Wireless Networks 9, 21–35, 2003

Havana: Supporting Application and Channel Dependent QoS in Wireless Packet Networks

JAVIER GOMEZ * and ANDREW T. CAMPBELL
COMET Group, Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Abstract. For wireless channels, interference mitigation techniques are typically applied at the packet transmission level. In this paper, we present the Havana framework which supports *integrated adaptive-QoS* in wireless packet networks by responding to impairments over multiple time scales that are present at the flow/session level. The Havana framework is based on three different control mechanisms that operate over distinct adaptation time scales. At the packet transmission time scale, a packet-based channel predictor determines whether to transmit a packet or not depending on the state of the wireless channel. At the packet scheduling time scale, a compensator credits and compensates flows that experience bad link quality. Over even longer time scales an adaptor regulates flows taking into account the ability of wireless applications to adapt to changes in the available bandwidth and channel conditions. We present the design and implementation of our framework and evaluate each of the proposed control mechanisms using the *ns*-2 simulator.

Keywords: adaptive wireless networks, QoS

1. Introduction

There has been considerable discussion in the mobile networking research community about the most suitable service model for the delivery of mobile multimedia over wireless packet networks. One school of thought believes that the radio can be simply engineered to provide hard-QoS assurances (e.g., guaranteed delay or constant bit rate services) found in wireline networks. While others argue that the wireless link cannot be viewed in this manner because of the inherent time varying environmental conditions evident in radio communications (e.g., fading). In this case, wireless services lend themselves to more adaptive QoS approaches [8] or better than best-effort service paradigms [12].

In this paper, we take our lead from the adaptive camp and propose the Havana framework for application and channel dependent QoS control. Our approach incorporates adaptation techniques for packet scheduling and application-level rate control taking into account wireless channel conditions and the ability of application level flows/sessions to adapt to these conditions over multiple time scales. We argue that an adaptive-QoS service paradigm is suitable for the delivery of voice, video and data to mobile devices.

We introduce an *integrated adaptive-QoS* model founded on the notion of exchanging state information between control mechanisms capable of responding to different time-varying wireless characteristics. These mechanisms operate over three distinct time scales and include a *predictor*, *compensator* and *adaptor*. An *arbitrator* monitors the state of each of these components coordinating their operation in an integrated and systematic manner. Channel prediction allows the arbitrator to defer transmission to mobile devices that are experiencing time varying channel conditions (e.g., fading). Channel prediction, however, cannot compensate mobile devices that have previously experienced "outages" due to poor channel conditions. To address this issue the arbitrator interworks with a compensator to deliver enhanced throughput to mobile devices. The compensator attempts to resolve any unfairness issues experienced by different spatially distributed receivers and operates on the packet scheduling time scale. When persistent fading conditions exceed the operational range of the compensator, the arbitrator activates the adaptor module to take further adaptive action. The adaptor is designed to operate over even longer time scales than the compensator taking into account application specific semantics (e.g., packet priorities within a flow/session) in the case of severe channel degradation or variations in available bandwidth. Ideally the integrated adaptive-QoS model should be used in conjunction with adaptive modulation/coding techniques and other interference mitigation techniques (e.g., smart antennas, multi-user detection, power control) in order to achieve optimum performance from the application level to the physical communication link.

In this paper, we present the design, implementation and evaluation of the Havana framework which includes the predictor, compensator and adaptor modules operating over a simulated wireless IP network supporting IEEE 802.11 last hop wireless LANs. Results from this work are also applicable to mobile ad hoc networks [7]. The paper is organized as follows. Section 2 discusses previous work in the area of channel prediction, compensation and adaptation. In section 3, we present an overview of the Havana framework. We describe the packet-based channel predictor and compensator schemes in sections 4 and 5, respectively. In section 6, we discuss the adaptor mechanism that supports application-specific adaptation policy. Following this, we present an evaluation of the system and its components in an incremental fashion in section 7. First, we analyze the performance of the predictor

* Corresponding author.

module in isolation. Next, we add the compensator module to the predictor and show the benefits of compensating flows under a variety of wireless channel conditions. Finally, we add the adaptor module to the predictor and compensator and show how the fully integrated QoS-adaptive system works in unison to deliver application and channel dependent QoS over wireless packet networks. We conclude in section 8 with some final remarks.

2. Related work

Previous work in the area of channel prediction, compensation and adaptation have mainly focused on the performance of individual mechanisms and their operation in isolation. In contrast, we argue that an integrated view of the problem must consider prediction, compensation and adaptation mechanisms working in unison in a "cut-through" manner. Such an approach, we argue, may lead to more comprehensive solutions to the delivery of media to mobile hosts over wireless packet networks.

Much of the literature that discusses packet-based compensation mechanisms capable of responding to fading conditions assume either perfect prediction of the channel state or some apriori knowledge of the channel behavior. In [1], for example, fading periods are considered to last between 50 to 100 ms. Given this assumption, the scheduler defers transmission to a mobile device for a period of 50–100 ms when a fade occurs. In [4], the base station assumes it has instantaneous knowledge of channel conditions. In [10] link layer acknowledgments are used to determine if a packet is received correctly or not. A packet exchange protocol that uses Request-To-Send (RTS) and Clear-To-Send (CTS) as a channel predictor is proposed in [6]; however, no evaluation of the scheme is discussed. In this paper, we evaluate the use of RTS-CTS as a channel predictor and show the limits of such an approach.

A mechanism for compensation of flows in wireless networks is presented in [10]. Flows unable to transmit packets due to channel fading conditions are credited for future transmissions when the link returns to a good state. This strategy has the drawback that a flow coming out of a fading condition will be instantaneously compensated in one operation. This raises a number of performance issues. Even if the maximum compensation is bounded, it will introduce delays for other flows having good link qualities. This problem is resolved in [4] by limiting the amount of bandwidth that "leading flows" (i.e., flows receiving more bandwidth than requested) provide to "lagging flows" (i.e., flows receiving less bandwidth than requested due to past fading conditions) as part of the compensation strategy. In this paper, the compensator limits the amount of one-time compensation. We do not, however, couple the amount of compensation given to lagging flows with "leading" bandwidth. Rather, we base our compensation strategy on the availability of unused bandwidth in the system and limit compensation given during periods of high network load. Our scheme, therefore, does not maintain state associate with "leading flows" to compute compensation given to "lagging flows". This results in a greatly simplified compensator design. The compensator mechanism discussed in this paper is based on Deficit Round Robin (DRR) [15], an implementation of Fair Queuing which provides throughput fairness among flows.

The adaptor mechanism proposed in this paper is capable of operating on an end-to-end and wireless hop basis. The end-to-end component is application specific and regulates traffic at bottleneck wireless access points over the end-to-end time scale. The other component of the adaptor is similar to mechanisms such as Random Early Detection (RED) [5] in the sense that the adaptor drops packets when there is congestion due to packet loss triggered by bad channel conditions. The main goal of RED is to maximize the utilization of an output link shared by several flows. This differs from the design of our adaptor, which attempts to maintain buffer occupancy at a level that assures high priority packets (e.g., "In" traffic) can be forwarded with high probability even when unexpected bad channel conditions or bursty data are observed. The adaptor attempts to meet this goal without dropping packets prematurely in order to achieve good utilization across wireless links. In this respect, the adaptor results in very similar behavior to RED with "In" and "Out" priorities (RIO) [3].

We argue that the prediction, compensation and adaptation mechanisms need to operate in an integrated and systematic manner to meet the challenge of delivering real-time services over wireless packet networks.

3. The Havana framework

Network dynamics found in wireless networks are the result of several different system interactions operating over multiple time scales. These time scales range from received signal strength variations in the order of nanoseconds to deep fades or variations in available bandwidth occurring anywhere between hundreds of milliseconds to minutes. It is well known that several mechanisms such as modulation, forward error correction, automatic repeat request (ARQ) and interleaving are very useful in dealing with fast radio channel impairments at the packet transmission level time scale. It is unclear, however, which mechanisms are the most appropriate when channel impairments become severe and go far beyond the operational range of these mechanisms. The integrated adaptive-QoS model attempts to take this time-varying behavior into account by operating over three distinct time scales in response to wireless network dynamics.

Figure 1 illustrates the Havana framework that operates at wireless access points and mobile devices. The wireless network model assumes that gateway routers interconnect a set of cellular access networks to the Internet, (e.g., as in the case of Cellular IP access networks [2]). A cellular access network comprises one or more forwarding nodes that can be configured to provide access points to mobile hosts. The main controller of the Havana framework is a central arbitrator present at each wireless access point and wireless device,

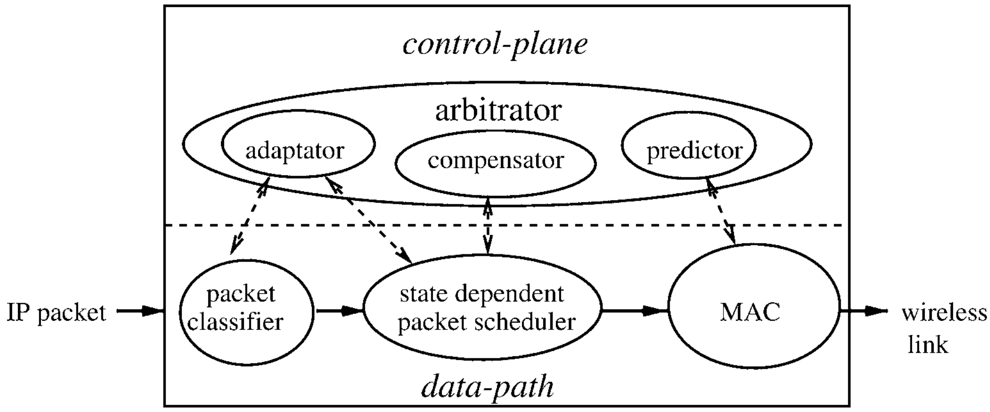

Figure 1. Havana framework.

as illustrated in figure 1. The access point model comprises a data path, which includes a packet classifier, state dependent packet scheduler and MAC, as illustrated in figure 1. In addition to the data path, a control plane comprises a number of QoS control mechanisms that support the data path's buffering, packet processing and forwarding mechanisms. An arbitrator coordinates and passes state information between the predictor, compensator and adaptor. Before a packet can be transmitted, the arbitrator requests the predictor to test the state of the wireless link. Depending on the state of the channel, the arbitrator will either initiate the transmission of a packet or arrange to buffer the packet and trigger the compensator to "credit" the flow-state. When a flow's buffer is about to overflow, the arbitrator invokes the adaptor to configure suitable filters in the adaptor to drop low priority packets. In what follows, we describe the operation of predictor, compensator and adaptor.

4. The predictor

Channel prediction allows a transmitter to probe the state of the wireless channel before transmitting a packet. If the predictor detects that the channel is in a "bad" state then the packet remains queued in the scheduler for later transmission and the flow-state is "credited" accordingly. If the channel is detected to be in a "good" state then a packet is transmitted [6]. Channel prediction also allows for better utilization of power resources in mobile devices because packets are transmitted only when channel conditions are good avoiding costly retransmissions of corrupted packets. Previous work on channel prediction either assumes that the state of the channel or the duration of bad link periods are known in advance [4,6,10], as discussed in section 2. In practice, however, the state of wireless links cannot be entirely predicted in advance.

To estimate the state of the channel, we have implemented a simple handshake probing protocol based on the RTS/CTS mechanism. Our channel predictor operates as follows. Before the start of each packet transmission to a mobile device a short probing RTS packet is sent to the designated receiver. The mobile device responds by sending a CTS packet as an acknowledgment to the RTS. If the CTS packet is received intact the state of the channel is assumed to be good. If, on the other hand, the CTS is not received after a given timeout then the channel state is considered to be bad. The assumption is that the RTS or CTS could have been corrupted, lost or incorrectly received because of degraded channel conditions which manifest as increased bit errors and loss of signal at the receiver.

In IEEE 802.11, RTS-CTS is used in the Distributed Control Function mode (DCF) to compensate for the hidden terminal problem, which can lead to a very large number of collisions for heavily loaded channels. However, even if RTS-CTS fails because of channel errors, the transmitting mobile device always assumes the problem is a result of hidden terminals and will back off before trying again. During the Point Coordination Function (PCF) operation, the access point is able to acquire the channel before any neighboring mobile devices in the coverage area. Therefore, there is no need to use RTS-CTS to prevent collisions in this instance. Rather, any packet received in error in the PCF mode is unambiguously the result of channel conditions. In our framework, the predictor operates in the PCF mode to verify the state of the channel. In the IEEE 802.11 PCF mode the access point always initiate transmission for both the downlink (transmitting the packet) or uplink (polling a mobile device), allowing RTS-CTS to be used for both downlink/uplink transmissions.

4.1. Analysis

In what follows, we use an analytical framework to investigate the bounds and utility of the prediction approach described above. A Markov model is used to model the good and bad states of a wireless channel [16]. We assume that the transmission of packets during good periods assures error free delivery. On the other hand, during a bad period we assume that a transmitted packet will be received in error. This assumption simplifies the analysis and is realistic for IEEE 801.11 where no Forward Error Correction (FEC) protection is applied to transmitted packets and only CRC is used [13]. The transition between states occurs at discrete time instances according to the transition rates. Rather than using a single set of transition rates for a particular channel model, we analyzed

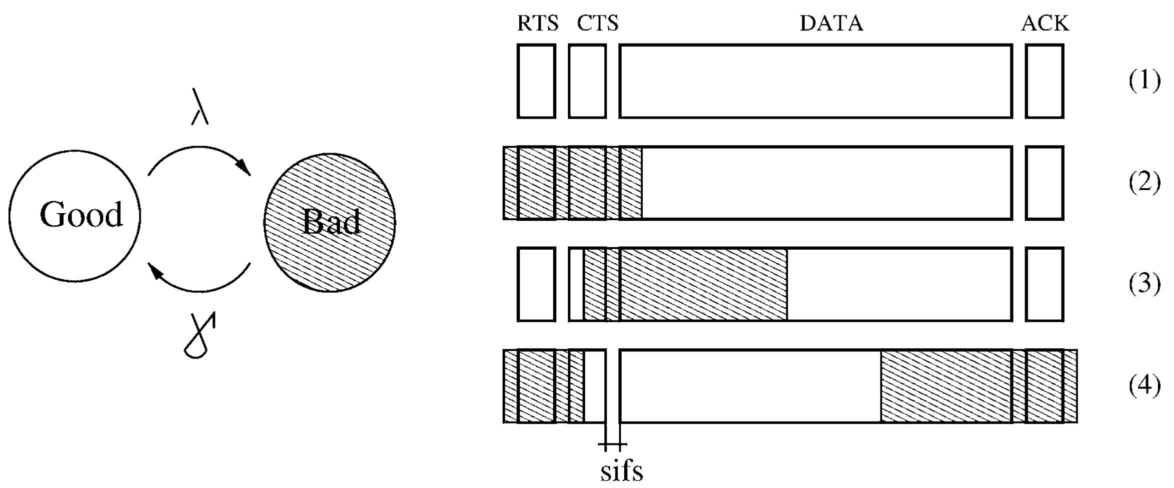

Figure 2. Channel model and predictor scenarios.

Table 1
Event combinations for a transmitted DATA packet. Legend: 0, error-free; 1, error; *, timeout.

RTS	0	0	1	0	0	0	0	1	1
CTS	0	1	*	0	0	1	1	*	*
Prediction	0	1	1	0	0	1	1	1	1
DATA	0	0	0	1	0	1	0	1	0
ACK	0	0	0	*	1	*	1	*	1
Transmission	0	0	0	1	1	1	1	1	1

the performance of the channel predictor for a wide range of rates.

Table 1 shows all the possible outcomes of RTS, CTS, DATA and ACK event combinations for a single packet transmission. Note that uplink analysis is similar but uses Request to Receive (CTR) and Clear to Send (CTS). Any packet transmitted can be received error-free (0) or in error (1). If both RTS and CTS packets are received correctly, the state of the channel is predicted as error-free, otherwise the channel is predicted in error. Depending on the reception of the DATA and the ACK packets the transmission is evaluated in the same way as the predictor.

Figure 2 shows a typical two-state Markov model. More formally, let $1/\lambda$ and $1/\gamma$ be the average time the channel is in good and bad states, respectively. The transition matrix of the Markov process is as follows [16]:

$$P = \begin{pmatrix} P(0|0) & P(1|0) \\ P(0|1) & P(1|1) \end{pmatrix} = \begin{pmatrix} 1-\lambda & \lambda \\ \gamma & 1-\gamma \end{pmatrix}, \quad (1)$$

with the steady state probability of the channel being in bad/good state given by

$$\pi_1 = \frac{\lambda}{\lambda+\gamma}, \qquad \pi_0 = 1 - \pi_1. \quad (2)$$

Initially at time T_0 the process is in one of the states X_{T_0} and at some time T_1, the process jumps to the other state X_{T_1}, and so on. If the process has just moved to state $X_{T_n} = i$ at time T_n, the time interval from T_n until the instance the process moves to the other state, denoted by w_n, is an exponentially distributed random variable with parameter a, then

$$P = \{T_{n+1} - T_n > t | X_{T_n} = i\} = P\{w_n > t\} = \mathrm{e}^{-at}. \quad (3)$$

Now let T_{pred}, T_{tran} and $T_{\text{pred+tran}}$ be the time it takes to transmit the predictor packets (RTS and CTS), the data packets (DATA and ACK) and the whole sequence (RTS, CTS, DATA and ACK), respectively. Before the transmission of CTS, DATA and ACK packets in IEEE 802.11 the transmitter waits for a Short-Inter-Frame-Space (SIFS) [13].

The probability that the channel predictor is correct, P_{pred}, is equal to the probability that RTS, CTS, DATA and ACK packets are received error-free plus the probability that predictor (RTS and CTS) and transmitted packets (DATA and ACK) are received in error, as shown in table 1. Figure 2 illustrates the four cases (1)–(4) in which prediction and transmission outcomes are equal. Because the four cases are disjoint, the probability that channel prediction is correct equals the sum of the probabilities for each case. If P^i_{pred} represents case i in figure 2, then $P_{\text{pred}} = \sum_{i=1}^{4} P^i_{\text{pred}}$. We simplify the model by taking into consideration that at most, only one channel state transition can occur during the transmission of RTS and CTS packets. Channel state transitions that are much smaller than RTS and CTS packets are considered to be outside of the operational range of channel prediction, where other techniques such as forward error correction and interleaving are more appropriate.

Figure 2 scenario (1) illustrates the case where the channel is in a good state at the beginning of an RTS and remains in good state until the corresponding ACK is received. We neglected the case when the channel changes from good to bad and from bad to good during a SIFS interval, then using equation (3),

$$P^{(1)}_{\text{pred}} = P\{T_1 - T_0 > T_{\text{pred+tran}};\ X_{T_0} = \text{good}\}, \quad (4)$$

$$\begin{aligned} P^{(1)}_{\text{pred}} &= P\{T_1 - T_0 > T_{\text{pred+tran}} | X_{T_0}\} P\{X_{T_0} = \text{good}\} \\ &= \pi_0 \mathrm{e}^{-\lambda t_{\text{pred+tran}}}. \end{aligned} \quad (5)$$

Predictor packets (RTS-CTS or RTR-CTR) and data packets (DATA-ACK) can be received in error in different ways, as illustrated in figure 2 scenarios (2), (3) and (4). The scenario (2) shown in figure 2 illustrates the case where the channel is in bad state at the beginning of the RTS and remains in bad state at least until the beginning of the DATA packet. Using a similar derivation to equation (5) but now with the channel in a bad state at time zero gives

$$P_{\text{pred}}^{(2)} = P\{T_1 - T_0 > T_{\text{pred}};\, X_{T_0} = \text{bad}\}, \tag{6}$$

$$\begin{aligned} P_{\text{pred}}^{(2)} &= P\{T_1 - T_0 > T_{\text{pred}} | X_{T_0} = \text{bad}\} P\{X_{T_0} = \text{bad}\} \\ &= \pi_1 \mathrm{e}^{-\gamma T_{\text{pred}}}. \end{aligned} \tag{7}$$

Scenario (3) shown in figure 2, illustrates the case where the channel is in a good state at the beginning of the RTS (T_0), at time T_x the channel changes to a bad state before the CTS is completely transmitted and it remains in a bad state until the beginning of the transmission of the DATA packet. If w_0 and w_1 are the intervals the channel is in good and bad states, respectively, then

$$P_{\text{pred}}^{(3)} = \int_{T_0}^{T_{\text{pred}}} P\{w_1 > T_{\text{pred}} - T_x;\, w_0 = T_x - T_0\}\, \mathrm{d}t_x \tag{8}$$

$$\begin{aligned} P_{\text{pred}}^{(3)} = \int_{T_0}^{T_{\text{pred}}} & P\{w_1 > T_{\text{pred}} - T_x | w_0 = T_x - T_0\} \\ & \times P\{w_0 = T_x - T_0\}\, \mathrm{d}T_x \end{aligned} \tag{9}$$

$$P_{\text{pred}}^{(3)} = \int_{T_0}^{T_{\text{pred}}} \big(1 - \mathrm{e}^{-\lambda T_x}\big)\big(\mathrm{e}^{-\gamma (T_{\text{pred}} - T_x)}\big)\, \mathrm{d}T_x. \tag{10}$$

The final scenario (4), shown in figure 2, illustrates the case where the channel is in a bad state at the beginning of transmission of the RTS and at time t_x changes to a good state before the CTS is completely transmitted. Furthermore, the channel returns back to a bad state before the corresponding ACK is received. Following a similar formulation to scenario (3) shown in figure 2, and assuming the channel is in a bad state at time T_0:

$$\begin{aligned} P_{\text{pred}}^{(4)} = \int_{T_0}^{T_{\text{pred}}} & P\{w_1 < T_{\text{pred+tran}} - T_x; \\ & w_0 = T_x - T_0\}\, \mathrm{d}t_x \end{aligned} \tag{11}$$

$$\begin{aligned} P_{\text{pred}}^{(4)} = \int_{T_0}^{T_{\text{pred}}} & P\{w_1 < T_{\text{pred+tran}} - T_x | w_0 = T_x - T_0\} \\ & \times P\{w_0 = T_x - T_0\}\, \mathrm{d}T_x \end{aligned} \tag{12}$$

$$P_{\text{pred}}^{(4)} = \int_{T_0}^{T_{\text{pred}}} \big(1 - \mathrm{e}^{-\gamma T_x}\big)\big(1 - \mathrm{e}^{-\gamma (T_{\text{pred+tran}} - T_x)}\big)\, \mathrm{d}T_x. \tag{13}$$

The RTS-CTS probe introduces a small overhead in the protocol in the PCF mode. For mobile devices experiencing continuous fading conditions, the predictor will provide enhanced throughput. In contrast, mobile devices continuously experiencing a good link will receive little benefit from the use of the predictor channel probe. The downside of this scheme is the overhead for sending the probe pair for each data packet transmission. An enhancement to this approach is to have a simple mechanism that turns the predictor off when the channel has been in a good state for some time and then turn it on only when a packet is observed to be erroneous.

Since the predictor avoids unwarranted retransmissions to a receiver in a bad channel state, the channel's throughput is enhanced. In section 4, we present an evaluation of the predictor mechanism. Channel prediction, however, does not provide any compensation techniques for receivers which deferred transmission in the past due to bad channel state conditions [1]. Although receivers in a good state can benefit from the deferred transmission of other receivers in bad states, they are not typically compensated after the state of the deferred receiver becomes good. Therefore, a mechanism to "credit" and "compensate" flows/sessions becomes necessary.

5. The compensator

Our compensator uses a modified version of Deficit Round Robin (DRR) [15] to "credit" and "compensate" flows in response to potential unfairness experienced by mobile devices due to different channel conditions. Transmission of data packets using DRR is controlled by the use of a *quantum* (Q) and a *deficit counter* (DC) [15]. The quantum accounts for the quota of bytes given to each flow for transmission in each round, whereas the deficit counter keeps track of the transmission credit history for each flow. A "round" is defined as the process of visiting each of the queues in the scheduler once. At the beginning of each round, a quantum is added to the deficit counter for each flow. The scheduler visits each flow comparing the size of the deficit counter with the size of the packet at the head of the queue. As long as the packet size is smaller than the deficit counter the packet will be transmitted over the wireless link and the deficit counter reduced by the packet size. When the packet size is greater than the deficit counter the transmission of the packet is simply deferred. In this case, the scheduler does not decrement the deficit value in the flow-state table for the next round but simply moves to the next flow in a round robin order. As long as the quantum size is larger than the maximum packet size the system is work-conserving [15].

In the case where the quantum size for all flows is the same, an equal allocation of the wireless link is achieved. Making the quantum size for some flows different leads to Weighted Round Robin (WRR), which allows for a proportional sharing of the wireless link according to the weights given to each flow.

5.1. Operations

We modified the Deficit Round Robin algorithm by adding a *compensation counter* (CC) that is maintained for each receiver. The compensator counter maintains the necessary state information when the mobile device defers transmission. For each round, αCC_i additional bytes are allocated if the compensation counter for $flow_i$ is positive, where α is a value between 0 and 1. The value of α represents the fraction of the compensation credit given to a flow in one round. Each time

αCC bytes are consumed to compensate a flow its compensation counter is decreased by the same amount. This action compensates receiver sessions which were deferred in the previous rounds due to bad channel conditions. To this end, even if the channel has estimated a bad channel state (hence, the data packet is not transmitted), the deficit counter for the receiver is decreased by the quantum size. In return for this decrease, the compensation counter for the session is increased by the quantum size. The actual compensation may vary between 0 and the quantum size according to the observed load of the system.

The compensation process meets two goals. First, it determines how many bytes to credit a flow by after the predictor observes a bad channel. Second, it determines what portion of the credit is used for compensation of the flow in each round.

Considering the first goal, it is intuitive to credit Q bytes every time transmission is deferred. When the network is heavily loaded this is a good solution. However, when the system is lightly loaded the rate at which the round robin scheduler serves a flow is typically faster than the worst case scenario of full network load. Crediting Q bytes at this rate will over-credit lagging flows, leading to unfairness for newly arriving flows over the long term. In this paper, we propose to credit flows according to the load of the system providing less credit in lightly loaded systems and a quantum size of credit under heavily loaded conditions. More formally, if n flows are registered with the central scheduler at the wireless access point (each $flow_i$ with a weight Q_i), the load of the system is defined as the ratio of the sum of Q_i for active flows (Q_i^{Act}) and G which represent the total capacity of the system in each round. We consider an "active" flow to be one that has at least one packet in the scheduler's queue. The definition of G can be considered arbitrary but has to be consistent. For example, if G is set to 1000 and a particular flow receives a 15% share of the wireless link the quantum size for the flow should be set to 150. Let $CC_i(k)$ be the compensation counter for flow i in round k then if $flow_i$ deferred transmission in round k the compensation counter in round $k+1$ will be:

$$CC_i(k+1) = \begin{cases} CC_i(k) + \left(\dfrac{\sum_{j=1}^{n} Q_j^{\mathrm{Act}} - Q_i}{(G - Q_i)}\right) Q_i \\ \quad \text{if } G > Q_i, \\ CC_i(k) \\ \quad \text{if } G = Q_i. \end{cases} \tag{14}$$

Only when $G = \sum_{i=1}^{n} Q_i^{\mathrm{Act}}$ is the system operating at full load and the compensation given to $flow_i$ is equal to Q_i. When $\sum_{i=1}^{n} Q_i^{\mathrm{Act}} = Q_i$, only $flow_i$ is active resulting in no compensation to avoid over crediting $flow_i$, as previously discussed.

In what follows, we analyze the second goal discussed above related to how many bytes of credit should be used for compensation in a single round. It is desirable to compensate a flow that is behind schedule as soon as possible due to the potential real time requirements of a flow. This calls for adding CC bytes to DC in one round no matter what the size of CC. The problem with this approach is that the latency for flows is likely to be sensitive to the amount of compensation that is given to each flow in each round especially during loaded periods. In order to attempt to bound the latency it is necessary to bound the maximum compensation that a flow is given in a single round. Similar to the approach discussed in [4], we bound the maximum amount of bytes $flow_i$ can transmit in one round to a constant parameter $DC_i^{\max}$ even under loaded conditions.

Let $\sum_{i=1}^{n} Q_i^{+}$ be the sum of the quantum for flows having positive compensation counters (e.g., $Q_i^{+} = 0$ if $CC_i = 0$ and $Q_i^{+} = Q_i$ if $CC_i > 0$), then the number of bytes available for the compensation to $flow_i$ in one round is given by

$$\alpha CC_i = \min\left[\max\left[\left(\frac{Q_i^{+}}{\sum_{j=1}^{n} Q_j^{+}}\right)\left(G - \sum_{j=1}^{n} Q_j^{\mathrm{Act}}\right), DC_i^{\max} - Q_i\right], CC_i\right]. \tag{15}$$

The first term inside the brackets in equation (15) accounts for the compensation in the case where unused bandwidth is available. This is obtained by computing the available bandwidth and the weighted portion of bandwidth that corresponds to each flow with a positive CC. The second term in equation (15) ($DC_i^{\max} - Q_i$), accounts for the minimum compensation given to a flow in one round in case where the system is heavily loaded and there is no unused bandwidth available. In both cases the amount of compensation given to $flow_i$ is bounded by CC_i.

An illustration of the operation of the compensator is shown in figure 3. The figure shows a snapshot of the scheduler at the beginning of a round and after the quantum and compensation bytes have been added. The scenario shows three active flows associated with three different mobile devices with the sum of the allocated rates equal to the system capacity (i.e., the system is fully loaded). In this example, $DC_i^{\max} = 2Q_i$ for each $flow_i$. Figure 3(b) illustrates the state of the scheduler at the end of the first round. The following events take place during the round: (i) channel prediction for $flow_1$ detects a bad channel and the scheduler defers the transmission of the packet, updates the compensation counter by the quantum size and reduces the deficit counter by the same amount; (ii) prediction for $flow_2$ indicates a good channel so the scheduler transmits the packet reducing the deficit counter by the packet size which is a normal weighted round robin operation; and finally (iii) prediction for $flow_3$ indicates a good channel, so two packets are transmitted and the deficit counter is decreased by the packet size. Figure 3(c) illustrates the state of the scheduler at the beginning of the next round when Q_i plus αCC_i bytes are added to the deficit counter for each flow i if the compensation counter is positive. Note that only a portion of CC for $flow_1$ and $flow_3$ is added to their deficit counter so that $DC_i \leqslant DC_i^{\max}$.

The choice of $DC^{\max}$ is a design parameter. Choosing a small $DC^{\max}$ will reduce the latency bound but increase a flow's compensation time. In contrast, choosing a large $DC^{\max}$ increases the latency bound during periods of heavy

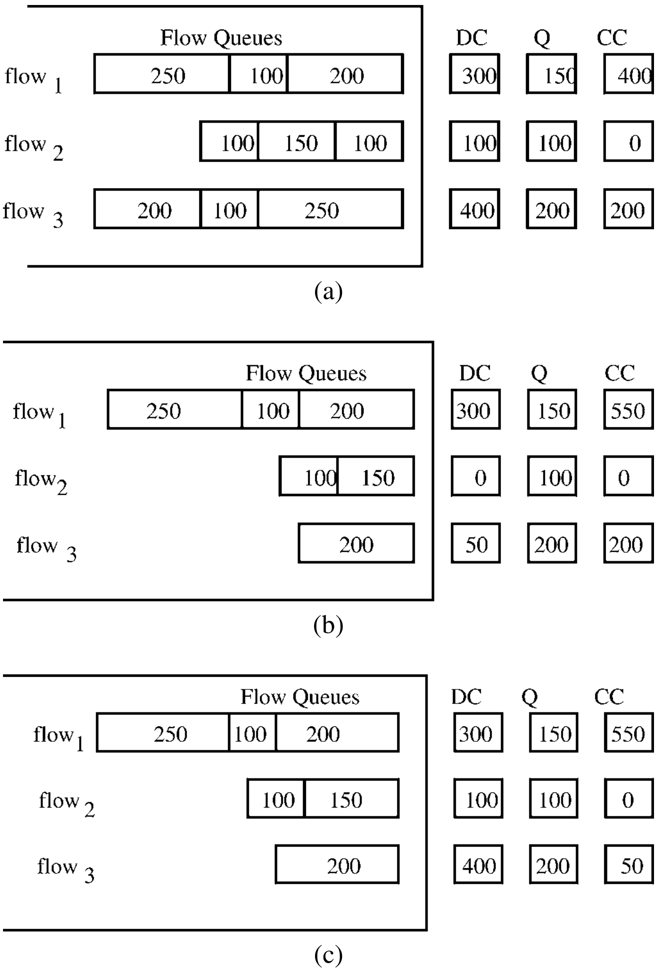

Figure 3. An illustration of the compensator operation. (a) Beginning of round k. (b) End of round k. (c) Beginning of round $k+1$.

load but decreases the compensation time. Since only a fraction of CC is used for compensation, CC can become large without affecting the latency bound of flows in the system. Because of this observation we do not limit the maximum size of the compensation counter.

5.2. Delay analysis

The latency bound provided by WRR is given by $\sum_{i=1}^{n} Q_i/C$ [15], where C represents the transmission speed of the wireless LAN when n flows are being serviced by the scheduler. This equation is valid only when the quantum size is greater than the maximum packet length, which is a necessary condition in DRR to make the system work-conserving. Otherwise, Q_i should be replaced by the maximum packet size. We can interpret from this equation that a small packet arriving at the head of a queue can be delayed by a quantum size due to other flows in the scheduler. In this case, the quantum size could be greater than the default size (Q) when compensation bytes are added to the deficit counter; therefore, the latency bound is given by

$$LatencyBound = \frac{\sum_{i=1}^{n} DC_i^{\max}}{C}. \tag{16}$$

This equation does not take into account the time taken for the RTS-CTS packet exchange. This delay is approximately 3 ms for an IEEE 802.11 radio operating at 2 Mbps. For small packets this delay can generate a large overhead. The value of $DC^{\max}$ represents the maximum amount of bytes that flows can transmit in one round. Therefore, $DC^{\max}$ determines the time required to fully compensate a flow as well as the worst case latency bound. This latency bound does not represent the worst case packet delay but the worst-case channel prediction delay. Clearly the scheduler has little control over how long the channel remains in a bad state. The scheduler can, however, bound the time taken between channel predictions for each flow.

If the channel is in a bad state and the transmission of a packet is deferred, then system should attempt to probe the channel as soon as possible. Experimental results show [1], however, that fading periods are usually correlated. Therefore, waiting some time before testing the channel may be intuitive. On the other hand, waiting too long to test the channel may lead to poor performance because the scheduler may miss periods in which the channel is good and capable of transmission and delivery of packets. The fact that DRR visits flows at discrete times (once every round) complements the probing time of the predictor. Determining the optimal interval and channel probing period during channel fades is an open research issue that is dependent on how well the duration of bad periods can be accurately estimated.

5.3. Fairness

The fairness property of DRR is analyzed in [15]. The compensator achieves fairness even in the presence of channel fading. In what follows, we discuss the fairness properties of the compensator under full network load assumptions. Using the same nomenclature as [15], let $DC_i(k)$ and $CC_i(k)$ be the value of the deficit counter and compensation counter, respectively, for $flow_i$ at the end of round k. Let $bytes_i(k)$ be the bytes sent by $flow_i$ in round k, and let $sent_i(k)$ be the sum of bytes sent by $flow_i$ in rounds 1 through k, clearly $sent_{i,K} = \sum_{k=1}^{K} bytes_i(k)$. Based on the protocol description the next equation follows:

$$\begin{aligned} &bytes_i(k) + DC_i(k) + CC_i(k) \\ &\quad = Q_i + DC_i(k-1) + CC_i(k-1). \end{aligned} \tag{17}$$

In order to prove fairness for the compensator, we must first consider a scenario when the mobile device first defers scheduled transmission due to bad channel prediction in some rounds and then when the channel is predicted to be in a good state and compensation is provided to the mobile device. Let us now assume that $DC_{i,0} = CC_{i,0} = 0$ as the initial conditions for round 1 letting the predictor diagnose a bad channel state for the next N rounds. The fact that the mobile device defers transmission for rounds $k = 1$ through $k = N$ implies that $sent_{i,N} = 0$. It follows immediately that

$$DC_i(k+1) = DC_i(k) = DC_i^{\max} - Q_i, \quad 1 \leqslant k \leqslant N. \tag{18}$$

Using this result in equation (17) we get

$$CC_i(k+1) = Q_i + CC_i(k), \quad 1 \leqslant k \leqslant N. \tag{19}$$

Now let us assume that the predictor diagnoses a good channel state for rounds $k = N+1$. In this case, the mobile device transmits packets and will be compensated for the previous rounds it deferred transmission. Then the amount of bytes of compensation given in one round is

$$CC_i(k-1) - CC_i(k) = \left(DC_i^{\max} - Q_i - DC_i(k-1)\right), \quad k > N. \tag{20}$$

Since $DC_i(k-1) = DC_i^{\max} - bytes_i^{\text{pred}}(k-1)$, equation (20) can also be written as

$$CC_i(k-1) - CC_i(k) = bytes_i^{\text{pred}}(k-1) - Q_i, \quad k > N, \tag{21}$$

where $bytes^{\text{pred}}$ shows the dependency of the number of bytes transmitted successfully based on the accuracy of the prediction (e.g., $bytes^{\text{pred}} = 0$ if the packet is corrupted by channel errors not detected by the predictor). Since conditions at round N are $CC_i(N) = NQ$ and $DC_i(N) = DC_i^{\max} - Q_i$, we get

$$CC_i(k) = kDC_i^{\max} - bytes_i^{\text{pred}}(k) - \cdots - bytes_i^{\text{pred}}(N), \quad k > N, \tag{22}$$

or

$$CC_i(k) = kDC_i^{\max} - sent_i^{\text{pred}}(k), \quad k > N. \tag{23}$$

Clearly, compensation of $flow_i$ will occur as long as $CC_i(k)$ remains positive and will stop when equal to zero. The ideal number of bytes allocated to flow i in WRR after k rounds under normal conditions (i.e., persistently good channel) is $sent_i(k) = kQ_i$ [15]. Subtracting this from equation (23), it follows that as soon as $CC_i(k)$ equals zero, a flow reaches its ideal bandwidth allocation (e.g., the flow has been fully compensated). The mobile device can only transmit after round N when the channel is predicted in a good state. Since $bytes_i^{\text{pred}}(k)$ is always smaller than $DC_i^{\max}$, then as long as $DC_i^{\max} > Q_i$ a flow will reach its bandwidth allocation. The main implication of this analysis, is that even if the mobile device goes into a deep fade, as long as the channel becomes good in the future fairness can be reached.

Fairness in practical situations does not hold when channel prediction fails because packets are transmitted and corrupted by channel errors not anticipated by the predictor. In this situation, the accuracy of the predictor plays an important role in our integrated adaptive-QoS system. The fairness properties of the compensator assume that the buffer space is infinite and packets can be buffered indefinitely. Buffer space is, however, a finite resource. If bad channel periods persist then buffers build up and arriving packets may be dropped. Therefore, an adaptation mechanism that can respond to these conditions over longer time scales than discussed previously becomes important.

6. The adaptor

The final component of the Havana framework exploits the ability of applications to adapt to channel dependent conditions or variations in available bandwidth over longer time scales. The adaptor includes two components that support the notion of adaptive wireless services; these are:

(1) *a buffer controller*, which operates over the wireless hop; and

(2) *a regulator*, which operates on an end-to-end basis.

The buffer controller responds to adverse network conditions by dropping low priority packets while the regulator performs end-to-end rate control over longer time scales if adverse network conditions persist. The buffer controller and regulator work in unison to deliver adaptive wireless services to mobile devices. The regulator assumes that there is a buffer controller at the access point (as illustrated in figure 4(a)) which responds to severe network conditions experienced over the wireless hop. Conversely, the buffer controller assumes that there is a regulator operating on an end-to-end basis that maintains low buffer occupancy rates at wireless access points.

The Havana framework supports the delivery of adaptive wireless services to mobile devices that require minimum and maximum bandwidth assurances. Adaptive wireless services are regulated by the buffer controller and end-to-end regulator and attempt to keep service semantics meaningful to the user during periods of changes in channel conditions or available bandwidth. Minimum bandwidth provides support for a *base-QoS* whereas maximum bandwidth supports *enhanced-QoS*.

Adaptive wireless services provide preferential delivery of base-QoS when the channel conditions degrade, and enhanced-QoS, when additional bandwidth becomes available or as channel conditions improve. The buffer controller is responsible for dropping packets in response to these conditions. Different priorities are represented using a priority field in the IP packets associated with a flow. A signaling protocol based on INSIGNIA [9] is used to establish adaptive wireless services on an end-to-end basis and also to report the measured receiver's QoS to the source node. Both base and enhanced-QoS require admission control to establish adaptive wireless services. The adaptor can also be used to provide power-aware QoS depending on the local power reserves of wireless devices. In this respect, the base station may filter enhanced-QoS packets when the power resources of the receiver mobile device are getting low.

Many existing transport protocols (e.g., TCP) are not well suited to delivering multimedia over existing IP networks. In contrast, UDP is more suited to this task but lacks any adaptation capability. RTP [14] does, however, support adaptation mechanisms allowing applications to regulate and respond to observed network conditions (e.g., jitter and bandwidth availability). Using end-to-end regulation in this manner limits the likelihood of persistent high buffer occupancy rates for

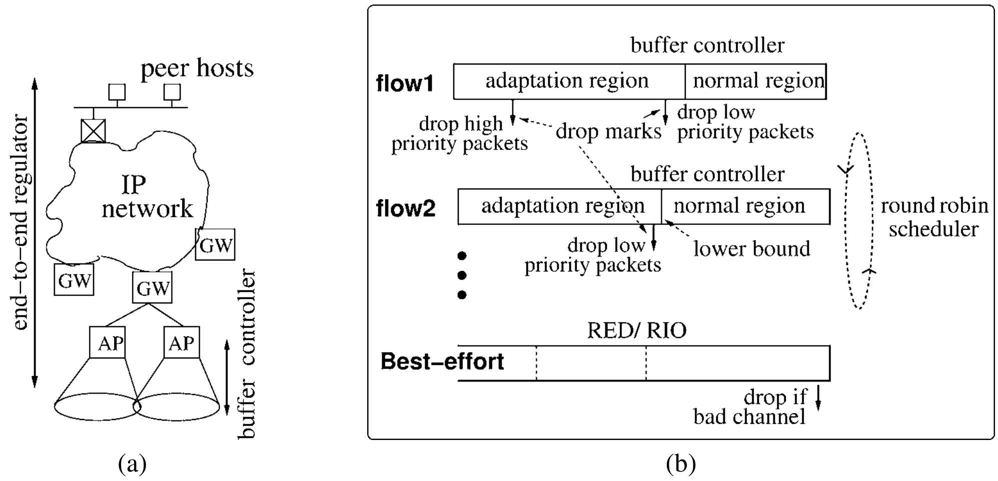

Figure 4. Adaptor model. (a) Wireless access network. (b) Adaptor scenarios.

queues maintained at the wireless access point during periods of channel or bandwidth degradation. The end-to-end regulator responds to degradation over longer time scales by regulating source traffic to match it to the bottleneck bandwidth experienced at the wireless access point. Typically, this timescale is at least a round trip period.

Enhanced-QoS packets are dropped before base-QoS packets in the case of congestion or persistent poor channel conditions. In the Havana framework, the buffer controller supports this operation by partitioning per mobile device buffers in the wireless access point into normal and adaptation regions, as illustrated in figure 4(b):

- *Normal region*. During normal operations the buffer occupancy is likely to be small when the channel is in a good state. The lowest position for the drop mark is delimited by a "lower" bound, as illustrated in figure 4(b).
- *Adaptation region*. When severe channel degradation persists, the buffer occupancy can reach high levels. In this case, the controller may be forced to drop packets to maintain service semantics. The adaptor sets "*drop marks*" in the adaptation region in per mobile device buffers. When the buffer occupancy goes above these drop marks, the arbitrator notifies the adaptor which configures suitable filters in the packet classifier to drop low priority packets (e.g., enhanced-QoS packets), as illustrated in figure 4(b).

Several access point buffering scenarios are illustrated in figure 4(b). In the figure, two adaptive flows are supported by per-mobile queues with all best effort traffic aggregated into a single best effort queue. Flow 1 in figure 4(b) illustrates the case where a flow consists of three different priorities. This could represent a video flow that comprises a base layer and two enhancement layers. Flow 2 has two priorities, as indicated in figure 4(b). This could represent a web session with base text information and enhanced picture quality or audio and video multiplexed into a single end-to-end session.

The optimal position of the drop marks in per mobile buffers depends on the acceptable buffer occupancy. Assuming a regulator operates on an end-to-end basis (e.g., TCP flow control or RTP rate control), we expect the source to match its rate to the available bandwidth at the wireless access point. When the predictor is operational, the average number of packets in the buffer will increase as the length of fade periods increase. If the average buffer occupancy is small the drop mark should be correspondingly large. This allows the wireless link to operate at relatively high utilization without dropping packets. When the buffer overflows periodically, the drop mark should be set small. This allows the buffer controller to drop low priority packets at the earliest opportunity saving buffer space for high priority packets in case of persistently bad network conditions.

7. Evaluation

In this section, we provide an evaluation of the Havana framework, which is implemented using the *ns*-2 simulator [11]. We present the evaluation of the Havana system and its components in an incremental manner. First, we analyze the performance of the predictor module in isolation from all other components. Next, we add the compensator module to the predictor and show the benefits of the compensation scheme to credit and compensate flows under a variety of wireless channel conditions. Finally, we add the adaptor to the predictor and compensator and show how the composite Havana system works in unison to deliver application and channel dependent QoS in wireless packet networks.

7.1. Channel prediction

Two main factors govern the accuracy of the channel predictor: (i) the packet size influences accuracy (e.g., channel prediction for small packets is typically more accurate in comparison to larger packets); and (ii) the rate at which the radio channel changes between good and bad state. In what follows, we analyze a single wireless hop between an access point and several mobile devices based on IEEE 802.11 operating at 2 Mbps. The IEEE 802.11 code suite [11] was modified for the predictor to operate in the PCF mode, as discussed

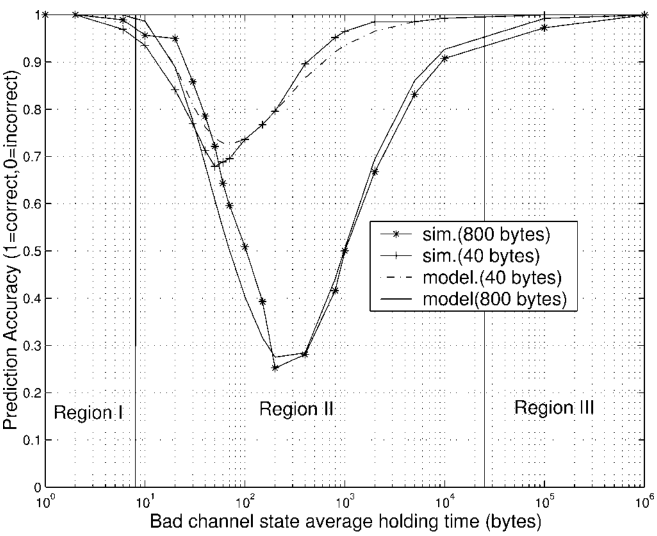

Figure 5. Predictor accuracy versus packet size.

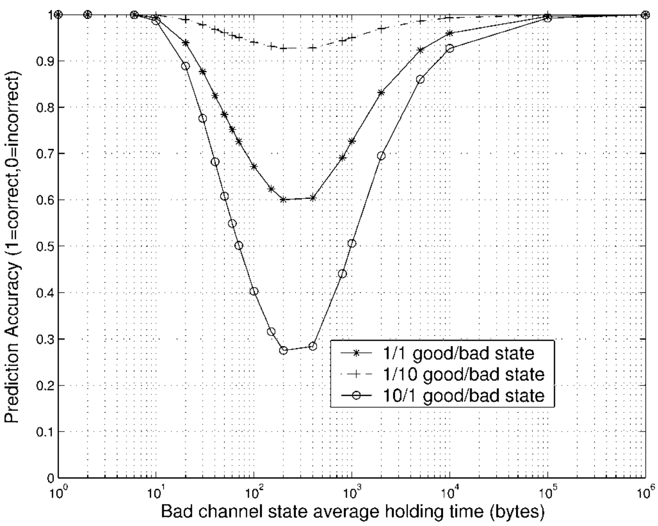

Figure 6. Predictor accuracy versus good/bad state ratio.

in section 3.1. Each mobile device receives a constant bit rate stream with the same packet size used for all flows. A two-state Markov model is used to model the good and bad state transitions of the wireless channel. Even if the state of the channel is predicted in error, we continue with the transmission of data packets to verify the accuracy of the prediction.

Figure 5 shows simulation and analytical results for the predictor scheme discussed in section 4.1 using packet sizes of 800 and 40 bytes. RTS and CTS packets are 20 bytes in length as defined by the IEEE 802.11 standard. Each point on the x-axis represents different wireless channel conditions (e.g., average duration of good and bad state periods) whereas the y-axis shows the probability of making a good channel prediction. We use bytes as a measure of holding times in figure 5 because it is easier to compare the holding times of good and bad channel states with the size of the packet. Good channel holding times are 10 times longer than the bad channel holding times. Each point in the figure represents an average of approximately 1000 packet transmissions.

Figure 5 shows that analytical and simulation results closely follow each other. We have divided figure 5 into three different regions of interest:

Region I. Transitions between good and bad states occur at very high frequency (e.g., every few bytes). In this case, the RTS-CTS and DATA-ACK packets are corrupt due to channel errors with very high probability resulting in accurate channel prediction.

Region II. Transition between good and bad states becomes similar to the packet transmission time scale causing the accuracy of the predictor to decrease rapidly. This is because the channel observed by the predictor packets (RTS-CTS) and the channel observed by the data packets (DATA-ACK) may be different leading to incorrect channel prediction.

Region III. Transitions between good and bad states are 2–3 orders of magnitude greater than the packet transmission time scales; therefore RTS-CTS and DATA-ACK packets will observe the same channel resulting in an accurate prediction.

Figure 5 shows the evaluation of the predictor performance for two different packet sizes. As indicated in the figure, the accuracy of the predictor becomes less effective as the packet size increases. The optimal transmission packet size for wireless LANs (e.g., 1.5 Kbytes in IEEE 802.11) is relatively large. For communication systems where large packets are commonplace this will inevitably lead to an increase in the uncertainty of channel prediction algorithms.

Figure 6 shows simulation results for channel prediction for a packet size of 800 bytes and three different good/bad channel state ratios (viz. 10/1, 1/1, 1/10). Each point in this figure represents an average of 1000 packet transmissions. When the channel ratio is high (e.g., 10/1) the accuracy of the predictor diminishes because bad channel periods corrupt either RTS-CTS or DATA-ACK set of packets but not both of them leading to poor channel prediction. When the channel ratio is small (e.g., 1/10), bad channel periods are likely to corrupt both RTS-CTS and DATA-ACK packets leading to good channel prediction.

We observed from figures 5 and 6 that the accuracy of the predictor drops off as the channel state transitions become similar to the packet transmission time scales. Other channel-mitigation techniques (e.g., forward error correction, interleaving, etc.) may, however, improve the accuracy of prediction for fading transitions at the packet transmission time scale.

7.2. Predictor and compensator

In what follows, we show the throughput achieved by mobile devices when combining the predictor and compensator mechanisms under the same channel conditions. We then compare the throughput for a wide range of channel conditions and establish boundary conditions on the performance of the predictor and compensator modules.

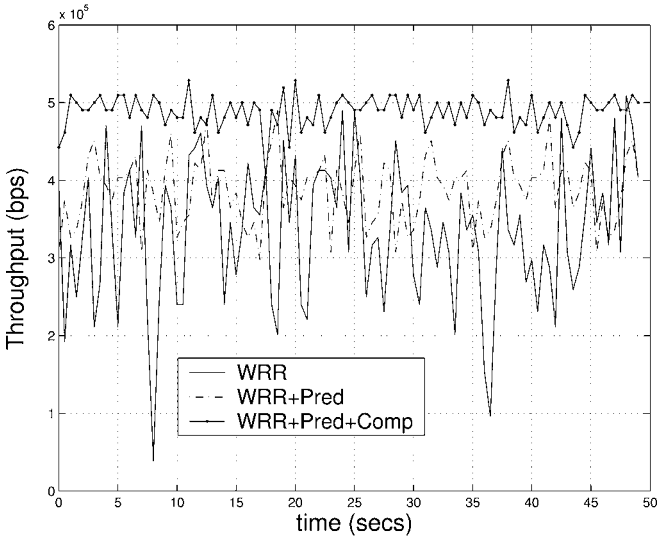

Figure 7. Throughput performance using prediction and compensation.

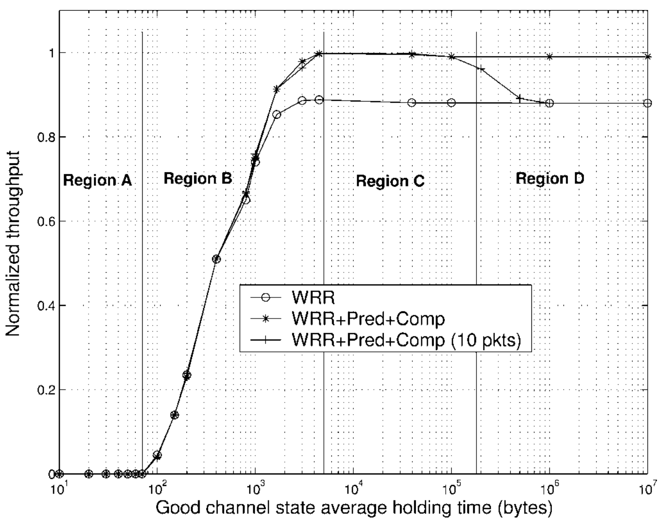

Figure 8. Throughput performance.

Using our *ns*-2 implementation for the predictor and compensator we simulate a single wireless LAN with ten active mobile devices to highlight the benefit of using the predictor and the compensator modules. Two of the mobile devices receive adaptive wireless service with a base-QoS of 500 Kbps with the packet size set to 500 bytes. The remaining mobile devices establish TCP sessions as background traffic consuming best-effort bandwidth. In this experiment, the WRR scheduler is configured to support the appropriate weights using an in-band signaling protocol for session establishment [9]. The average holding times in good and bad states are set to 20,000 bytes and 8,000 bytes, respectively[1]. These holding times represent a challenging environment to test the operation of the compensator. For the selected transition rates the predictor has an accuracy above 95%, as shown in figure 6. If the channel is predicted in error, transmission is deferred and transmission for the next flow/session is carried out, as described in section 5. In this example, $DC^{\max} = 2Q$, which results in the maximum amount of compensation bytes given to a flow in one round being twice the quantum size for that flow. Figure 7 shows throughput traces for one of the 500 Kbps flows under three different system configurations:

(1) the scheduler alone;

(2) the scheduler with the predictor; and

(3) the scheduler, predictor and compensator.

Through incrementally adding each module of the composite experimental system we can clearly evaluate the benefit of each module on the overall performance of the system. As shown in figure 7, when the standard WRR scheduler is used in isolation (i.e., without prediction and compensation) the effects of channel errors greatly diminish throughput. When the predictor is added to the system the throughput to mobile devices improves but does not reach the requested bandwidth of 500 Kbps, as illustrated in figure 7. The final configuration considers the predictor and compensator working in unison with the scheduler. The arbitrator detects bad channel conditions efficiently, deferring transmission and compensating flows when the link degrades and improves, respectively. In this case, the predictor and compensator deliver the requested 500 Kbps throughput to the mobile device, as illustrated in figure 7.

Figure 8 compares the throughput achieved for a wide range of wireless channel conditions under finite/infinite buffer conditions. The channel ratio is set to 10/1 for this experiment. Each point in the figure represents the average of 10 runs of the simulation. Three different system configurations are considered:

(1) the scheduler alone;

(2) the scheduler with the predictor and compensator with *infinite* buffer space; and

(3) the scheduler with the predictor and compensator with *finite* buffer space.

In the case of the final configuration, we configure the buffer capacity for flows to accommodate a maximum of 10 packets only. In figure 8, we divided the graph into four different regions of interest (viz. regions A, B, C, D):

Region A. Channel state transitions occur on a very fast time scale with a resulting throughput of zero.

Region B. Channel state transitions are fast, however, good state periods are long enough to allow the transmission of packets. We observed from the results that the later two configurations (2 and 3 above) perform as poorly as the first configuration (1 above) due to the poor accuracy of the predictor in this region.

Region C. Channel state transitions are 1–2 orders of magnitude greater than the packet transmission time scales resulting in very accurate prediction. However, the bad channel periods are not long enough to overflow the buffer in

[1] Dividing the value in bytes on the x-axis in figure 6 by 2 Mbps will provide the equivalent holding time in seconds.

region C. As a result, the later two configurations (2 and 3 above) are observed to have similar performance, as illustrated in figure 8.

Region D. Duration of bad channel periods are 3–4 orders of magnitude greater than the packet transmission time scales resulting in buffer overflow. As a result, for a good state holding time of 10^6 bytes the performance of the first and last configurations (1 and 3 above) are similar. In this experiment, the arbitrator makes no distinction between packets dropped. Region D in figure 8 represents the operational range over which the adaptor can provide enhanced-QoS when applying selective dropping.

7.3. Predictor, compensator and adaptor

The final part of our evaluation considers the complete composite system in operation. First, we analyze how buffer provisioning affects the throughput achieved by mobile devices in the presence of slow fading channel conditions. Next, we simulate how throughput is affected by performing early packet dropping for layered applications. We analyze both buffer provisioning and early packet dropping issues under channel conditions similar to region D in figure 8 where the buffer occupancy is likely to experience overflow conditions.

7.3.1. Buffer provisioning

We simulate how buffer provisioning impacts the throughput achieved by mobile devices using a single wireless LAN where two mobile devices receive adaptive wireless services with a base-QoS of 500 Kbps using the scheduler, predictor, compensator, adaptor and arbitrator for a packet size of 375 bytes. The remaining mobile devices establish TCP sessions as background traffic consuming best-effort bandwidth. Figure 9 shows the normalized throughput achieved by one of the mobile devices receiving an adaptive service for a buffer size of 5, 10 and 30 packets, respectively. As shown in figure 9, the throughput of the mobile device increases as the buffer capacity increases. This result is because the buffer is able to hold more packets before experiencing overflow due to persistent fading conditions. The down side of increasing the buffer size is both the use of additional memory in base stations and mobile devices as well as potentially increasing end-to-end delays associated with larger buffer occupancies. While the increase in buffer requirements can be leveraged by sharing a common pool of memory among several flows the delay issue remains. Real time applications may require an upper bound on the buffer size to limit end-to-end delays even if doing so reduces the overall throughput of a flow.

7.3.2. Early packet dropping

Now we analyze the impact of performing early packet dropping for the case of flows containing two priorities. Results of this experiment can be extended to flows containing multiple priorities. In this case, the arbitrator starts dropping low priority packets (e.g., enhanced-QoS packets) after the buffer occupancy surpasses a critical dropping mark. The simulation settings are the same as in figure 9 with a constant

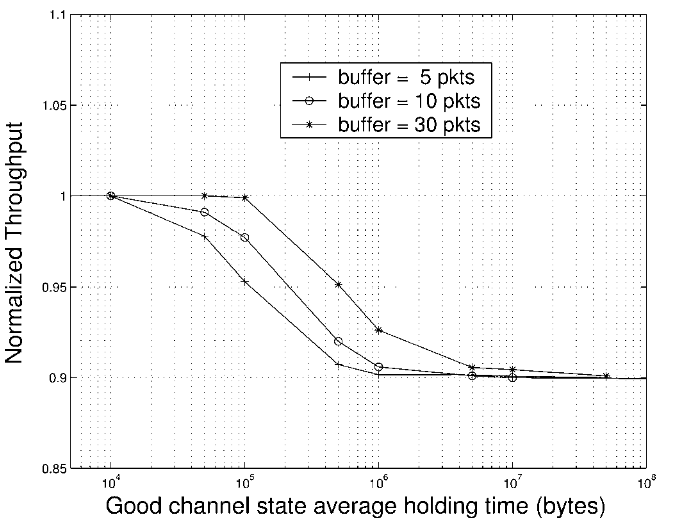

Figure 9. Adaptor: buffer provisioning impact on throughput.

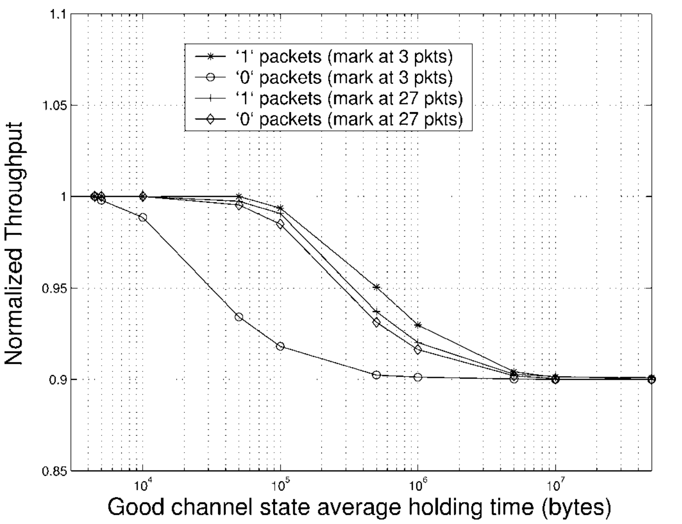

Figure 10. Adaptor: drop mark impact on throughput.

buffer size of 30 packets except that now the flow under test is composed of high and low priorities (e.g., base-QoS and enhanced-QoS), each of which accounts for 50% of the total flow rate. Figure 10 shows the normalized throughput of high and low priority "subflows" where the dropping mark is positioned at 3 and 27 stored packets from the head of the flow's queue, respectively. Dropping low priority packets after 3 or 27 stored packets contrasts the operation of the adaptor for early and late operations. As shown in figure 10, dropping low priority packets late (e.g., when the drop mark is positioned at 27 stored packets) has little effect on the throughput achieved by high and low priority subflows. In contrast, when the arbitrator drops low priority packets early (e.g., when the mark is positioned at 3 stored packets), the high priority subflow achieves a better throughput but this is at the expense of the throughput achieved by the low priority subflow.

The position of the drop mark also impacts the aggregated throughput of a flow, when considering high plus low priority subflows together. Figure 11 shows the the normalized aggregate throughput for the same experiment as in figure 10

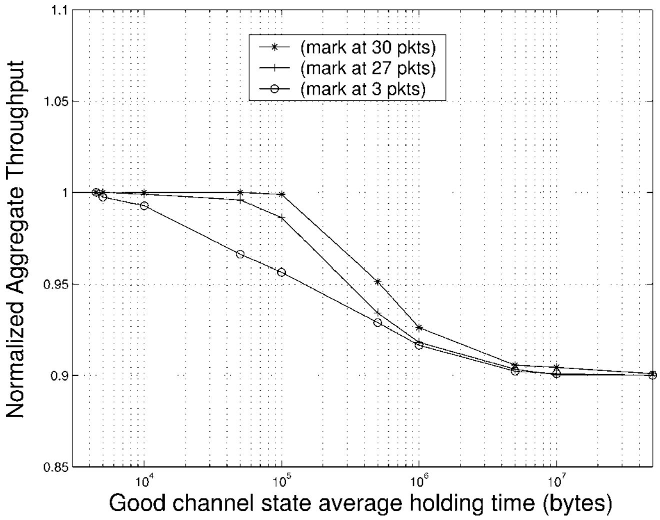

Figure 11. Adaptor: drop mark impact on utilization.

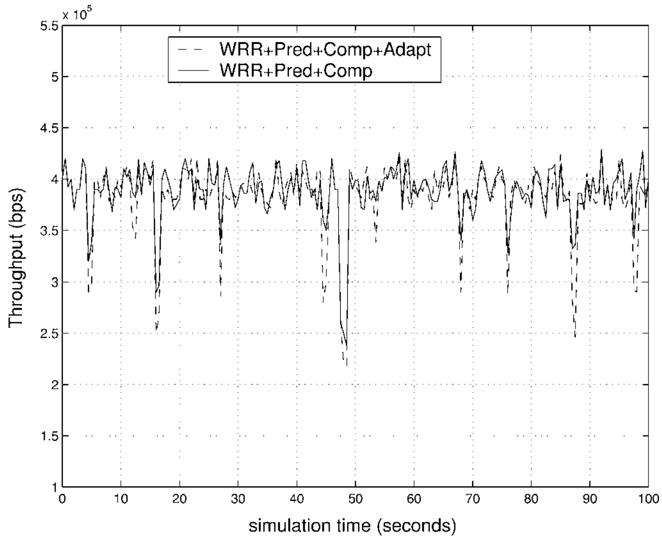

Figure 12. Adaptor throughput: early dropping versus no dropping.

for a drop mark positioned at 3 and 27 stored packets as well as for the case where there is no packet dropping (e.g., the drop mark is positioned at 30 stored packets). As shown in figure 11, aggregated throughput decreases as the drop mark is positioned near the head of the queue. This is the result of the effect of dropping low priority packets early in situations where fading conditions did not overflow the buffer.

Setting an optimum position for dropping marks is a difficult task because of the uncertainty of predicting future channel conditions on the flow/session time scale. Dropping low priority packets early may increase the rate at which high priority packets are being delivered but this is at the expense of lower flow throughput (e.g., less low priority packets are transmitted). The weights (e.g., cost based on semantics) of packets with different priorities can also be used to bias the throughput of one priority subflow against another. For example, a flow in which a low priority packet cannot be utilized without the corresponding high priority packet being received correctly may set the dropping mark earlier than later to increase the probability that high priority packets will arrive correctly.

In order to illustrate the effect of early packet dropping on layered applications we show one example where we simulate a single wireless LAN with three mobile devices receiving continuous media services using the scheduler, predictor, compensator, adaptor and arbitrator. Several mobile devices establish TCP sessions as background traffic that consumes best-effort bandwidth with these flows joining and leaving the system during the course of the simulation. A continuous media flow supports the delivery of video based on a "True Lies" MPEG-2 video clip which delivers a multi-resolution flow with two video layers of resolution. The base layer (BL) represents the main profile of MPEG-2 requiring 200 Kbps as base-QoS and one enhancement layer (E1) requiring 100 Kbps for enhanced-QoS. The average packet size is 1000 bytes long while the size of the buffer is configured to hold 30 packets and the drop mark is set after 5 stored packets. For this experiment we choose transition rates for the Markov model to be 10^6 and 10^5 bytes for good and bad state, respectively, where the buffer occupancy is likely to experience overflow conditions.

Figure 12 shows the throughput trace for the video flow for the case where early packet dropping is applied as well as for the case where packet dropping takes place only after overflow conditions (e.g., when no adaptor is used). When the predictor, compensator and adaptor are present, the arbitrator does not have to drop high priority packets (e.g., base-QoS packets) except in the case of deep fades which happen at 17, 47 and 86 s into the trace. At 4, 26, 45, 68, 76, 98 s into the trace early packet dropping allows the adaptor to store and deliver base-QoS packets while this is not true for the case when no adaptor is used at 4, 26, 76 and 98 s into the trace when base-QoS packets are dropped.

Comparing the two traces shown in figure 12, we observed that the non-adaptor trace achieves a higher throughput compared with the adaptor configured trace during fading periods. However, figure 12 does not fully convey the behavior of the system because this observation alone does not provide the complete picture. In fact disruption to video delivery occurs only 3 times (at 16, 47 and 77 s into the trace) when early packet dropping is used compared with 7 instances when no early dropping takes place. Therefore, the adaptor is able to maintain the minimum quality for a longer duration even when the resulting throughput is smaller than a system where no adaptor is configured.

7.4. Discussion

In what follows, we illustrate some scenarios showing how different applications can customize the operation of predictor, compensator and adaptor to suit specific requirements. We split the adaptor module into two parts that accounts for buffer provisioning and drop marks positioning. Table 2 illustrates several scenarios for different applications. In these examples, we assume that a regulator operates on an end-to-end basis so any degradation in network conditions is mainly

Table 2
Havana system component settings for some common applications.

Applications	Predictor	Compensator	Adaptor	
			Buffer provisioning	Early packet dropping
Data (one buffer per flow)	ON	optional	A larger buffer increases the resilience of the flow in case of deep fades.	NA
Data (several flows aggregated in a single buffer)	ON	optional	A larger buffer allows for better multiplexing of multiple sources.	Packets must be dropped if bad channel conditions appear to avoid the head-of-line blocking problem.
Multi-priority data (one buffer per flow)	ON	optional	A larger buffer increases the resilience of the flow in case of deep fades.	Dropping marks should be set according to flow semantics, as discussed in section 6.
Multi-priority data (several flows aggregated in a single buffer), Similar to the DiffServ Assured Service	ON	optional	A larger buffer allows a better multiplexing of multiple sources.	Packet must be dropped if bad channel conditions appear. However, drop marks can be set to prevent buffer overflow in case of bursty data.
Real time media with a single priority (single buffer per flow)	ON	ON	Small-medium size buffer depending of the maximum delay requirements of the application.	NA
Multi-priority real-time media (single buffer per flow)	ON	ON	Small-medium size buffer depending of the maximum delay requirements of the application.	Dropping marks should be set according to flow semantics, as discussed in section 7.3.

due to bad channel conditions. Early packet dropping is not available (NA) for those applications with only one priority. In this table we assume different flows corresponds to different mobile devices.

8. Conclusion

In this paper, we have discussed the components of the Havana framework for wireless packet networks that include a predictor, compensator and adaptor all governed by an arbitrator. We believe that the predictor, compensator and adaptor mechanisms should work in unison rather than in isolation to deliver adaptive wireless services. The implementation discussed in this paper is based on IEEE 802.11, however, the ideas and results presented are broadly applicable to emerging wireless protocol that need to respond to QoS fluctuations in a controlled manner.

Simulation results have been presented. The results indicate that channel prediction accuracy diminishes quickly as the packet transmission time scales increase and as the channel state transitions approximate the packet transmission time scale. The impact of the accuracy of channel prediction on the performance of the compensator was analyzed. Simulation results indicate that the compensator is capable of achieving fairness among flows in fading environments unless the channel predictor fails or buffer overflow occurs.

We have discussed the notion of application specific adaptation. The adaptor module exploits the ability of applications to adapt to longer-average changes in available bandwidth as well as shorter times scales changes such as channel degradation when compensation alone would be inadequate. The adaptor discussed in this paper attempts to keep the deliver of packets semantically meaningful to applications by dropping lower priority packets first in responses to degradation in channel conditions and available bandwidth. Simulation results indicate that an integrated approach governed by the arbitrator and comprising the predictor, scheduler, compensator and adaptor provide an effective approach to delivering application and channel dependent QoS in wireless packet networks.

Acknowledgement

This work was support in part by the Army Research Office (ARO) under Award DAAD19-99-1-0287 and with support from COMET Group industrial sponsors. In particular, we would like to acknowledge the support of Ericsson, Intel, Nortel Networks and IBM.

References

[1] P. Bhagwat, P. Bhattacharya, A. Krishna and S. Tripathi, Enhancing throughput over wireless LANs using channel state dependent packet scheduling, in: *Proceedings of IEEE INFOCOM*, Kobe, Japan (April 1997).

[2] A.T. Campbell, J. Gomez, S. Kimand, Z. Turanyi, C.-Y. Wan and A.G. Valko, Design, implementation, and evaluation of cellular IP, IEEE Personal Communications, Special Issue on IP-based Mobile Telecommunications Networks (August 2000) 42–49.

[3] D. Clark and J. Wroclawski, An approach to service allocation in the Internet, Work in progress, `draft-clark-diff-svc-alloc-00.txt` (1997).

[4] T.S. Eugene, I. Stoica and H. Zhang, Packet fair queueing algorithms for wireless networks with location-dependent errors, in: *Proceedings of IEEE INFOCOM*, San Francisco, CA (March 1998).

[5] S. Floyd and V. Jacobson, Randon early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking (August 1993).

[6] C. Fragouli, V. Sivaraman and M. Srivastava, Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling, in: *Proceedings of IEEE INFOCOM*, San Francisco, CA (March 1998).

[7] J. Gomez, A.T. Campbell, M. Naghshineh and C. Bisdikian, Conserving transmission power in wireless ad hoc networks, in: *Proceedings of IEEE 9th International Conference on Network Protocols (ICNP)*, Riverside, CA (November 2001).

[8] R.H. Katz, Adaptation and mobility in wireless information systems, IEEE Personal Communications Magazine 1(1) (First Quarter 1994).

[9] S. Lee and A. Campbell, INSIGNIA: In-band signaling support for QoS in mobile ad hoc networks, in: *Proceedings of IEEE International Workshop on Mobile Multimedia Communications (MOMUC)*, Berlin (October 1998).

[10] S. Lu, V. Bharghavan and R. Srikant, Fair scheduling in wireless packet networks, in: *Proceedings of ACM SIGCOMM*, San Francisco, CA (1997).

[11] G. Nguyen, Wireless features in *ns* simulator (1998) `http://www.cs.berkeley.edu/~nguyen/ns/`

[12] K. Nichols, V. Jacobson and L. Zhang, A two-bit Differentiated Services Architecture for the Internet, Internet Draft, Work in progress, `draft-nichols-diff-svc-arch-00.txt` (1997).

[13] P802.11, IEEE Standard for wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 802.11 (November 1997).

[14] H. Schulzrinne, RTP: The real-time Transport Protocol, in: *Proceedings of MCNC 2nd Packet Video Workshop*, Vol. 2, North Carolina (December 1992).

[15] M. Shreedhar and G. Varghese, Efficient fair queueing using deficit round robin, in: *Proceedings of ACM SIGCOMM*, Berkeley, CA (1995).

[16] M. Zorzi and R. Rao, Error control strategies for the wireless channel, in: *Proceedings of IEEE International Conference on Universal Personal Communications (ICUPC)*, Cambridge, MA (1996).

Javier Gomez Castellanos obtained a B.S. degree with honors in electrical engineering in 1993 from National Autonomous University of Mexico (UNAM), Mexico, and M.S. degree in electrical engineering in 1996 from Columbia University, New York. Since 1996 he has been a Ph.D. student in the COMET Group at the Center for Telecommunications Research, Columbia University, New York. His research interests cover routing, QoS, and power-aware design for cellular and ad-hoc wireless networks. He is a student member of the IEEE.
E-mail: javierg@comet.columbia.edu

Andrew T. Campbell is an Associate Professor in the Department of Electrical Engineering, Columbia University, and a member of the COMET Group and the Columbia Networking Research Center (CNRC). His research encompasses mobile computing and networking, open programmable networks, and QoS research. Dr. Campbell currently serves as technical program co-chair for the 8th ACM International Conference on Mobile Computing and Networking (ACM MobiCom 2002), and technical chair of the special track on networking technologies, services and protocols for IFIP Networking 2002. Dr. Campbell received his Ph.D. in computer science in 1996, the IBM Faculty Award in 1999 and the NSF CAREER Award for his research in programmable mobile networking in 1999.
E-mail: campbell@comet.columbia.edu