The Havana Framework for Supporting Application and Channel Dependent QOS in Wireless Networks

Javier Gomez, Andrew T. Campbell
Dept. of Electrical Engineering and
Center for Telecommunications Research.
Columbia University, NY 10027, USA
{javierg,campbell}@comet.columbia.edu

Hiro Morikawa
Department of Information and Communication
Engineering, The University of Tokyo
mori@mlab.t.u-tokyo.ac.jp

Abstract

For wireless channels, interference mitigation techniques are typically applied at the packet transmission level. In this paper, we present the Havana Framework for supporting integrated adaptive-QOS that also responds to impairments over multiple time scales that are present at the flow/session level. Our framework is based on three different mechanisms that operate over distinct adaptation time scales. At the packet transmission time scale, a 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 available bandwidth and channel conditions.

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 networks. One school of thought believes that the radio can be simply engineered to provide wireline type hard-QOS assurances (e.g., guaranteed delay or constant bit rate services). While others argue that the wireless link can not be viewed in this manner because of 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 [11].

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 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 component coordinating their operation in an integrated and systematic manner. Channel prediction allows the arbitrator to defer transmission to mobile devices experiencing time varying conditions (e.g., fading). Channel prediction, however, can not 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 support further adaptive actions. 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. 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 for wireless communications. Section 4 presents the evaluation of the system and its components in an incremental fashion. First, we analyze the performance of the predictor 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 system works in unison to deliver application and channel dependent QOS over wireless networks. We conclude in Section 5 with some final remarks.

2 Related Work

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

Much of the literature that discusses compensation mechanisms for flows in response to fading conditions assume either perfect prediction of the channel state or some apriori knowledge of the channel behavior. In [1] for example, fade periods are considered to last between 50 to 100 msec. Given this assumption, the scheduler defers transmission to a mobile device for a period of 50-100 msec when a fade occurs. In [3], the base station assumes it has instantaneous knowledge of channel conditions. In [12] 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 [5]; 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 [12]. Flows unable to transmit packets due to channel fading conditions are credited for future transmissions when the link returns to a good state again. This strategy, however, has the drawback that a flow coming out of a fade condition will be immediately 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 [3] 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 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) [14], 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) [4] or RED with 'In' and 'Out' priorities (RIO) [2] 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 and RIO 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. The buffer drop level in our adaptor scheme (as discussed in Section 3.3) can be dynamically set using a measurement-based protocol in comparison to RED and RIO where drop marks are typically statically configured in advance.

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 fade situations or variations in available bandwidth occurring anywhere between hundred 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 adaptive-QOS model introduced in this section attempts to take this time-varying behavior into account by operating over three distinct time scales in response to wireless network dynamics.

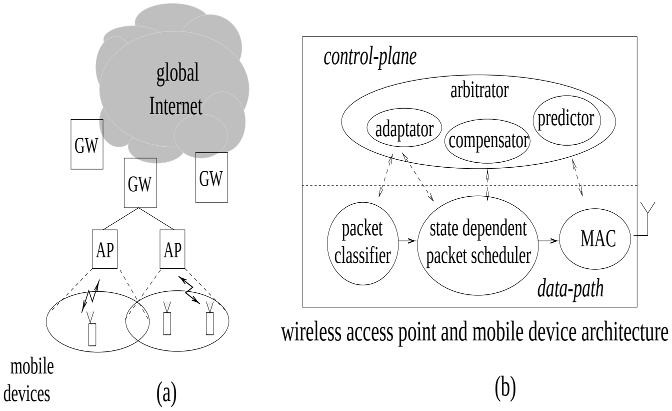


Figure 1. *The Havana Framework*

Figure 1 (a) and (b) illustrates the wireless network model and the Havana Framework, respectively. The wireless network model assumes that gateway routers (GW) interconnect a set of cellular access networks to the global Internet as illustrated in Figure 1 (a). A cellular access network comprises one or more forwarding nodes that can be configured to provide access points (AP) to mobile hosts. The main controller of the Havana Framework is a central arbitrator present at each wireless access point as illustrated in Figure 1 (b). The access point model comprises a $data\ path$, which includes a packet classifier, state dependent packet scheduler and MAC as illustrated in Figure 1 (b). In addition to the data path a $control\ plane$ comprises a number of QOS mechanisms that support the data path's buffering, packet processing and forwarding capability. 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 will briefly describe the operation of predictor, compensator and adaptor. For full details of the framework see [6].

3.1 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 [5]. Previous work on channel prediction either assumes that the state of the channel or the duration of bad link periods are known in advance [5] [3] [12] 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 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 degrading channel conditions may manifest as increased bit errors and loss of signal at the receiver.

In IEEE 802.11, RTS-CTS is used in DCF mode to compensate for the hidden terminal problem, which can lead to a very high numbers of collisions for channels that are heavily loaded. 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 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 PCF mode to verify the state of the channel. In the IEEE 802.11 PCF mode the access point always initiate transmission for both downlink (transmitting the packet) or uplink (polling a mobile device). Therefore, RTS-CTS can be used for both downlink/uplink transmissions.

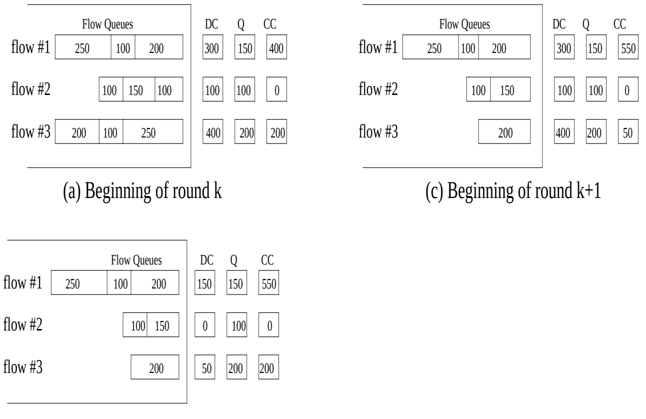
Since the predictor avoids unwarranted multiple retransmissions to a receiver in a bad channel state, the channels throughput is enhanced. In Section 4, we present an evaluation of the predictor mechanism. Channel prediction, however, does not provide any compensation techniques for receivers that have 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 receivers in a bad state, they are not typically re-compensated after the state of the deferred receiver becomes good. Therefore a mechanism to 'credit' and 'compensate' flows/sessions is necessary.

3.2 The Compensator

Our compensator uses a modified version of Deficit Round Robin (DRR) [14] 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) [14]. 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 [14].

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 [14].

We have modified Deficit Round Robin 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 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 that were deferred in 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 [1].



(a) Beginning of round k

(c) Beginning of round k+1

(b) End of round k

Figure 2. *Compensator Operation*

An illustration of the operation of the compensator is shown in Figure 2. 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 $DC_i^{max} = 2 * Q_i$ for each $flow_i$. Figure 2(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, update the compensation counter by the quantum size and reduced the deficit counter by the same amount; (ii) prediction for flow #2 indicates a good channel and 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 and two packets are transmitted and the deficit counter decreased by the packet size. Figure 2(c) illustrates the state of the scheduler at the beginning of next round when Q_i plus αCC_i bytes are added to the deficit counter for each flow i if the compensation counters is positive. Note that only a portion of CC for $flow_1$ and $flow_3$ is added to their deficit counter so that $DC_i \leq DC_i^{max}$.

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

[1] Compensation for the proposed scheme will vary between 0 and the quantum size according to the observed system load.

a large DC^{max} increases the latency bound during periods of heavy 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.

3.3 The Adaptor

The final component of our integrated adaptive-QOS model 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:

•$buffer\ controller$, which operates over the wireless hop; and

•$regulator$, 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 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 the wireless access point.

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 the 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 the 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. Both base and enhanced-QOS require admission control to establish adaptive wireless services.

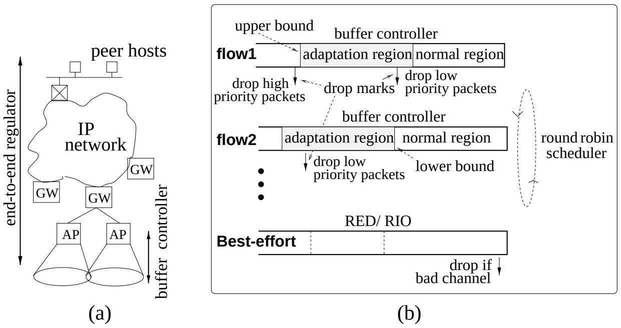


Figure 3. *Adaptor Model*

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 delivering continuous media but lacks any adaptation capability. RTP [13] does, however, incorporate 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 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.

Enhanced-QOS packets are dropped before base-QOS packets in the case of congestion or poor channel conditions persisting. In the Havana Framework, the buffer controller supports this operation by partitioning per mobile device buffers into normal and adaptation regions as illustrated in Figure 3:

• normal region: during normal operations the buffer occupancy is likely to be small when the channel is in a good state. The lowest position of the drop mark is delimited by a 'lower' bound as illustrated in Figure 3.

• 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 dynamically sets '*drop marks*' between a lower and upper bounds 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 3.

Several access point buffering scenarios are illustrated in Figure 3. 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 3 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 in Figure 3 illustrates a flow having two priorities. 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 bandwidth available 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 a relatively high throughput without dropping packets. When the average buffer occupancy is large, the drop mark should be set small. This allows the buffer controller to drop low priority packets saving buffer space for high priority packets in case of severe network conditions.

Lower and upper thresholds bound the position of the drop mark in the buffer as illustrated in Figure 3. Initially, the drop mark can be set arbitrarily between the lower and upper bounds. The buffer controller then adjust the position of the drop marker according to the following rules: (i) drop marks will be increased by *delta* bytes at discrete time intervals, and (ii) drop marks are set to the lower bounds any time the buffer occupancy crosses the upper bound. The time interval for adjusting the drop marks can be a fix time interval or can be set as a function of the round robin scheduler. This interval should be larger than the round interval to avoid processing overhead.

4 Evaluation

In this section we provide an evaluation of the Havana Framework, which is implemented using the NS-2 simulator [10]. We present the evaluation of the system and its components in an incremental fashion. First we analyze the performance of the predictor module in isolation. 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 system works in unison to deliver application and channel dependent QOS in wireless networks.

4.1 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 [10] was modified for the predictor to operate in the PCF mode as discussed 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 4 shows simulation and analytical results for the predictor scheme discussed in Section 3.1 for channel prediction using a packet size 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) [2] whereas the y-axis shows the probability of good channel prediction. The good channel holding times are 10 times longer than the bad channel holding times. Each point in this figure represents an average of approximately 1000 packet transmissions in the simulator. Figure 4 highlights that analytical and simulation results closely follow each other (for analytical results refer to [7]). We have divided Figure 4 into three different regions of interest.

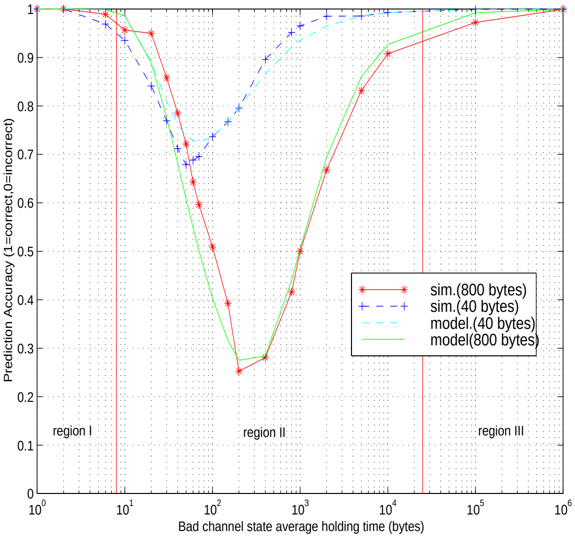


Figure 4. *Predictor Accuracy vs. Packet Size*

• *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

[2] We use bytes as a measure of holding times in Figure 4 because it is easier to compare the holding times of good and bad channel states with the size of the packet.

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 accurate prediction.

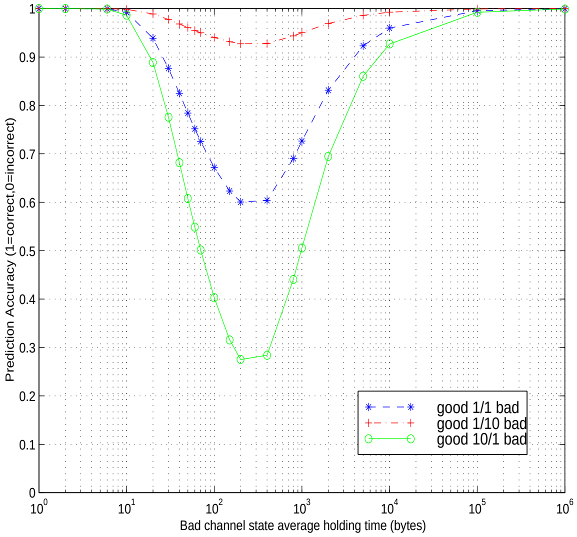


Figure 5. *Predictor Accuracy vs. Good/Bad State Ratio*

Figure 4 shows the evaluation of the predictor performance for two different packet sizes. As indicated 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 5 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 approximately 1000 packet transmissions in the simulator. 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 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 4 and 5 that the accuracy of the predictor drops off as the channel state transitions become similar to the packet transmission time scales. Other channel-mitigation methods (e.g., forward error correction, interleaving, etc.) may, however, improve the accuracy of prediction for fading transitions at the packet transmission time scale.

4.2 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.

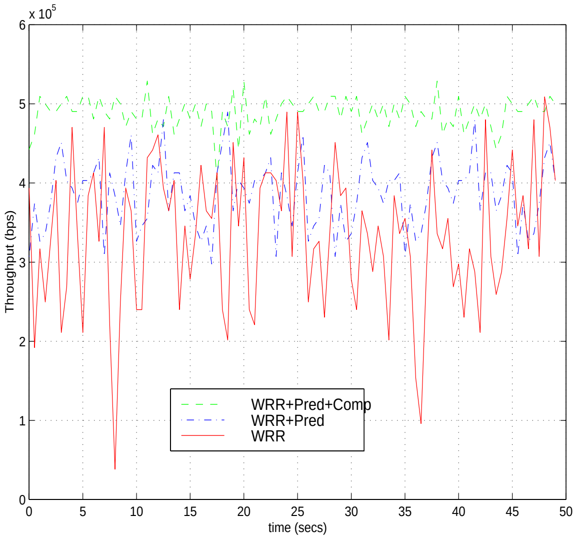


Figure 6. *Throughput Performance using Prediction and Compensation*

Using our NS-2 implementation of 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 and 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 appropriate weights using the in-band signaling protocol [9] to establish the session. The average holding times in good and bad states are set to 20,000 bytes and 8,000 bytes, respectively[3]. 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 percent as shown in Figure 5. If the channel is predicted in error, transmission is deferred and transmission for the next flow/session is carried out as described in Section 3.2. In this example $DC^{max} = 2 * Q$, which results in the maximum amount of compensation bytes given to a flow in one round is twice the quantum size for that flow. Figure 6 shows throughput traces for one of the 500 Kbps flows under three different configuration conditions which include:

1) the scheduler alone;

2) the scheduler with the predictor; and

3) the scheduler, predictor and compensator.

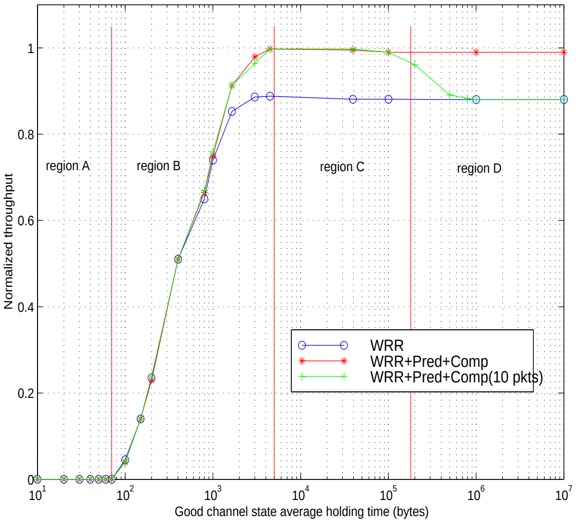


Figure 7. *Throughput Performance*

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 6 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 the mobile devices improves but does not reach the requested bandwidth of 500 Kbps as illustrated in Figure 6. 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 6. Figure 7 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 the experiment. Each point in this figure is the average of 10 runs of the simulation. Three different configurations are considered:

1) the scheduler alone;

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

3) the scheduler with the predictor, 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 7 we divided the graph in 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 poor 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 order 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 region C. As a result, the later two configurations (2 and 3 above) are observed to have similar performance as illustrated in Figure 7.

• *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 7 represents the operational range over which the adaptor can provide enhanced-QOS when applying selective dropping.

4.3. Predictor, Compensator and Adaptor

The final part of our evaluation considers the complete composite system in operation. We simulate the adaptor using a single wireless LAN where three mobile devices receive continuous media service using the scheduler, predictor, compensator, adaptor and arbitrator. Several mobiles 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” MPEG2 video clip which delivers a multi-

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

resolution flow with three video layers of resolution. The base layer (BL) represents the main profile of MPEG2 requiring 300 Kbps as base-QOS and two enhancement layers (E1 and E2) represents two scalable profiles both requiring 100 Kbps each for enhanced-QOS. The size of the buffer configured for the flow is configured to be 8000 bytes and the lower and upper bounds are set to 2000 and 6000 bytes, respectively. Because the flow contains three priorities, the buffer controller drops third priority packets before second priority packets, etc. The drop mark for third priority packets is automatically adjusted every 5 seconds if no buffer overflow is observed during that interval. This adjustment consist of increasing the drop threshold by 400 bytes after each interval. Second priority packets are dropped only if the buffer occupancy increases above 7000 bytes. For this experiment we choose transition rates for the Markov model in the range of region D from Figure 7 where the buffer occupancy is likely to experience overflow conditions.

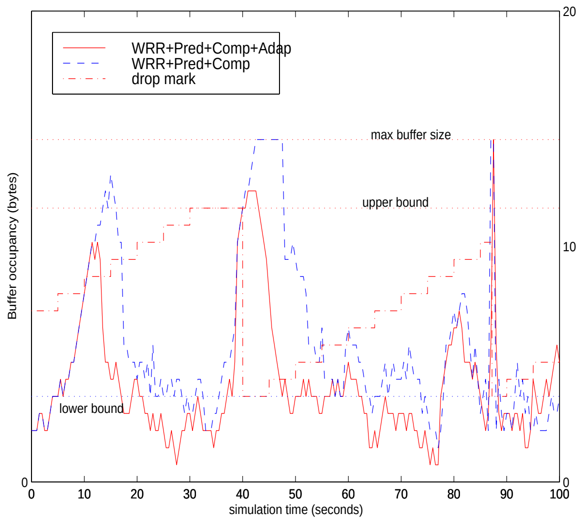


Figure 8. *Adaptor: Buffer Occupancy Behavior*

Figure 8 shows a trace of the buffer occupancy for two configurations:

1) the adaptor configured with the scheduler, predictor and compensator; and

2) the scheduler, predictor and compensator are present but the adaptor is not active.

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 very deep fades. As shown in Figure 8, the position of the drop mark for packets associate with the second enhancement layer increases when the network conditions are good and immediately backs off when overflow conditions are observed at 40 and 87 seconds into the trace. At 40 seconds into the trace the adaptor is able to maintain the minimum quality (e.g., base layer packets) whereas the non-adaptive (i.e., where no adaptor is operational) trace overflows. Finally at 87 seconds into the trace the channel experiences a deep fade that both adaptor and non-adaptor approaches are unable to respond to.

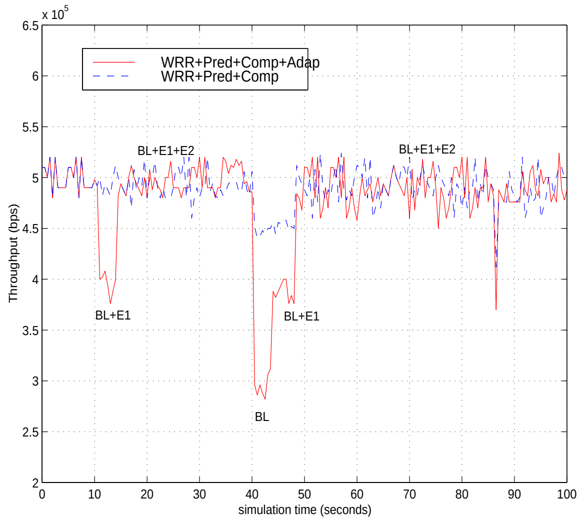


Figure 9. *Adaptor: Throughput Evaluation*

Figure 9 shows the throughput trace captured under the same operational conditions as discussed for Figure 8. In the case where the adaptor is present we show which video layers are delivered to mobile device during the simulation. In addition, we show the throughput in the case were no adaptor is present. Comparing the two traces shown in Figure 9, we observed that the non-adaptor trace achieves a higher throughput compared with adaptor configured trace. Figure 9, however, does not fully convey the behavior of the system because this observation alone does not provide the complete picture. Table 1 shows which percentage of packets were received and played-out at the mobile device for each configuration [4]. As observed from Table 1 the adaptor is able to maintain the minimum quality for a longer duration even when the resulting throughput is smaller than the system where no adaptor is configured. Results from Table 1 indicate that the adaptor achieves better utilization of resources over the wireless hop.

5. Conclusion

In this paper we have discussed the components of the Havana Framework for wireless networks that include a predictor, compensator and adaptor all governed by a central

[4] These results do not include those packets that were received correctly but could not be played out due to semantic issues. For example, enhancement layer 1 (E1) packets received without the corresponding base layer (BL) are useless and dropped.

	BL	BL+E1	BL+E1+E2
without adaptor	94%	94%	94%
with adaptor	99%	96%	89%

Table 1. *Adaptation Performance*

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. They indicate that channel prediction accuracy diminishes quickly as the packet transmission time scales increases and as the channel state transitions approximate the packet transmission time scales. 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 is 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 the most effective approach to delivering application and channel dependent QOS in wireless networks.

6 Acknowledgment

This work is supported in part by the National Science Foundation (NSF) under CAREER Award ANI-9876299 and with support from COMET Group industrial sponsors. In particular we would like to acknowledge the support of Ericsson, Intel, Nortel Networks and IBM for supporting this work.

References

[1] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Enhancing Throughput over Wireless LANs using Channel State Dependent Packet Scheduling. *Proc. of the IEEE INFOCOM*, Kobe, Japan, April 1997.

[2] D. Clark and J. Wroclawski. An Approach to Service Allocation in the Internet. *draft-clark-diff-svc-alloc-00.txt*, 1997, work in progress.

[3] T. S. Eugene, I. Stoica, and H. Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. *Proc. of the IEEE INFOCOM*, San Francisco, California, March 1998.

[4] S. Floyd and V. Jacobson. Randon Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1993.

[5] C. Fragouli, V. Sivaraman, and M. Srivastava. Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling. *Proc. of the IEEE INFOCOM*, San Francisco, California, March 1998.

[6] J. Gomez, A. T. Campbell, and H. Morikawa. A Systems Approach to Prediction, Compensation and Adaptation in Wireless Networks. *In Proc. of ACM First International Workshop on Wireless Mobile Multimedia*, Dallas, Texas, October 1998.

[7] J. Gomez, A. T. Campbell, and H. Morikawa. Supporting Adaptive-QOS over Multiple Time Scales in Wireless Networks. *DIMACS Workshop on Mobile Networks and Computing*, Piscataway, NJ, March 1999.

[8] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications Magazine*, Vol. 1, No.1, First Quarter, 1994.

[9] S. Lee and A. Campbell. INSIGNIA: In-Band Signaling Support for QOS in Mobile Ad Hoc Networks. *Proceedings of Mobile Multimedia Communications, MoMuC*, Berlin, October 1998.

[10] G. Nguyen. Wireless Features in ns Simulator. *http://www.cs.berkeley.edu/ gnguyen/ns/*, 1998.

[11] K. Nichols, V Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. *Internet Draft, draft-nichols-diff-svc-arch-00.txt*, 1997.

[12] V. Bharghavan S. Lu and R. Srikant. Fair Scheduling in Wireless Packet Networks. *In Proceedings of ACM SIGCOMM*, San Francisco, California, 1997.

[13] H. Schulzrinne. RTP: The real-time Transport Protocol. *MCNC 2nd Packet Video Workshop*, Vol. 2, North Carolina, Dec. 1992.

[14] M. Shreedhar and G. Varghese. Efficient Fair Queueing using Deficit Round Robin. *In Proceedings of ACM SIGCOMM*, Berkeley, California, 1995.