guaranteeable if delay D_i is guaranteeable for each i.

We present the pseudo code for the admission control algorithm in Figure 4. We next present the intuition behind the design.

Definition 2: A busy period is a sequence of contiguous slots such that the system has a packet to transmit in each slot in the sequence, and the system does not have a packet to transmit in the

*Under some scheduling policies the time stamp assigned corresponds to a virtual time, e.g PGPS. In most scheduling policies, there exist one to one map between the virtual time and the real time in every busy period. Hence without loss of generality, we can assume that a time stamps correspond to a real time.

Procedure Admission_Control($i, \mathcal{C}$)
begin

/* Session i seeks admission */

/* $\mathcal{C}$ is the set of existing sessions */

/* $\pi_{G,j}$ ($\pi_{B,j}$) is the steady state probability of session j's channel being in *good* (*bad*) state */

<u>*STEP 1:*</u> /* Compute $(V_1, \ldots, V_{|\mathcal{C}|}, V_i)$ */

Define $\alpha_j = \frac{2q_j\pi_{B,j}}{1+q_j-p_j} + \pi_{G,j}$, $\beta_j = \frac{1+q_j-p_j}{q_j}$ and $\zeta_j = \frac{2q_j}{1+q_j}$ for very $j \in \mathcal{C} \cup \{i\}$.

(R1) Compute $\widehat{Z} = \min_{t \geq 0} \left\{ \sum_{j \in \mathcal{C} \cup \{i\}} g_j[0,t] \leq \left(1 - \sum_{j \in \mathcal{C} \cup \{i\}} \frac{\log(\beta_j)}{\log\left(\frac{\beta_j}{\zeta_j}\right)} \right) t \right\}$.

For every $j \in \mathcal{C} \cup \{i\}$, obtain

(R2) $\delta_j = \frac{\log(\beta_j)}{\log\left(\frac{\beta_j}{\zeta_j}\right)} + \frac{\log\left(\frac{\alpha_j}{\mu_j}\right)}{\widehat{Z}\log\left(\frac{\beta_j}{\zeta_j}\right)}$.

(R3) Compute $Z = \min_{t \geq 0} \left\{ \sum_{j \in \mathcal{C} \cup \{i\}} g_j[0,t] \leq (1 - \sum_{j \in \mathcal{C} \cup \{i\}} \delta_i)t \right\}$.

$V_j = \delta_j Z$;

<u>*STEP 2:*</u> /* Check the guaranteeability of $\vec{D} - \sum_{j \in \mathcal{C} \cup \{i\}} V_j$ under perfect channel*/

$V = \sum_{j \in \mathcal{C} \cup \{i\}} V_j$;

if ($\vec{D} - V$ is guaranteeable under perfect channel) **then**

Admit the session;

$\mathcal{C} = \mathcal{C} \cup \{i\}$;

else

Block the session i;

end

Fig. 4. Pseudo code of a general admission control algorithm for an error-prone wireless channel

slots preceding and succeeding the sequence.

Let session i seek admission, and $\mathcal{C}$ be the set of existing sessions. If for a session j the expected fraction of packets dropped in every busy period is less than μ_j, then the overall fraction of packets dropped for session j is also upper bounded by μ_j. The admission control algorithm admits i only if after i is admitted, each session j's expected drop rate in each busy period is upper bounded by μ_j, for every $j \in \mathcal{C} \cup \{i\}$. Let $\mathcal{B}$ be an arbitrary busy period. Let the *number of channel errors in a busy period* $\mathcal{B}$ of a session j (E_j) be the number of slots in $\mathcal{B}$ in which j's channel has bad state. A session needs to retransmit due to channel errors. Retransmissions for a session increase the delay for all sessions (Figure 1(b)). Intuitively, an upper bound on the delay experienced by a packet in the error-prone system is the sum of the delay under the perfect channel and $\sum_{j:j\in\mathcal{C}\cup\{i\}} E_j$. Here, *perfect channel* refers to a channel that does not suffer from channel errors, i.e., the channel is good with probability (w.p.) 1. Thus, if under perfect channel conditions, a packet can be successfully transmitted before its finish time, then under error-prone channels it can be successfully transmitted before its finish time plus $\sum_{j:j\in\mathcal{C}\cup\{i\}} E_j$. If there exists a number V such that in each busy period $\sum_{j:j\in\mathcal{C}\cup\{i\}} E_j \leq V$, then the guaranteeability of delay $D_j - V$ for each j in $\mathcal{C}\cup\{i\}$ under perfect channel implies that under error-prone channels every packet of session j is transmitted successfully before its deadline and hence zero packet drop for each j in $\mathcal{C}\cup\{i\}$. Thus session i can be admitted if the delay $D_j - V$ is guaranteeable for each session j in $\mathcal{C}\cup\{i\}$ under perfect channel, and blocked otherwise. The admission control algorithms in wireline case can be used to check whether the above condition holds. But, the problem is that since the channel errors are random the sum of the channel errors $\sum_{j:j\in\mathcal{C}\cup\{i\}} E_j$ in each busy period can not be upper bounded by a constant V. Nevertheless, for a large enough V, in each busy period $\sum_{j:j\in\mathcal{C}\cup\{i\}} E_j$ can be less than or equal to V with a high probability.

Let $\mu_{\min} = \min\{\mu_j : j \in \mathcal{C}\cup\{i\}\}$. Consider the following two conditions.

Condition 1: Delay $D_j - V$ is guaranteeable for each session j in $\mathcal{C}\cup\{i\}$ under perfect channel.

Condition 2: The probability that the number of channel errors in an arbitrary busy period $\mathcal{B}$ exceeds V is less than or equal to $\mu_{\min}$.

If conditions 1 and 2 hold, it turns out that the expected fraction of packet drop for each session is less than or equal to $\mu_{\min}$. Thus i can be admitted if there exists a V that satisfies conditions 1 and 2. The value of such V depends on the (a) channel characteristics of the sessions (b) maximum length of a busy period (c) $\mu_{\min}$ and (d) $(D_1, \ldots D_{|\mathcal{C}|}, D_i)$. The challenge in designing an admission control algorithm then is to determine whether such V exists. Wireline admission control algorithms [45], [46], [47], [48], [49] can be used to verify whether a given V satisfies condition 1. Verification of condition 2 requires a knowledge of an upper bound on the length of the busy periods in presence of channel errors and the joint channel statistics of the states of all channels. A strategy to compute the former is not known in the literature. The source can not compute the joint channel statistics from the individual channel characteristics it knows, unless it can estimate the statistical correlations among the channel states which is difficult in practice.

Another problem with the above admission control scheme is that it admits session i only if the expected packet drop rate of i and each existing session is less than $\mu_{\min}$. But note that $\mu_{\min}$ may be less than the expected packet drop rate requested by several sessions. Thus, the admission control condition has an unnecessarily high session blocking.

The above problems can be addressed if the system satisfies the following additional properties for some vector $(V_1, \ldots V_{|\mathcal{C}|}, V_i)$ such that $V = \sum_{j \in \mathcal{C} \cup \{i\}} V_j$.

Condition $2'$*:* The probability that the number of channel errors of j in $\mathcal{B}$ exceeds V_j is less than or equal to μ_j.

Condition 3 *("Decoupling condition"):* If the number of channel errors of j in every busy period is less than or equal to V_j, then each packet from session j that arrived in $\mathcal{B}$ is successfully transmitted before its finish time plus V.

Let the system satisfy conditions $1, 2'$ and 3 for some $(V_1, \ldots V_{|\mathcal{C}|}, V_i), V = \sum_{j \in \mathcal{C} \cup \{i\}} V_j$. It then turns out that the expected fraction of packet drop for each session j is less than or equal to the desired quantity μ_j for each $j \in \mathcal{C} \cup \{i\}$. Thus, i can be admitted if conditions $1, 2'$ and 3 hold. We now discuss how to verify whether conditions $2'$ and 3 hold.

Condition 3 ensures that once a session is admitted its packet drop guarantees does not depend on the number of channel errors of other sessions. This "decoupling" condition need not hold for all scheduling schemes. Consider the following example.

Example 1: Consider the wireless system shown in Figure 1(c). For each $i \in \{1, 2\}$, in any slot, session i's channel has a bad state w.p. p, independent of its channel state in other slots and the channel states of the other session. Session 3 has a perfect channel. Let $\mu_1 = \mu_2 = 1-(1-p)^3$, $\mu_3 = 0$. Session 3 however drops a packet every time session 1 has a bad channel in the slot in which 1 is served (Figure 1(c)). This has a non-zero probability. Thus, session 3 can not be guaranteed a zero packet drop rate in presence of sessions 1 and 2, and therefore should not be admitted. Since session 3 can drop packets even when it does not have any channel error, there does not exist a V_3 that satisfies Condition 3.

We now show how to modify the scheduling in Figure 1(c) so as to satisfy condition 3. This modification would also allow each session to satisfy its required delay and packet drop guarantees. Let the scheduler transmit packets with the minimum deadline as before, but now it drops a packet if the first transmission is not successful. Thus, each busy period has length 3. Let $V = 0$. Thus $V_i = 0$, for all i. Thus, $\vec{D} - V$ is guaranteeable under perfect channel and hence condition 1 holds. The probability that the number of channel errors of session i exceeding 0 in a busy period is at most μ_i, for each $i \in \{1, 2, 3\}$. Thus condition $2'$ holds. Condition 3 also holds as a packet is dropped only if its session has a slot in which its channel is erroneous. The source admits sessions $1, 2, 3$ simultaneously. Session 1 and 2 have drop rates of $p \leq 1 - (1 - p)^3$ and session 3 has a drop rate of 0. Thus, each session i has packet drop less than μ_i. ■

Recall that the scheduling among the sessions depends on the finish times of the packets and the number of retransmissions required by the sessions. We will choose the relation between the scheduling among the sessions and the finish times of the packets so as to ensure that condition 3 holds if conditions 1 and $2'$ hold. Now, the admission control algorithm in Figure 5 admits a session i only if conditions 1 and $2'$ hold in presence of i. We later discuss why decoupling is a good feature of any system. In Section V, we provide a general framework to obtain decoupling irrespective of the allocation of the finish times.

We now discuss the two major steps of the admission control algorithm (Figure 4). In STEP 1, the algorithm first computes a quantity Z that upper bounds the length of each busy period with high probability. Then, using the knowledge of Z, the algorithm computes $(V_1, \ldots, V_{|\mathcal{C}|}, V_i)$ such that condition $2'$ holds. In STEP 2, the algorithm checks condition 1, with $V = \sum_{j \in \mathcal{C} \cup \{i\}} V_j$. Note that whether $\vec{D} - V$ is guaranteeable under perfect channels can be checked using Network Calculus [46], [47] for several policies, e.g., EDF [36], FCFS, PGPS [37], SCFQ [39], VC [40], WF^{2}Q [41], SFQ [42], RPS policies [43] and the policies that promise certain service-curve [44]. The new session is admitted if both conditions 1 and $2'$ hold; otherwise it is blocked.

Now we discuss how in STEP 1 we obtain $(V_1, \ldots, V_{|\mathcal{C}|}, V_i)$ that satisfy condition $2'$. Suppose we obtain a function $\phi_j(T)$ (channel error rate) which satisfies the property that the number of channel errors for session j in duration T exceeds $T\phi_j(T)$ with probability at most μ_j. Let Z_1

denote an upper bound on the length of the busy period. Then, to satisfy condition $2'$, it suffices to choose $V_j = Z_1\phi_j(Z_1)$. Such $\phi_j(T)$ can be obtained using Chernoff bounds for any T. A problem, however, is that Z_1 is a random variable, and $\phi_j(Z_1)$ depends on Z_1. For example, a higher channel error rate corresponds to a longer busy period. Hence, we first obtain a lower bound $\widehat{Z}$ on Z_1, which depends only on the channel statistics (relation (R1) in Figure 4). Then, we set $\delta_j = \phi_j(\widehat{Z})$ (relation (R2) in Figure 5). Finally, we obtain an upper bound Z on the length of any busy period when decoupling condition (condition 3) holds with $V_j = \delta_j Z$ (relation (R3) in Figure 4). Since, $\phi_j(Z) \leq \phi_j(\widehat{Z})$ and $Z \geq \widehat{Z}$, $V_j \geq Z\phi_j(Z)$. Thus, the probability that the number of channel errors for session j exceeds V_j is at most μ_j. So, condition $2'$ holds for the $(V_1, \ldots, V_{|\mathcal{C}|}, V_i)$ obtained in STEP 1.

In the next section, we describe the scheduling framework.

V. Scheduling Framework

Several scheduling policies are known for wireline and wireless networks. We now present a framework that allows the system to use any desired scheduling policy and still attain the desired delay and packet drop guarantees for each admitted session. We first assume that the scheduler does not know the channel state of any session in a slot before deciding which session to serve in the slot. We later discuss how to utilize prior information about channel states, if such information is available.

The framework we propose decouples the performance of different sessions. More precisely, the framework ensures that irrespective of the arrival traffic and channel conditions of other sessions, an admitted session i does not experience packet drop in any busy period in which it has at most V_i channel errors. The parameter V_i is determined by the admission control algorithm (Figure 4) during i's admission, and depends on the maximum delay D_i and the maximum packet drop rate μ_i specified by i. This decoupling ensures that the quality of service guarantees of different sessions are satisfied. We first present a definition.

Definition 3: Compensation-slot of a session is a slot in which its packet is transmitted and its channel has a *bad* state.

A packet may be transmitted when the session has a bad channel as channel states are not known before transmission. If a session's packet is transmitted when its channel has a bad state, then the packet must be retransmitted or dropped. Retransmission causes additional delays and subsequently deadline expiry and packet drop. Figure 1(c) illustrates that packet drop of a session depends on the retransmissions of other sessions. We decouple the performance of different sessions by limiting the number of retransmissions of each session. Let a session i's packet be received in error. If the number of compensation-slots of a session i in a busy period is less than V_i, then the packet is retransmitted; otherwise the packet may be dropped depending on the system load. Thus, V_i can be looked upon as the retransmissions guaranteed to session i to compensate for its channel errors. This strategy ensures that excessive channel errors of a session does not lead to a large number of retransmissions and hence additional delays and packet drops for other sessions.

In Figure 5, we present the pseudo code for the scheduling framework. We explain the operations here. Each session i maintains two queues: a primary queue ($prim\_queue(i)$) and a compensation queue ($comp\_queue(i)$). New packets of a session join its primary queue. The sessions that have packets in their primary queue are called *primary-busy sessions*. Similarly, the sessions that have packets in their compensation queue are called *comp-busy sessions*. A *primary busy period* is a sequence of contiguous slots such that the system has at least one primary-busy session in each slot in the sequence, and no primary-busy session in the slots preceding and succeeding the sequence. Each session i maintains a *compensation-counter* ($Comp\_counter(i)$) which is the number of compensation-slots of the session in the current primary busy period. After the compensation-counter exceeds V_i, the packets that need retransmission are transferred to the session's compensation queue. When a primary busy period ends, packets from the compensation queues are transmitted in increasing order of their respective finish times, unless their deadlines are expired. We now discuss the algorithm is detail.

The scheduler assigns a finish time to each new packet. Let $p_{k,j}$ denote the k^{th} packet of the j^{th} session. Also, let $a_{k,j}$ and $f_{k,j}$ denote the arrival time and the finish time respectively for $p_{k,j}$. For example $f_{k,j} = a_{k,j} + D_j$ under EDF, and $f_{k,j} = \max\{a_{k,j}, f_{k,j}\} + \frac{1}{r_j}$ under VC, where

r_j is j's weight. Ordering between the packets of the same session is the same as that between their arrival times. Thus the Head of Line (HoL) packet in the session's primary queue has the minimum finish time among all packets in the session's primary queue. But the ordering among the finish times of packets of different sessions need not be the same as that between their arrival times. A new packet joins the session's primary queue.

The finish time $ft(j)$ for a primary-busy session j is the finish time of the HoL packet in the session's primary queue. The system transmits the HoL packet in the primary queue of the primary-busy session with the minimum finish time. Let this session be i. If the transmission is successful, the packet is removed from the system. Let the packet be received in error. The scheduler increments $Comp\_counter(i)$. If $Comp\_counter(i)$ is less than or equal to V_i, the packet is retained at the HOL position in the primary queue. If $Comp\_counter(i)$ exceeds V_i, the packet is transferred to the compensation queue.

When the system does not have a primary-busy session, the packets from the compensation queues of the sessions are transmitted in the increasing order of their finish times. The transmission from the compensation queues continue until a new packet arrives. In every slot, the system drops the packets from each session's compensation queue whose deadlines have expired.

The algorithm we have described does not deliver packets in order. For example, let a packet $p_{k,j}$ be transferred to j's compensation queue at time t. Let in slot $t + 1$ the session j be selected for service, $p_{k+1,j}$ be in its primary queue and j's channel have a *good* state. Then, $p_{k+1,j}$ that arrived after $p_{k,j}$ is delivered to the destination in slot $t + 1$. Now, $p_{k,j}$ will be delivered to the destination after $t + 1$, when j's primary queue becomes empty. A remedy is to transmit $p_{k,j}$ when j is scheduled next, i.e., in slot $t + 1$ in this case. But then j will have minimum finish time and therefore served in $t + 2$ as well. This leads to additional service for j that can cause additional delays for other sessions.

The following modification prevents excess retransmissions and delivers packets in order. As before, the primary-busy session with the minimum finish time is selected for service. Let this session be j. If j's compensation queue is empty when j is selected, then the HoL packet from j's primary queue is transmitted and the rest is as before. Let j's compensation queue have at least one packet, when j is selected. Now the HoL packet from j's compensation queue is transmitted. If the transmission is successful, then the HoL packet from j's primary queue is transferred to j's compensation queue, and the transmitted packet is removed from j's compensation queue. If the transmission is not successful, then $Comp\_counter(j)$ is incremented, and the transmitted packet is retained at the HoL position of the compensation queue. If $Comp\_counter(j) \geq V_j$, the HoL packet from j's primary queue is transferred to j's compensation queue, otherwise the HoL packet in j's primary queue remains in its position. The rest is the same.

We show that before a packet's deadline expires, it is either successfully transmitted or transferred to the compensation queue. In every slot, the system drops packets from the compensation queue whose deadlines have expired. Thus, packets are never transmitted after their deadlines. We show that μ_j upper bounds the expected fraction of packets dropped for session j.

Observe that though in a primary busy period we may transmit a packet from compensation queue, we select the session that receives service on the basis of finish times of packets in the session's primary queue. To summarize, *in a primary busy period, we schedule session j, if it has the smallest finish time among all the primary-busy sessions. But, the first packet from the primary or compensation queue for session j, whose deadline has not expired, is transmitted. When a system does not have a primary-busy session, it transmits packets in increasing order of finish times from the compensation queues of all sessions*. This prevents packets that represent additional retransmission from contending for channel access. Packets from compensation queues contend for channel access when the primary queues are empty; then unused resources can be used for additional retransmissions.

Finally, we describe how to modify the framework so as to use information about the instantaneous channel state, and improve the system performance, if such information is available. Let the source know the channel states of all sessions in the beginning of each slot. Now, in a slot in a primary busy period, a primary-busy session that has a good channel and has the minimum finish time among all the primary-busy sessions with good channels is selected for service. A compensation-slot for a session is one in which it is primary-busy, it has a bad channel and it has the minimum finish time among all primary-busy sessions. The compensation counter for a session is incremented in each of its compensation-slots. When the system has no primary-busy session, packets are transmitted in increasing order of their finish times from the sessions with good channels. The rest remains the same.

VI. Performance Evaluation

We present the analytical guarantees provided by the joint admission control and scheduling framework. All results in this section hold irrespective of whether instantaneous channel states are known before scheduling. In the appendix we present proofs for the case when scheduler does not know instantaneous channel states. The proofs for the case in which scheduler knows exact channel states for the sessions in every slots follows using the similar arguments.

Lemma 1: If a packet is transmitted successfully in a slot t, then its deadline is greater than or equal to t.

Lemma 1 ensures that a packet does not reach its destination after its deadline. We next show that the scheduling interface decouples the packet drop for various sessions even in presence of channel errors.

Lemma 2: If in every primary busy period the number of channel errors for session i is at most V_i, then there is no packet drop for session i. Further, every session i packet is successfully transmitted before the packet's finish times plus V.

Lemma 3: If the probability that the number of slots in which session i's channel has a bad state in a primary busy period exceeds V_i is at most μ_i, then the expected fraction of packets dropped for session i is at most μ_i.

Lemma 2 shows that a session i does not have any packet drop irrespective of the number of erroneous slots, if i's channel has at most V_i erroneous slots in each primary busy period. Now, the admission control algorithm admits session i if the number of channel errors for i in each primary busy period is less than V_i with probability μ_i. From Lemma 3 this ensures that session i's expected packet drop rate is μ_i. Let in each primary busy period the number of channel errors of an admitted session j be significantly larger than V_j. This may happen if j's channel parameters are not estimated correctly when j seeks admission. Lemma 3 ensures that in this case only j's guarantees are violated. Several existing scheduling policies do not have this desirable feature. This is because existing scheduling policies serve the session that has the packet with the minimum finish time in the system. Thus, if a session has several channel errors, then its packets remain in the system for long durations, but are nevertheless repeatedly selected for service because of low finish times. This delays the transmission of packets of other sessions as well. Our framework serves the session with the minimum finish time, where the finish time of a session is the finish time of the HoL packet in the session's primary queue. When a session has several channel errors, its packets with low finish times are transferred to the compensation queue, and are not considered while deciding which session to serve. The following experiment illustrates this feature.

Consider a system with two sessions. Let the estimated channel parameters for the sessions be p_1 = p_2 = 0.999 and q_1 = q_2 = 0.9. These channel parameters correspond to a Raleigh fading channel that has good channel 99% time and mean fade duration 10 slots [10]. Let the actual channel parameters for session 1 equal the estimated values, but the actual channel parameters for session 2 be different from the estimated values. We consider leaky bucket constrained sources. We assume $\sigma_i = 10$, $\rho_i = 0.3$, $\mu_i = 10^{-3}$ and $D_i = 30$ for $i \in \{1, 2\}$. Let the finish time of each packet equal its deadline (EDF). We study the fraction of packets dropped for the sessions under our framework (Table I) (augmented-EDF), and when the packet with the minimum deadline is served (Table II) (plain-EDF). We run simulations for 10^6 slots. The column $1 - \hat{q}_2$ refers to the actual value of channel parameter for session 2. Under plain-EDF packet drops for both the sessions increase with the increase in the difference between session 2's actual value of the channel parameter and the estimated value (Table II). But under our framework packet drop increases only for session 2 and is almost constant for session 1 (Table I). Thus, unlike under plain EDF, in our framework the required packet drop guarantee is always provided to session 1 irrespective of the channel conditions for session 2.

Finally, in the following theorem we show that the joint admission control and scheduling framework provides the required QoS guarantees.

TABLE I

PACKET DROP WITH EDF SCHEDULING UNDER THE PROPOSED FRAMEWORK

$1-\widehat{q}_2$	Packet Drop for Session 1 (%)	Packet Drop for Session 2 (%)
0.1	0.01647	0.02987
0.075	0.01560	0.09093
0.050	0.01773	0.29313
0.025	0.01933	1.24013

TABLE II

PACKET DROP WITH PLAIN EDF SCHEDULING

$1-\widehat{q}_2$	Packet Drop for Session 1 (%)	Packet Drop for Session 2 (%)
0.1	0.02713	0.03467
0.075	0.05440	0.06787
0.050	0.15160	0.17940
0.025	0.60987	0.66433

Theorem 1: In the joint admission control and scheduling framework (Figures 4 and 5), the expected fraction of packets dropped for every admitted session i is at most μ_i.

Thus, the reservations provided by our framework are sufficient to guarantee the desired delay and loss guarantees. Using an example, we now demonstrate that the reservations are also necessary. Specifically, the following example shows that if a session i is admitted when there does not exist a $\vec{V}$ that satisfies conditions 1 and 2', then the expected fraction of packets dropped for i or an existing session j may exceed μ_i or μ_j respectively. Thus, the guarantees provided by the framework are stochastically tight.

Example 2: Consider the arrival pattern shown in Figure 1(c). Let sessions 1 and 2 be in the system when session 3 seeks admission. The delay requirement for each session is 3. Thus, busy period length is at most 3 irrespective of whether session 3 is admitted. Sessions 1 and 2 have i.i.d. channel errors. The probability that session 1 has *bad* channel in any slot is p, $1/2 < p < 1$. Let $\mu_1 = 3p^2(1-p) + (1-p)^3$ and $\mu_2 = (1-p)^3$. Session 3 has a perfect channel and $\mu_3 = p/2$. Before session 3 seeks admission, $V_1 = 1$ and $V_2 = 0$ satisfy conditions 1 and $2'$. Session 3 is not admitted as there does not exist V_1, V_2 and V_3 that can satisfy conditions 1 and $2'$.

Suppose the system admits session 3. Now, we show that if session 3 is admitted, then the drop guarantees for at least one session are violated under any assignment of V_i's. Under any assignment of V_i's, there are two possibilities, (a) $V_1 = 0$ and (b) $V_1 > 0$. Now, (a) violates $2'$ for session 1. Here, session 1's packet will be dropped if 1 has a channel error in slot 0. Thus, the expected fraction of packets dropped for session 1 is p. Since $p > 3p^2(1-p) + (1-p)^3$, session 1's drop guarantee is violated. Next, (b) violates condition 1. Here, session 3's packet will be dropped if session 1 has a channel error in slot 0. Thus the expected fraction of packets dropped for session 3 is p. Since, $p > p/2$, session 3's packet drop guarantee is violated. ■

We have demonstrated that the reservations provided by our framework are necessary in statistical worst case scenarios. The specific worst cases we considered are the following.

1) The framework provides reservations based on the assumption that the channel errors for various sessions occur in distinct slots.
2) The framework assumes that if any session has bad channel, then the server remains idle.

If, however, these worst cases do not occur, then our framework is conservative in a sense that it may have session blocking higher than the minimum required for providing the required statistical guarantees to the admitted sessions. Nevertheless, considering the statistical worst cases is imperative from practical considerations. We now explain why this the case. The channel errors for different sessions may not always occur in different slots. But, this fact can only be exploited (i.e., (1) can be relaxed) if the correlations between the channel error processes of different sessions are known precisely. The correlations between the error processes can be determined only when all statistical moments of the joint error processes are known. Estimating all moments of the joint error processes is difficult, if not impossible, in practice. In practice, a node may only be able to estimate the first order statistics (expectations) of the individual error processes. Thus, a node needs to provide statistical guarantees using only the first order statistics (expected error rates), and hence needs to consider the worst possible correlations of the error processes of different sessions in providing such guarantees. This is exactly what our framework does. Also, to relax assumption (2), knowledge of the correlation between arrival and channel error processes is required. There exists arrival and channel error patterns such that (2) is true for any scheduling policy (Figure 1 shows an example for EDF scheduling policy). Again, estimating the correlation between the arrival and channel error processes is difficult in practice, and hence the worst possible correlations between these must be assumed for providing statistical guarantees.

Furthermore, we expect that the admission control schemes will be executed by service providers. Oftentimes, service providers have service contracts with the customers which obliges the providers to provide the mutually agreed service guarantees. In these cases, even minor violations of the mutually agreed guarantees require the provider to refund the entire service fee which eventually results in a financial loss. The service providers often therefore prefer to admit the sessions only when they obtain the required service even under the worst possible statistical combinations, which in turn requires them to use conservative schemes. Specifically, a customer i provides a delay requirement of D_i and a loss requirement of μ_i. It does not accept a packet if it is delivered D_i units after it is generated, and considers such a packet to be lost. A customer demands a refund if more than μ_i fraction of its packets are lost. Since the channel error process is stochastic, more than μ_i fraction of i's packets may be lost for any given admission control and scheduling scheme. Thus, the provider may indeed have to refund service fees for some customers, irrespective of what it does. But, we show that by considering worst possible statistical combinations, the provider forfeits fees only with negligible probability. This is because drop probability for each packet is at most μ independent of the other packets. In other words, if $L_{i,k}$ is the random variable that takes value 1 if k^{th} packet of i^{th} session is dropped and 0 otherwise, then $L_{i,k}$ are iid Bernoulli random variables. Thus, the probability that the fraction of session i's packets dropped exceeds μ goes to zero exponentially fast [50]. Thus, we expect them to find our scheme useful.

In the following lemma, we prove one more useful property of our framework.

Lemma 4: Packets from a session reach the session's destination in increasing order of their arrivals.

All the analytical guarantees hold for more general Markovian models (e.g., with more than 2 states [51]). In these Markovian models, the state u corresponds to a loss probability of ℓ_u, where $0 \leq \ell_u \leq 1$. The only distinction in the analysis is that the Chernoff bounds can no longer be used to upper bound the expected number of channel errors in a busy period. The upper bound can, however, be obtained using large deviation theory. The analysis is more involved and lengthy, but remains essentially the same.

Finally, the computational complexity of our framework depends on the scheduling algorithm and the arrival processes. Depending on the scheduling algorithm and the arrival process, the scheme may require per-flow states. For example when the scheduling is either EDF or PGPS, and the arrivals are either leaky bucket constrained or Constant Bit Rate (CBR), the computational complexity of the framework is $O(N^2)$, where N is the number of sessions active at a node. But, N is typically small, and is not likely to increase significantly due to the limitations in available bandwidth. Hence, maintaining per flow states is not a serious deterrent, and the framework is computationally feasible. If, however, N does become large in future, then existing or new scheduling algorithms that do not need per flow states will be implemented. Now, the admission control algorithms may still need per flow states. But, these states will be used only when a session arrives or departs, and not during the transmission of each packet. Thus, the computational complexity of our framework remains low even for large N.

VII. Conclusion

Summarizing, we have proposed a joint framework for admission control and scheduling that delivers desired delay and packet drop guarantees to sessions traversing error-prone wireless channels. Such framework can only be developed if the amount of resources that need to reserved for each session can be quantified. Such quantification is non-trivial, e.g., even if only one session is admitted, it is not necessary that its delay and loss guarantees will be satisfied particularly when the wireless channel has poor transmission quality and the delay and loss guarantees are stringent. Thus, determining how much reservation will suffice is critical, unless the system chooses the trivial policy of rejecting all sessions (i.e., 100% call blocking ratio). Using large deviation bounds and renewal theory, we provide a systematic approach for such quantification. The framework can accommodate scheduling policies from a large class and obtain admission control algorithms for any given scheduling in this class and different arrival processes. We prove that once a session is admitted, irrespective of the scheduling policy, the session's expected packet drop rate is below its acceptable threshold, and the packets that are not dropped are delivered before their deadlines. The proposed framework is robust since an inaccurate estimation of channel statistics does not affect the performance of other existing sessions in the system. Furthermore, the admission control only uses individual channel statistics of the sessions and not joint channel statistics, and the scheduling need not know the instantaneous channel state. The framework can, however, use the knowledge of the instantaneous channel states, if these are known, and improve the performance further.

REFERENCES

[1] J Kurose and K. Ross, *Computer Networking: A top-down approach featuring the Internet*, Addison-Welsley Longman, 2nd edition edition, 2003.

[2] V. Bharghavan, S. Lu, and T. Nandagopal, "Fair queueing in wireless networks: Issues and approaches," *IEEE Personal Communications Magazine*, Feb 1999.

[3] S. Lu, V. Bharghavan, and R. Shrikant, "Fair scheduling in wireless packet networks," in *ACM SIGCOMM*, 1997.

[4] S. Lu, T. Nandagopal, and V. Bharghavan, "Design and analysis of an algorithm for fair service in error-prone wireless channels," *Wireless Networks*, pp. 323–343, 2000.

[5] T.S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location dependent errors," Tech. Rep. CMU-CS-00-112, Carnegie Mellon University, 2000.

[6] M. Krunz and J. G. Kim, "Fluid analysis of delay and packet discard performance for qos support in wireless networks," *IEEE J. on Selected Areas in Communications*, vol. 19, no. 2, Feb 2001.

[7] S. Wang, R. Nathuji, R. Bettati, and W. Zhao, "Providing statistical delay guarantees in wireless networks," in *ICDCS 2004*, Mar.

[8] S. Ayyorgun and R. L. Cruz, "A composable service model with loss and a scheduling algorithm," in *INFOCOM'04*, 2004.

[9] S. Ayyorgun and R. L. Cruz, "A service curve model with loss and multiplexing problem," in *ICDCS'04*, 2004.

[10] S. Shakkottai and R. Shrikant, "Scheduling real time traffic with deadlines over a wireless channel," *ACM/Beltzer Wireless Networks Journal*, 2001.

[11] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queue with randomly varying connectivity," *IEEE Trans. Info. Theory*, vol. 30, no. 2, 1993.

[12] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," Translations of the AMS, A volume in memory of F. Karpelevich, Providence, R.I.: American Mathematical Society, 2001.

[13] M. Goyal, A. Kumar, and V. Sharma, "Power constrined and delay optimal policies for scheduling transmission over a fading channel," in *INFOCOM'03*, 2003.

[14] J. G. Kim and M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet loss performance," *IEEE Trans. on Vehicular Technology*, vol. 49, no. 5, Sep 2000.

[15] R. Jain and E. W. Knightly, "A framework for design and evaluation of admission control algorithms in multi-service mobile networks," in *INFOCOM'99*, 1999.

[16] David Julian, Mung Chiang, Daniel O'Neill, and Stephen Boyd, "Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," in *INFOCOM02*, 2002.

[17] Bahareh Sadegi, "Architecture and algorithms for scalable mobile qos," M.S. thesis, Rice University, 2000.

[18] D. Shen and C. Ji, "Admission control of multiple traffic in third generation cdma networks," in *INFOCOM'00*, 2000.

[19] D. Shen and C. Ji, "Capability based admission control for broadband cdma networks," in *VTC'01*, 2001.

[20] C. Chao and W. Chen, "Connection admission control for mobile multiple-class personal communications networks," *IEEE J. on Selected Areas in Communications*, vol. 15, no. 8, pp. 1618–1626, 1997.

[21] D. Levine, I. Akyildiz, and M. Naghshineh, "A resource estimation and call admission control algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Trans. on Networking*, vol. 5, no. 1, pp. 1–12, Feb. 1997.

[22] A. Talukdar, B. Badrinath, and A. Acharya, "On accommodating mobile hosts in an integrated services packet networks," in *INFOCOM'97*, 1997.

[23] A. Acampora and M. Naghshineh, "An architecture and methodology for mobile-executed handoff in cellular atm networks," *IEEE J. on Selected Areas in Communications*, vol. 12, no. 8, pp. 1365–1375, Oct. 1994.

[24] A. Acampora and M. Naghshineh, "Qos provisioning in micro-cellular networks supporting multimedia traffic," in *INFOCOM'95*, 1995.

[25] A. Aljadhai and T. Zhanti, "A framework for call admission control and qos support in wireless environments," in *INFOCOM'99*, 1999.

[26] P. Bhagwat, A. Krishna, and S. Tripathi, "Enhancing throughput over wireless lans using channel state dependent packet scheduling," in *INFOCOM*, 1996, pp. 1133–1140.

[27] Y. Cao and V.O.K. Li, "Scheduling algorithms for broadband wireless networks," *Proc. of IEEE*, vol. 89, no. 1, pp. 76–87, Jan 2001.

[28] D. A. Eckhardt and P. Steenkiste, "Effort limited fair (elf) scheduling for wireless networks," in *INFOCOM 2000*, 2000, pp. 1097–1106.

[29] C. Fragouli, V. shivaraman, and M. Shrivastava, "Controlled multimedia wireless link sharing via enhanced class-based queueing with channel-state dependent packet scheduling," in *INFOCOM*, 1998, pp. 572–580.

[30] J. Gomez, A. T. Campbell, and H. Morikawa, "The havana framework for supporting application and channel dependent qos in wireless networks," in *ICNP'99*, 1999, pp. 235–244.

[31] S. Shakkottai and A. L. Stolyar, "Scheduling algorithm for a mixture of real-time and non-real-time data in hdr," in *ITC 2001*, 2001.

[32] P. Ramanathan and P. Agrawal, "Adapting fair queueing algorithms to wireless systems," *Proceedings of MOBICOM*, 1998.

[33] R. L. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[34] E. N. Gilbert, "Capacity of burst-noise channel," *Bell Syst. Technical Journal*, vol. 39, no. 8, pp. 1253–1265, Sep 1960.

[35] E. O. Elliaot, "Estimation of error rates for codes on burst-noise channels," *Bell Syst. Technical Journal*, vol. 42, no. 9, pp. 1977–1997, Sep 1963.

[36] C.L. Liu and J.W. Layland, "Scheduling algorithms for multiprogramming in a hard real time environment," *J. of ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[37] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case.," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[38] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," *Internetworking Research and Experience*, vol. 1, 1990.

[39] S. J. Golestani, "A self-clocked fair queueing scheme for high speed applications," in *INFOCOM'94*, Apr. 1994, pp. 636–646.

[40] L. Zhang, "Virtualclock: A new traffic control algorithm for packet switching networks," in *Proceeddings of ACM SIGCOMM'90*, Aug 1990, pp. 19–29.

[41] J. C. R. Bennett and H. Zhang, "wf^2q : Worst-case fair weighted fair queueing," in *INFOCOM'96*, Mar. 1996, pp. 120–127.

[42] P. Goyal and H. M. Vin, "Generalized guaranteed rate scheduling algorithms:a framework," *IEEE/ACM Trans. on Networking*, vol. 5, no. 4, pp. 561–571, Aug. 1997.

[43] D. Stiliadis and A. Varma, "Rate proportional servers: A design methodology for fair queueing algorithms," *IEEE/ACM Trans. on Networking*, vol. 6, no. 2, pp. 164–174, Apr. 1998.

[44] H. Sariowan, R.L. Cruz, and G.C. Polyzos, "Sced: A generalized scheduling policy for guaranteeing quality-of-service," *IEEE/ACM Trans. on Networking*, vol. 7, no. 5, pp. 669–684, Oct. 1999.

[45] H. Zhang, "Service desciplines of guaranteed performance service in packet-switching networks," *Proceedings of The IEEE*, vol. 83, no. 10, pp. 1374–1396, Oct. 1995.

[46] R. Agrawal and R. Rajan, "Performance bounds for guaranteed and adaptive services," Tech. Rep., IBM, 1996.

[47] J.Y. Le Boudec and G. Hebuterne, "Application of network calculus to guaranteed service networks," *IEEE/ACM Trans. on Networking*, vol. 8, no. 3, 1998.

[48] H. Sariowan, *A Service-Curve Approach to Performance Guarantees in Integrated-Service Networks*, Ph.D. thesis, University of California, San Diego, 1996.

[49] P. Chaporkar, "An approach to the analysis of scheduling policies for guaranteeing delay with arbitrary arrivals," M.S. thesis, Indian Institute of Science, Bangalore, India, 2000, http://shravana.cedt.iisc.ernet.in/∼scprasan.

[50] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis: Queues, communications, and computing*, CRC Press, 1995.

[51] Q. Zhang and S. A. Kassam, "Finite-state markov model for rayleigh fading channel," *IEEE Trans. on Communications*, vol. 47, no. 11, pp. 1688–1692, Nov 1999.

APPENDIX

The appendix is arranged as follows. First, we introduce notations. Then we state the supporting lemmas (Lemma 5, 6, 7 and 8) used to prove the performance guarantees (Lemmas 1, 2, 3 and 4, and Theorem 1). Next, we prove Lemmas 1, 2, 3 and 4, and Theorem 1. Finally, we prove the supporting lemmas.

NOTATION

In this section, we discuss important notation and terminology used in the proofs. We denote the joint admission control and scheduling scheme proposed in Sections IV and V as EC. We consider two fictitious systems: PC and E. The system PC does not have channel errors, and is the same as a wireline system. The system E is similar to EC, except that E does not have compensation queues and hence each session i's packet is dropped in E whenever the number of compensation slots for i exceeds V_i in a busy period. For convenience, in Figure 6 we give the pseudo code for the scheduling under E.

We first obtain performance guarantees like delay and packet drop in systems PC and E. Then we establish relations between the performances under EC and the fictitious systems. Using these relations we prove Lemmas 1, 2, 3 and 4, and Theorem 1.

The systems EC, PC and E have the following features.

(A1) All these systems have the same set of sessions and also the same arrivals.

(A2) The finish times of packets are the same under PC and EC. The finish time of a packet under E equals the finish time under PC plus V.

(A3) The channel state for every session i is the same in every slot t under systems EC and E. Under PC channel state is always *good*.

(A4) Under PC and E, a packet with the smallest finish time is transmitted in every slot. If multiple packets have the same finish time, then the packet from the session with the smallest index is transmitted. EC uses scheduling as shown in Figure 5.

TABLE III

NOTATIONS USED IN THE APPENDIX

Notation	Definition
$\mathcal{C}$	A set of sessions $1, 2, \ldots, n$ that are present in the system.
$p_{k,i}$	The k^{th} packet of session i.
$a_{k,i}$	The arrival time of $p_{k,i}$.
$f_{k,i}$	The finish time of $p_{k,i}$.
$d_{k,i}$	The departure time of $p_{k,i}$. A packet departs when it is either dropped or transmitted successfully.
$\gamma[t, t+\tau]$	The number of packets that arrived in the interval $[t, t+\tau]$.
$w[t, t+\tau]$	The number of packets that arrived in the interval $[t, t+\tau]$ with finish time less than or equal to $t+\tau$.
$S[t, t+\tau]$	The number of packets transmitted successfully in the interval $[t, t+\tau]$.
$\widehat{S}[t, t+\tau]$	The number of packets that departed from the system in the interval $[t, t+\tau]$.

The basic notations used in the appendix are defined in Table III. These symbols will be augmented to denote various dependencies. A superscript EC, PC or E will indicate quantities corresponding to the systems EC, PC and E respectively. Further, subscript i will indicate quantities corresponding to session i. Absence of subscript will indicate that the quantities correspond to the whole system. For example $S_i^{PC}[t, t+\tau]$ is the number of successfully transmitted packets of session i under PC in the interval $[t, t+\tau]$, and $S^{PC}[t, t+\tau] = \sum_{i \in \mathcal{C}} S_i^{PC}[t, t+\tau]$.

SUPPORTING LEMMAS USED IN THE PROOFS OF RESULTS IN SECTION VI

Here, we state the supporting lemmas (Lemmas 5, 7, 6 and 8) that we use to prove Lemmas 1, 2, 3 and 4, and Theorem 1. We prove each supporting lemma in a separate subsection latter. In Lemmas 5, 6 and 7 we establish relation between PC and E. In Lemma 5, we show that successful transmissions under E achieve the required QoS if such is achieved under PC. In Lemmas 6 and 7, we stochastically bound the number of packet drops under E. Thus most of the packets achieve the required QoS under E. Finally, in Lemma 8 we relate the performances under E and EC.

Lemma 5: If all the packets are successfully transmitted before their respective finish times under PC, then all the packets that are successfully transmitted also depart before their respective finish times under E.

Lemma 6: Let n_i and $\widehat{n}_i^E$ denote the number of packets generated by session i and the number of session i packets that can not be transmitted successfully before their finish times in E respectively. Then, $\mathbf{E}\left[\frac{\widehat{n}_i^E}{n_i}\right] \leq \mu_i$ if (a) delay $D_j - V$ is guaranteeable to every packet in PC for every session $j \in \mathcal{C}$ and (b) the probability that the number of channel errors of i exceeds V_i in any busy period under E is less than or equal to μ_i.

Lemma 7: Let $\alpha_i = \frac{2q_i\pi_{B,i}}{1+q_i-p_i} + \pi_{G,i}$, $\beta_i = \frac{1+q_i-p_i}{q_i}$, and $\zeta_i = \frac{2q_i}{1+q_i}$. Recall that $g_i(t)$ denote the maximum number of packets that can arrive in the interval $[0, t]$ for session i. Also, let

$$Z = \min_{t\geq 0}\left\{t \;:\; \sum_{i\in\mathcal{C}} g_i(t) \leq (1-\sum_{i\in\mathcal{C}}\delta_i)t\right\}, \quad \text{and} \tag{2}$$

$$\delta_i = \frac{\log(\beta_i)}{\log\left(\frac{\beta_i}{\zeta_i}\right)} + \frac{\log\left(\frac{\alpha_i}{\mu_i}\right)}{\widehat{Z}\log\left(\frac{\beta_i}{\zeta_i}\right)}, \qquad \text{where} \tag{3}$$

$$\widehat{Z} = \min_{t\geq 0}\left\{t \;:\; \sum_{i\in\mathcal{C}} g_i(t) \leq \left(1 - \sum_{i\in\mathcal{C}}\frac{\log(\beta_i)}{\log\left(\frac{\beta_i}{\zeta_i}\right)}\right)t\right\}. \tag{4}$$

Then the probability that the number of channel errors of i lots in which i in a busy period under E exceeds $V_i = \delta_i Z$ is at most μ_i.

Lemma 8: (C1) A packet $p_{k,i}$ has the smallest finish time under E at time t *iff* $p_{k,i}$ has the smallest finish time among the packets in primary queues under EC at time t.

(C2) In every slot t, the value of compensation counter is the same under E and EC for every session $i \in \mathcal{C}$.

(C3) If E transmits a packet $p_{k,i}$ successfully in slot t, then EC transmits a packet $p_{k_1,i}$ successfully in slot t where $k_1 \leq k$. If $k_1 < k$, $p_{k_1,i}$ is transmitted from i's compensation queue.

(C4) A packet $p_{k,i}$ departs from system E (either transmitted successfully or dropped) at time t *iff* the packet departs i's primary queue (either transmitted successfully or transfered to i's compensation queue) under EC at time t.

Proofs of Results in Section VI

In this section we prove performance guarantees provided by the joint admission control and scheduling framework (Figures 4 and 5). Hence, $\mathcal{C}$ is a set of sessions that are admitted by the admission control algorithm in Figure 4. First, we present a lemma which is used in many of the following proofs.

Lemma 9: (O1) Under PC, every packet $p_{k,i}$ is successfully transmitted before its finish time $f_{k,i}^{PC}$.

(O2) Under EC, the deadline for every packet $p_{k,i}$ is greater than or equal to the packet's finish time $f_{k,i}^{EC}$ plus V.

Proof: The admission control algorithm (Figure 5) ensures that $D_i - V$ is guaranteeable under PC for every $i \in \mathcal{C}$. Thus, (O1) follows from Definition 1. Also from Definition 1, every packet $p_{k,i}$ satisfies $f_{k,i}^{PC} \leq a_{k,i} + D_i - V$. Thus from (A1), $f_{k,i}^{EC} + V \leq a_{k,i} + D_i$. Hence (O2) follows. ■

A. *Proof of Lemma 1*

Proof: Under EC, let $p_{k,i}$ be transmitted successfully in slot t. If $p_{k,i}$ is transmitted from i's compensation queue, then the scheduling ensures that the deadline for the packet is greater than t. If $p_{k,i}$ is transmitted from i's primary queue, then $p_{k,i}$ has the smallest finish time among the packets present in the primary queues. From (C1) in Lemma 8, $p_{k,i}$ has the smallest finish time under E in slot t. Thus, it is transmitted under E. The channel state for session i is *good* in slot t as the packet transmission is successful under EC in slot t. So, the packet transmission is successful under E as well. Now, all successful transmissions depart before their respective finish times under E (Lemma 5 and (O1) in Lemma 9). Thus, $t \leq f_{k,i}^{E}$. Now, the result follows from (5) and (O2) in Lemma 9. ■

B. *Proof of Lemma 2*

Proof: From the given condition, in every primary busy period under EC the number of i's channel errors is upper bounded by V_i. Under EC, the number of i's compensation slots in a primary busy period is always less than or equal to i's channel errors in the busy period (Definition 3). Thus, the number of i's compensation slots is upper bounded by V_i in every primary busy period. Now, if a compensation queue is empty at the beginning of a primary busy period, then a packet for session i is transfered from the primary queue to the compensation queue only if the number of i's compensation slots is greater than V_i in the primary busy period (Figure 5). Since the compensation queue for session i is empty at the beginning of the first primary busy period, it can be seen using an induction argument that none of the session i's packets are transfered to its compensation queue. Thus, each of the session i's packets is transmitted from i's primary queue. Since, packets are never dropped from the primary queue (Figure 5), we conclude that all of the i's packets are transmitted successfully. Now, from (C4) in Lemma 8, a packet $p_{k,i}$ is transmitted successfully under E in slot t *iff* $p_{k,i}$ is transmitted successfully under EC in t. The result follows from (O1) in Lemma 9 and Lemma 5. ■

C. *Proof of Lemma 3*

Proof: From (A1), arrivals under E and EC are the same. Hence, from (C4) in Lemma 8 it follows that the busy periods under E and the primary busy periods under EC coincide, i.e., if the v^{th} busy period under E starts at b_v and finishes at B_v, then so does the v^{th} primary busy period under EC for every v. Hence, if the probability that the number of i's channel errors exceeds V_i in a primary busy period under EC is at most μ_i, then the probability that the number i's channel errors exceeds V_i in a busy period under E is at most μ_i. Since every $i \in \mathcal{C}$ is admitted by the admission control algorithm, $\vec{D} - V$ is guaranteeable under PC. Thus from Lemma 6, the expected fraction of i's packets dropped in E is at most μ_i. From (C3) in Lemma 8, the number of i's packets transmitted successfully under EC is greater than or equal to that under E. Thus, the result follows. ■

D. *Proof of Theorem 1*

Proof: Recall that $\mathcal{C}$ is a set of sessions that are admitted after verifying the admission control condition (refer Figure 4). Thus, $V_i = \delta_i Z$ for every $i \in \mathcal{C}$, where Z and δ_i are as in (2) and (3) respectively. Furthermore, under PC delay $D_i - V$ is guaranteeable for every $i \in \mathcal{C}$. Now, from (A1), E and EC have the same set of sessions, and from (A3) the channel state for every session is the same under both the systems. From Lemma 7, the probability that the number of i's channel errors is in a busy period under E exceeds V_i is at most μ_i. From (C4) in Lemma 8, the busy periods under E and the primary busy periods under EC coincide. Hence, the probability that the number of i's channel errors in a primary busy period under EC exceeds V_i is also at most μ_i. The result follows from Lemma 3. ■

E. *Proof of Lemma 4*

Proof: Let $k_1 < k_2$. It suffices to show that in any slot t, $p_{k_2,i}$ can not be successfully transmitted in EC if $p_{k_1,i}$ has not already been transmitted successfully or dropped. The packets depart the primary and compensation queues in increasing order of their arrivals, i.e., if p_1 enters i's primary (compensation) queue before p_2, then p_1 departs from the primary (compensation) queue before p_2 (Figure 5). Clearly the result holds if $p_{k_2,i}$ is dropped. Hence, we only consider that $p_{k_2,i}$ is successfully transmitted. First, let $p_{k_2,i}$ be transmitted from i's primary queue in slot t. Then, $p_{k_2,i}$ is at the HoL position in i's primary queue, and i's compensation queue is empty in t. Thus, $p_{k_1,i}$ is not in the system at t. Next, let $p_{k_2,i}$ be transmitted from i's compensation queue in t. Then, $p_{k_2,i}$ is the HoL packet in i's compensation queue after expunging i's packets whose deadlines have expired. Thus, $p_{k_1,i}$ is not in i's compensation queue, and also not in i's primary queue, since $p_{k_2,i}$ has already left the primary queue. Thus, $p_{k_1,i}$ is not in the system at t.

Thus, in both the cases, $p_{k_2,i}$ is transmitted in t only if $p_{k_1,i}$ is not present in the system in t. ■

PROOFS FOR SUPPORTING LEMMAS

F. *Proof of Lemma 5*

Using Lemma 10, we first establish a relation between $w_i^{PC}[t, t+\tau]$ and $w_i^E[t, t+\tau]$. Recall that $w_i[t, t+\tau]$ is the number of packets generated by session i in the interval $[t, t+\tau]$, which have finish times less than or equal to $t+\tau$. Also, by (A2)

$$f_{k,i}^E = f_{k,i}^{PC} + V = f_{k,i}^{EC} + V, \quad \text{for every } k \text{ and session } i. \tag{5}$$

The following lemma is a direct consequence of these definitions.

Lemma 10: For every $i \in \mathcal{C}$

$$w_i^{PC}[t, t+\tau] = w_i^E[t, t+\tau+V].$$

The scheduling under E does not check for deadline violation. Under E, a session i packet is dropped only when i's channel errors in a busy period exceeds V_i. Lemma 5 guarantees that every successfully transmitted packet reaches its destination before its deadline. A formal proof follows.

Proof: We use the following result from [49], [48].

Result from [48]: Let $W^{PC}[t, t+\tau]$ represent the maximum value of $w^{PC}[t, t+\tau]$ under any arrival pattern that conforms to the traffic characterization. Then, all the packets arriving in a system without channel errors depart before or at their respective finish times iff

$$W^{PC}[t, t+\tau] \leq \tau \quad \forall\, t \ \text{ and } \tau > 0. \tag{6}$$

We prove Lemma 5 using contradiction. Let there exists a packet $p_{k,i}$ such that $p_{k,i}$ is transmitted successfully after its finish time under E, i.e $d_{k,i}^E > f_{k,i}^E$.

Let T_k denotes the starting epoch of the busy period under E in which $p_{k,i}$ arrives.

In the busy period, it is possible that the packets with finish time greater than $f_{k,i}^E$ are transmitted in certain slots. This may happen if none of the sessions have packets with finish times less than or equal to $f_{k,i}^E$.

We define $\hat{T}_k$ to be the smallest time instant such that

(T1) $\hat{T}_k \in [T_k, a_k]$, and

(T2) no packet with finish time greater than $f^E_{k,i}$ is transmitted in the interval $[\hat{T}_k, d^E_{k,i}]$.

Note that $\hat{T}_k$ always exists. This is because a packet with the smallest finish time is transmitted in every slot under E, and between $a_{k,i}$ and $d^E_{k,i}$ packer $p_{k,i}$ with finish time $f^E_{k,i}$ is present in the system. Thus, no packet with finish time greater than $f^E_{k,i}$ is transmitted in $[a_{k,i}, d^E_{k,i}]$.

Since $d^E_{k,i} > f^E_{k,i}$, in $[\hat{T}_k, f^E_{k,i}]$, only packets with the finish time less than or equal to $f^E_{k,i}$ are transmitted.

Now we show that $w^E[\hat{T}_k, f^E_{k,i}] \geq \widehat{S}^E[\hat{T}_k, f^E_{k,i}]$, i.e., every packet transmitted in $[\hat{T}_k, f^E_{k,i}]$ contributes to $w^E[\hat{T}_k, f^E_{k,i}]$. From the property (T2) in the definition of $\hat{T}_k$, the packets transmitted in $[\hat{T}_k, f^E_{k,i}]$ have finish times less than or equal to $f^E_{k,i}$. Using contradiction we now show that these packets also arrive in $[\hat{T}_k, f^E_{k,i}]$. Let a packet $p_{u,j}$ arrive before $\hat{T}_k$, and be transmitted in a slot $t \in [\hat{T}_k, f^E_{k,i}]$. Thus, its finish time is less than $f^E_{k,i}$. Since the scheduler always transmits packets in the increasing order of their finish times, no packet with finish time greater than $f^E_{u,j}$ can be transmitted in $[a_{u,j}, t]$. Since $f^E_{k,i} \geq f^E_{u,j}$, no packet with finish time greater than $f^E_{k,i}$ is transmitted in $[a_{u,j}, t]$. From (T2) in the definition of $\hat{T}_k$, no packet with finish time greater than $f^E_{k,i}$ is transmitted in $[\hat{T}_k, f^E_{k,i}]$. Since $t \in [\hat{T}_k, f^E_{k,i}]$, from the definition of $\hat{T}_k$ we conclude that $\hat{T}_k \leq a_{u,j}$, which is a contradiction.

Now, we show that $w^E[\hat{T}_k, f^E_{k,i}] > (f^E_{k,i} - \hat{T}_k - V)$. Under E, if in a busy period the number of i's compensation slots exceeds V_i, then session i's packet is removed from the queue if transmitted, i.e., the packet is either successful transmitted or dropped. Recall that $V = \sum_{i \in \mathcal{C}} V_i$. Thus, $\widehat{S}^E[\hat{T}_k, f^E_{k,i}] \geq f^E_{k,i} - \hat{T}_k - V$.

Now, $p_{k,i}$ contributes to $w^E[\hat{T}_k, f^E_{k,i}]$ but departs after $f^E_{k,i}$.

$$\text{Hence,} \quad \begin{aligned} w^E[\hat{T}_k, f^E_{k,i}] &> \widehat{S}[\hat{T}_k, f^E_{k,i}] \\ &\geq (f^E_{k,i} - \hat{T}_k - V). \end{aligned} \tag{7}$$

Now using (5) and Lemma 10, (7) can be modified to

$$W^{PC}[\hat{T}_k, f^{PC}_{k,i}] \geq w^{PC}[\hat{T}_k, f^{PC}_{k,i}] > (f^{PC}_{k,i} - \hat{T}_k).$$

Then from (6), all the packets do not depart before their respective finish times in PC. This is a contradiction. ∎

G. Proof of Lemma 6

Proof: Since delay $D_j - V$ is guaranteeable under PC for every $j \in \mathcal{C}$, from Definition 1 we conclude that for every packet $p_{k,j}$, $f_{k,j}^{PC} \leq a_{k,j} + D_j - V$ and the packet $p_{k,j}$ is transmitted before $f_{k,j}^{PC}$. From Lemma 5, all the packets that are successfully transmitted under E depart before $f_{k,j}^{E} = f_{k,j}^{PC} + V$. Hence for every $j \in \mathcal{C}$, $\widehat{n}_j^E$ is equal to the number of packets dropped for session j.

Consider a busy period $\mathcal{B}$ under E and a packet P that arrives from session i in $\mathcal{B}$. Now, let $\mathcal{A}$ denote the event that a packet from session i is dropped in $\mathcal{B}$. Now, packets for session i are dropped in $\mathcal{B}$ only if the number of i's channel errors in $\mathcal{B}$ under E exceed V_i. Thus, $\mathbf{P}\{\mathcal{A}\} \leq \mu_i$.

$$\begin{aligned}\mathbf{P}\{\text{Packet } P \text{ is dropped}\} &\leq \mathbf{P}(\mathcal{A}) \\ &\leq \mu_i. \end{aligned} \tag{8}$$

Let $\mathbf{1}_{k,i}$ denote an indicator random variable that is equal to 1 if $p_{k,i}$ is dropped and it is 0 otherwise. Since P and $\mathcal{B}$ are arbitrary, from (8) we conclude that for every $k \in \{1, \ldots, n_i\}$

$$\mathbf{E}[\mathbf{1}_{k,i}] \leq \mu_i. \tag{9}$$

Now, the fraction of packets dropped is $\frac{\widehat{n}_i^E}{n_i} = \frac{\sum_{k=1}^{n_i} \mathbf{1}_{k,i}}{n_i}$. From (9), $\frac{\mathbf{E}[\widehat{n}_i^E]}{n_i} \leq \mu_i$. ∎

H. Proof of Lemma 7

Let T be a constant. We obtain a function $\phi_i(T)$ such that the probability of the number of channel errors of i exceeds $T\phi_i(T)$ in duration of T slots is at most μ_i (Lemmas 11, 12 and 13). We prove Lemma 7 using properties of the function $\phi_i(T)$. First, an observation follows.

Let random variable X denote the number of slots between successive visits to state *good* for session i. The distribution of X is as follows.

$$\begin{aligned}\mathbf{P}\{X = 1\} &= p_i \\ \mathbf{P}\{X = u\} &= (1 - p_i)(1 - q_i)q_i^{(u-2)} \text{ for all } u \geq 2. \end{aligned} \tag{10}$$

$$\text{Thus,} \qquad \mathbf{E}[e^{\theta X}] = p_i e^{\theta} + \sum_{u=2}^{\infty} e^{\theta u}(1-p_i)(1-q_i)q_i^{u-2}$$
$$= \frac{p_i e^{\theta} - (p_i + q_i - 1)e^{2\theta}}{1 - q_i e^{\theta}} \qquad \text{for } \theta \in \left[0, \log \tfrac{1}{q_i}\right). \tag{11}$$

Lemma 11: Let channel for session i have *good* state in slot 0. Let X_k be a random variable which represents the number of slots between $k-1^{th}$ and k^{th} visits to state *good*. Then,

$$\mathbf{P}\left\{\sum_{k=1}^{n} X_k \geq na\right\} \leq \left[\frac{1+q_i-p_i}{q_i}\left(\frac{2q_i}{1+q_i}\right)^{a-1}\right]^n. \tag{12}$$

Proof: Since the channel error process is Markovian, random variables X_k's are iid, and the X_k's have the same distribution as that of random variable X that is given in (10).

$$\text{Now,} \qquad \mathbf{P}\left\{\sum_{k=1}^{n} X_k \geq na\right\} \leq \inf_{\theta \geq 0}\left[e^{-n\theta a}\mathbf{E}\left(e^{\theta \sum_{k=1}^{n} X_k}\right)\right]. \tag{13}$$

$$\text{Since } X_k\text{'s are iid,} \qquad \mathbf{P}\left\{\sum_{k=1}^{n} X_k \geq na\right\} \leq \inf_{\theta \geq 0}\left[e^{-\theta a}\mathbf{E}\left(e^{\theta X}\right)\right]^n. \tag{14}$$

$$\mathbf{P}\left\{\sum_{k=1}^{n} X_k \geq na\right\} \leq \left[e^{-\theta a}\frac{p_i e^{\theta} - (p_i+q_i-1)e^{2\theta}}{1-q_i e^{\theta}}\right]^n \qquad \forall \quad \theta \in \left[0, \log \frac{1}{q_i}\right). \qquad \text{(from (11))}$$
$$= \left[e^{-\theta(a-1)}\frac{p_i - (p_i+q_i-1)e^{\theta}}{1-q_i e^{\theta}}\right]^n.$$

We choose $\widehat{\theta} = \log\left(\frac{1+q_i}{2q_i}\right)$. Then, $\widehat{\theta} \in \left[0, \log \frac{1}{q_i}\right)$. Thus,

$$\mathbf{P}\left\{\sum_{k=1}^{n} X_k \geq na\right\} \leq \left[\frac{1+q_i-p_i}{q_i}\left(\frac{2q_i}{1+q_i}\right)^{a-1}\right]^n. \tag{15}$$

■

Lemma 12: Let the MC indicating the channel state of session i be in steady state and let Y_k be a r.v. indicating the number of slots between the $k-1^{th}$ and the k^{th} visits to state *good*. Also, let $\pi_{G,i}$ and $\pi_{B,i}$ be the steady state probabilities that the MC is in *good* and *bad* states respectively. Then,

$$\mathbf{P}\left\{\sum_{k=1}^{n} Y_k \geq na\right\} \leq \left[\frac{2q_i\pi_{B,i}}{1+q_i-p_i} + \pi_{G,i}\right]\left[\frac{1+q_i-p_i}{q_i}\left(\frac{2q_i}{1+q_i}\right)^{a-1}\right]^n. \tag{16}$$

Proof: In Lemma 11 we have considered a case in which the MC starts from the *good* state. In this lemma, the MC can start from either *good* or *bad* state with steady state probability $\pi_{G,i}$ and $\pi_{B,i}$ respectively. Recall that the random variable X denotes the number of slots for the first visit to state *good* starting from state *good*. Similarly, let random variable $\widehat{X}$ denote the number of slots for the first visit to state *good* starting from state *bad*. Observe that Y_1 has the same distribution as X and $\widehat{X}$ w.p. $\pi_{G,i}$ and $\pi_{B,i}$ respectively.

The distribution for random variable X is given in (10). The distribution for $\widehat{X}$ is as follows.

$$\mathbf{P}\{\widehat{X}=u\}=(1-q_i)q_i^{u-1}, \text{ for all } u\geq 1.$$

$$\text{Thus,}\qquad \begin{aligned}\mathbf{E}[e^{\theta\widehat{X}}] &= \sum_{u=1}^{\infty} e^{\theta u}(1-q_i)q_i^{u-1}\\ &= \frac{e^{\theta}(1-q_i)}{1-q_i e^{\theta}}, \text{ for every } \theta\in\left[0,\log\tfrac{1}{q_i}\right).\end{aligned} \tag{17}$$

$$\text{Now,}\qquad \mathbf{P}\left\{\sum_{k=1}^{n} Y_k\geq na\right\}\leq \inf_{\theta\geq 0}\left[e^{-n\theta a}\mathbf{E}\left(e^{\theta\sum_{k=1}^{n}Y_k}\right)\right]. \tag{18}$$

The r.v.'s Y_k's are independent as the epochs of visits to the *good* state are the stopping time for the MC, and also Y_k's have the same distribution as X for every $k\geq 2$. Furthermore, Y_1 also has the same distribution as the r.v. X ($\widehat{X}$) if the MC is in *good* (*bad*) state as time 0. Hence Y_1 is equal to X ($\widehat{X}$) w.p. $\pi_{G,i}$ ($\pi_{B,i}$). Now,

$$\begin{aligned}\mathbf{E}\left(e^{\theta\sum_{k=1}^{n}Y_k}\right) &= \pi_{B,i}\mathbf{E}\left(e^{\theta\widehat{X}}e^{(n-1)\theta X}\right)+\pi_{G,i}\left[\mathbf{E}\left(e^{\theta X}\right)\right]^n\\ &= \left[\pi_{B,i}\frac{\mathbf{E}\left(e^{\theta\widehat{X}}\right)}{\mathbf{E}\left(e^{\theta X}\right)}+\pi_{G,i}\right]\left[\mathbf{E}\left(e^{\theta X}\right)\right]^n.\end{aligned} \tag{19}$$

Substituting (19) in (18) we obtain

$$\mathbf{P}\left\{\sum_{k=1}^{n} Y_k\geq na\right\}\leq \inf_{\theta\in[0,\log\frac{1}{q_i})}\left\{\left[\pi_{B,i}\frac{\mathbf{E}\left(e^{\theta\widehat{X}}\right)}{\mathbf{E}\left(e^{\theta X}\right)}+\pi_{G,i}\right]\left[e^{-\theta a}\mathbf{E}\left(e^{\theta X}\right)\right]^n\right\}.$$

Let $\widehat{\theta}=\log\frac{1+q_i}{2q_i}$. Then, $\widehat{\theta}\in\left[0,\log\frac{1}{q_i}\right)$. Thus,

$$\mathbf{P}\left\{\sum_{k=1}^{n} Y_k\geq na\right\}\leq \left[\pi_{B,i}\frac{\mathbf{E}\left(e^{\widehat{\theta}\widehat{X}}\right)}{\mathbf{E}\left(e^{\widehat{\theta}X}\right)}+\pi_{G,i}\right]\left[e^{-\widehat{\theta}a}\mathbf{E}\left(e^{\widehat{\theta}X}\right)\right]^n.$$

Using (11) and (17) with $\widehat{\theta} = \log \frac{1+q_i}{2q_i}$ we obtain (16) after some algebra. ■

Lemma 13: Consider any interval of T slots and a constant c. Let $\mathcal{A}$ denote the event that the number of visits to *bad* state for session i in duration T is greater than cT. Let $\alpha_i = \frac{2q_i\pi_{B,i}}{1+q_i-p_i}+\pi_{G,i}$, $\beta_i = \frac{1+q_i-p_i}{q_i}$ and $\zeta_i = \frac{2q_i}{1+q_i}$. Then $\mathbf{P}(\mathcal{A}) \leq \mu_i$ if $c \geq \phi_i(T)$, where

$$\phi_i(T) = \frac{\log(\beta_i)}{\log\left(\frac{\beta_i}{\zeta_i}\right)} + \frac{\log\left(\frac{\alpha_i}{\mu_i}\right)}{T\log\left(\frac{\beta_i}{\zeta_i}\right)}. \tag{20}$$

Proof: From Lemma 12,

$$\mathbf{P}\left\{\sum_{k=1}^{n} Y_k \geq na\right\} \leq \alpha_i\left(\beta_i\zeta_i^{a-1}\right)^n. \tag{21}$$

$$\text{Let} \qquad n = (1-c)T \ \text{ and} \tag{22}$$

$$a = \frac{1}{1-c}. \tag{23}$$

Now, the event $\sum_{k=1}^{(1-c)T} Y_k \geq T$ implies that the time required for $(1-c)T$ visits to *good* state is greater than or equal to T. This is equivalent to the event that the number of visits to *bad* state in duration T is greater than cT, which is the event $\mathcal{A}$.

$$\text{Thus,} \qquad \mathbf{P}(\mathcal{A}) \leq \alpha_i\left(\beta_i\zeta^{a-1}\right)^{\frac{T}{a}} \qquad \text{(using (21) and } n = \tfrac{T}{a}\text{)}. \tag{24}$$

$$\begin{aligned} \text{Now} \quad \frac{\log(\beta_i)}{\log\left(\frac{\beta_i}{\zeta_i}\right)} + \frac{\log\left(\frac{\alpha_i}{\mu_i}\right)}{T\log\left(\frac{\beta_i}{\zeta_i}\right)} &\leq c \\ \Rightarrow \qquad \alpha_i\left(\beta_i\zeta^{a-1}\right)^{\frac{T}{a}} &\leq \mu_i. \end{aligned}$$

Thus the result follows from (20) and (24). ■

Now, we prove Lemma 7.

Proof: Let $\phi_i(T)$ be as defined in Lemma 13. From (20), we conclude that

$$\phi_i(T) \leq \phi_i(\widehat{T}), \ \text{whenever } T \geq \widehat{T}. \tag{25}$$

Now, from (3), (2) and (4), we observe that $Z \geq \widehat{Z}$. Hence from (25) it follows that

$$\phi_i(Z) \leq \phi_i(\widehat{Z}). \tag{26}$$

From (26) and Lemma 13, it follows that the number of channel errors for session i in the duration Z is greater than or equal to $Z\phi_i(\widehat{Z})$ w.p. at most μ_i. Observe that $\delta_i = \phi_i(\widehat{Z})$.

Now we show that Z is an upper bound on the length of busy period under E when $V_i = \delta_i Z$ for every i. Suppose not. Then there exists a busy period under E that has length greater than Z. Without loss of generality let this busy period begin at time 0. Since the system is busy at time Z, the number of packets that arrived till Z is greater than or equal to number of packets that departed the system till Z. Thus,

$$\sum_{i\in\mathcal{C}} \gamma_i[0,Z] > \widehat{S}^E[0,Z]. \tag{27}$$

Now, if the number of compensation slots for session i in a busy period under E is greater than or equal to V_i, then packets of session i depart the system when transmitted. Hence,

$$\begin{aligned} \widehat{S}^E[0,Z] &\geq Z - \sum_{i\in\mathcal{C}} V_i \\ &= (1 - \sum_{i\in\mathcal{C}} \delta_i)Z. \end{aligned} \tag{28}$$

Furthermore, the number of packets that arrived in an interval is less than the value of envelop function for the arrival process. Thus,

$$\sum_{i\in\mathcal{C}} \gamma_i[0,Z] \leq \sum_{i\in\mathcal{C}} g_i(Z). \tag{29}$$

From (27), (28) and (29) we conclude that $\sum_{i\in\mathcal{C}} g_i(Z) > (1 - \sum_{i\in\mathcal{C}} \delta_i)Z$. This contradicts (2). Hence, Z is an upper bound on the busy period length under E.

Thus, any busy period has length $B \leq Z$. Hence, an event that the number of channel errors for session i in B slots is greater than V_i implies the event that the number of channel errors for session i in Z slots is greater than V_i. Hence, the result follows. ■

I. Proof of Lemma 8

Proof: We prove Lemma 8 using induction on t. Recall that by (A1), for every packet $p_{k,i}$

$$f^E_{k,i} = f^{EC}_{k,i} + V. \tag{30}$$

<u>*Baseline Case $t = 0$:*</u>

The systems E and EC have no packet queued in slot 0. Hence, if no packet arrives at $t = 0$, then (C1) to (C4) hold by vacuity. So, we only consider a case in which packets arrive in slot 0.

Proof of (C1) in slot 0: The arriving packets are queued in the primary queues under EC. Further by (30) the finish times under EC are equal to finish times under E plus a constant. Thus, (C1) holds.

Proof of (C2) in slot 0: Since a busy period under E and a primary busy period under EC start in slot 0, the value of compensation counter for every session j is initialized to 0 under both the systems. Thus, (C2) holds.

Proof of (C3) in slot 0: From (C1), $p_{k,i}$ has the smallest finish time in slot 0 *iff* $p_{k,i}$ has the smallest finish time among all the packets in the primary queues under EC. From Figure 6, $p_{k,i}$ will be transmitted under E. The transmission is successful if i has *good* channel in slot 0. From Figure 5, $p_{k,i}$ will be transmitted under EC as well since the compensation queues are empty. Again, the transmission will be successful if the channel for i is in a *good* state. Thus, (C3) holds.

Proof of (C4) in slot 0: Let $p_{k,i}$ be transmitted in slot 0 under E. Since, the compensation queues are empty under EC in slot 0, the same packet is transmitted under EC as well (from (C1) and Figure 5). Consider two conditions: (a) the channel for session i is *bad* in slot 0, and the value of V_i under E is less than or equal to i's compensation counter (b) the channel for session i is *good* in slot 0. The value of the compensation counter is the same for every session under E and EC (from (C2)). Now, $p_{k,i}$ will depart under E in slot 0 iff either (a) or (b) holds. In case (a), $p_{k,i}$ will be dropped (Figure 6), and in case (b) $p_{k,i}$ will be transmitted successfully. Under EC, $p_{k,i}$ will depart from i's primary queue *iff* either (a) or (b) holds. Here, if (a) happens then $p_{k,i}$ is transfered to i's compensation queue (Figure 5), and if (b) happens then $p_{k,i}$ is transmitted successfully. Thus, (C4) holds.

<u>*Induction Hypothesis (IH):*</u> Statements (C1) to (C4) hold in every $t \in \{0, \dots, T\}$.

Recall that packet arrivals under E and EC are the same, and the arriving packets are stored in the primary queues in EC. Let $\widehat{t} \leq T$. The packets present at $\widehat{t}$ in E are the ones that arrived in the interval $[0, \widehat{t}]$ and have not departed in the interval $[0, \widehat{t}]$. Similarly, packets present in the primary queues of EC at $\widehat{t}$ are the ones that arrived in the interval $[0, \widehat{t}]$ and have not departed from the primary queues in interval $[0, \widehat{t}]$. By *IH*, (C4) holds for all slots till $\widehat{t}$. Thus, the packets departed under E are the ones that departed the primary queues under EC. Thus, the packets in E and the packets in the primary queues of EC are the same in any slot $\widehat{t} \leq T$.

Proof of (C1) in slot $T+1$*:* The packets present in E and those in the primary queues of EC are the same in slot T. Also, the same packets arrive in slot $T+1$ under both the systems. The arriving packets are stored in the primary queue in EC. Thus, the packets in E and those in the primary queues of EC are the same in the beginning of slot $T+1$. The finish times under E and EC differ by a constant (refer (30)). Thus, (C1) holds.

Proof of (C2) in slot $T+1$*:* First we assume that the queues for all the sessions under E are empty in slot T. Then from *IH*, the primary queues for all the sessions under EC are also empty in T. Note that under E the values of the compensation counters do not change when all the queues are empty (Figure 6). Also, under EC the values of the compensation counters do not change when all the primary queues are empty (Figure 5). Thus if no packet arrives in $T+1$, the values of the compensation counters in $T+1$ are the same as that in T under both E and EC for every session i in $T+1$. By *IH*, (C2) holds in T. Hence, (C2) holds in $T+1$ as well. If packets arrive in $T+1$, then a busy period under E and a primary busy period under EC start in $T+1$. Hence, the value of the compensation counter for every session i is initialized to 0 under both the systems. Thus, (C2) holds in $T+1$.

Now, we consider a case in which at least one queue have a packet in slot T under E. Thus, under E source transmits a packet say $p_{k,i}$ in T. Thus $p_{k,i}$ has the smallest finish time in E. From *IH* and (C1) in slot T, $p_{k,i}$ has the smallest finish time among all the packets in the primary queues in EC. Thus, some packet of session i is transmitted in slot T under EC. Let session i's channel be *good* in slot T. Recall that the channel state for every session is the same under both the systems. Hence, the transmissions under E and EC are successful and the values of the compensation counters do not change under both the systems. By *IH*, (C2) holds in slot T. Thus, (C2) holds in $T+1$. Now, let session i's channel be *bad* in slot T. Then, the transmissions under E and EC are not successful and subsequently i's compensation counters are incremented by one in both the systems. By *IH*, (C2) holds in slot T. Thus, (C2) holds in $T+1$.

Proof of (C3) in slot $T+1$*:* Let $p_{k,i}$ be transmitted successfully under E in slot $T+1$. From (C1) in slot $T+1$, $p_{k,i}$ has the smallest finish time under E *iff* $p_{k,i}$ has the smallest finish time among the packets in the primary queues of EC in slot $T+1$. The channel state for every session is the same under E and EC. Thus, channel state for session i is good in slot $T+1$. We consider two cases.

Case 1: The compensation queue for session i has a packet whose deadline is not violated. Let $p_{k_1,i}$ be the first among such packets.

Case 2: No packet in i's compensation queue has valid deadline.

In Case 1, from Figure 5 we note that $p_{k,i}$ will be transfered to i's compensation queue, and $p_{k_1,i}$ will be transmitted. In Case 2, $p_{k,i}$ will be transmitted. Since i's channel is in *good* state in slot $T+1$, the transmission of either $p_{k_1,i}$ or $p_{k,i}$ is successful. Since $p_{k_1,i}$ is in i's compensation queue when $p_{k,i}$ is in the primary queue, $p_{k_1,i}$ has arrived before $p_{k,i}$. Hence, $k_1 < k$. Thus, (C3) holds in $T+1$.

Proof of (C4) in slot $T+1$*:* Here we prove *only if* part of (C4). Proof for *if* part follows from similar arguments.

Let $p_{k,i}$ depart from E in slot $T+1$. Thus, $p_{k,i}$ is transmitted in slot $T+1$. From (C2) in slot $T+1$, the compensation counters have the same value for every session under E and EC. Also, from (C3) in $T+1$, i's packet is transmitted in slot $T+1$. Thus, if i's compensation queue is empty in slot $T+1$, then the proof is the same as the proof for (C4) in slot 0. Hence, let i's compensation queue has a packet. Now, $p_{k,i}$ departs from E *iff* condition (a) or (b) hold (refer to proof for (C4) in slot 0). Now, we argue that $p_{k,i}$ departs from i's primary queue *iff* (a) or (b) holds. From Figure 5, when (a) holds, $p_{k,i}$ is transfered to i's compensation queue. When (b) holds, then also $p_{k,i}$ is transfered to i's compensation queue and then the first packet with valid deadline is transmitted. Further, these are the only cases under which $p_{k,i}$ departs from i's primary queue. Thus, (C4) holds in slot $T+1$.

Hence by induction, (C1) to (C4) hold in every slot. ■

Procedure Scheduling_Interface()
begin

When a new slot t begins
Assign finish times to packets arrived in the slot;
$\mathcal{C}_{primary-busy} = \{j : prim\_queue(j)$ is not empty$\}$;
$\mathcal{C}'_{comp-busy} = \{j : comp\_queue(j)$ is not empty$\}$;
$ft_j =$ the finish time of HoL packet in $prim\_queue(j)\ \forall j \in \mathcal{C}_{primary-busy}$;
Update $ft'_j =$ the finish time of HoL packet in $comp\_queue(j)\ \forall j \in \mathcal{C}'_{comp-busy}$;
If a new primary busy period begins at t, $Comp\_counter(j) = 0$ for every session j;
if ($\mathcal{C}_{primary-busy} \neq \phi$) **then**
    /* If some session has a packet in primary queue */
    $i = \arg\min_{j \in \mathcal{C}_{primary-busy}} \{ft_j\}$;
    if ($Transmit\_comp\_packet(i)$ = FALSE) **then**
        /* session i does not have packet with valid deadline in compensation queue */
        Transmit HoL packet of $prim\_queue(i)$;
    if (the packet is received without error) **then**
        /* i had good channel */
        delete the transmitted packet from the corresponding queue;
        if (Packet is transmitted from $comp\_queue(i)$) **then**
            Transfer HoL packet of $prim\_queue(i)$ to $comp\_queue(i)$;
    else
        /* i had a bad channel */
        $Comp\_counter(i) = Comp\_counter(i) + 1$;
        if ($Comp\_counter(i) > V_i$) **then**
            /* i has already taken allotted compensation */
            Put HoL packet of $prim\_queue(i)$ into $comp\_queue(i)$;
else
    $flag$ = FALSE;
    while (($\mathcal{C}'_{comp-busy} \neq \phi$) AND ($flag$ = FALSE)) **do**
        $i = \arg\min_{j \in \mathcal{C}'_{comp-busy}} \{ft'(j)\}$;
        $flag = Transmit\_comp\_packet(i)$;

end

Procedure Transmit_comp_packet(i)
begin

$transmit\_flag$ = FALSE;
while ($comp\_queue(i)$ is not empty) **do**
    if (finish time of HoL packet in $comp\_queue(i) \leq curr\_time$) **then**
        Drop the packet;
    else
        Transmit HoL packet of $comp\_queue(i)$;
        $transmit\_flag$ = TRUE;
        exit while loop;
return($transmit\_flag$)

end

Fig. 5. Pseudo code of a general framework with compensation queues given no knowledge of instantaneous channel state

Procedure Scheduling_for_E()
begin

When a new slot t begins

Assign finish times to packets arrived in the slot;

$\mathcal{C}_{busy} = \{j :$ session j has a packet$\}$;

$ft_j =$ finish time of HoL packet of session $j \ \forall j \in \mathcal{C}_{busy}$;

If a new busy period begins at t, $Comp\_counter(j) = 0$ for every session j;

if ($\mathcal{C}_{busy} \neq \phi$) **then**

/* If some session has a packet */

$i = \arg\min_{j \in \mathcal{C}_{busy}} \left\{ ft_j \right\}$;

Transmit HoL packet of session i;

if (the packet received without error) **then**

/* i had good channel */

delete the transmitted packet from the system;

else

/* i had a bad channel */

$Comp\_counter(i) = Comp\_counter(i) + 1$;

if ($Comp\_counter(i) \geq V_i$) **then**

/* i has already taken allotted compensation */

drop HoL packet of session i;

end

Fig. 6. Pseudo code of a scheduling under E