A Systems Approach to Prediction, Compensation and Adaptation in Wireless Networks

J. Gomez†, A.T. Campbell†, H. Morikawa‡
†Department of Electrical Engineering and
Center for Telecommunications Research
Columbia University, New York, NY 10027, USA
{javierg,campbell}@comet.columbia.edu
http://comet.columbia.edu/wireless
‡Department of Information and Communication Engineering
The University of Tokyo
mori@mlab.t.u-tokyo.ac.jp

Abstract

This paper presents a framework for provisioning application and channel dependent quality of service in wireless networks. The framework is based on three different adaptation mechanisms that operate over distinct adaptation time scales. At the packet transmission time scale, channel prediction determines whether to transmit a packet or not depending on the state of the wireless channel. At the packet scheduling time scale, a channel state dependent scheduler compensates flows that experience bad link quality while attempting to maintain minimum bandwidth assurances. The packet scheduling scheme is complemented by an application-specific adaptation mechanism that operates over longer time scales and takes into account the ability of wireless applications to adapt to changes in available bandwidth and channel conditions. Unlike packet scheduling, adaptation takes into account application-level semantics and operates over time scales that can be programmed by user.

1 Introduction

A goal of next-generation Internet is to enable mobile users to access and distribute voice, video and data anywhere anytime. As the demand for new mobile services grows, existing (e.g., IEEE 802.11 [15]) and future (e.g., mobile ATM [16]) wireless Internet technology will be required to better support the delivery of multimedia services to mobile terminals with suitable quality. There has been considerable discussion in the research community concerning the best service model for the delivery of mobile multimedia services over wireless networks. One school of thought believes that the radio can be engineered to provide wireline type 'hard' quality of service assurances, e.g., guaranteed delay or constant rate services. The other school argues that the wireless link can not be viewed in this manner because of time-varying environmental factors, e.g., fading. In this case, wireless services lend themselves to more adaptive approaches [9] or better than best-effort type paradigms [14].

We take our lead from the 'adaptive' camp and propose a packet-based *controlled-QOS* framework for application and channel dependent quality of service 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 to adapt to these conditions over multiple time scales. In this paper, we argue that a controlled-QOS service paradigm is suitable for the delivery of voice, video and data to mobile devices.

The controlled-QOS model operates over three distinct time scales found in wireless networks. Different components of the controlled-QOS model are operational at each time scale. These components include *channel prediction*, *compensation* and *adaptation*. Channel prediction allows the scheduler to defer transmission to mobile devices experiencing fading conditions. Channel prediction, however, does not compensate mobile devices that have previously experienced 'outages' due to poor channel conditions. To overcome this problem, we propose *Improved Channel State Dependent Packet Scheduling* (I-CSDPS), based on [2], to deliver enhanced throughput to mobile devices. I-CSDPS attempts to resolve unfairness experienced by different spatially distributed receivers and operates on the packet scheduling time scale. I-CSDPS is complemented by a second adaptation strategy called *active adaptation* that operates over longer time scales and takes into account application-specific adaptation profiles in the case of variations in available bandwidth and channel conditions.

The paper is organized as follows. In Section 2, we present an overview of the controlled-QOS model. In Section 3, we describe our channel predictor followed by a description of the Improved-Channel State Dependent Packet Schedulering scheme in Section 4. In Section 5, we discuss an active adaptation mechanism that supports application-level adaptation. Currently, the controlled-QOS model has been implemented using existing wireless LAN technology (e.g., IEEE 802.11) using the ns simulator [13]. We conclude in Section 6 with some final remarks.

WOWMOM 98 Dallas Texas USA

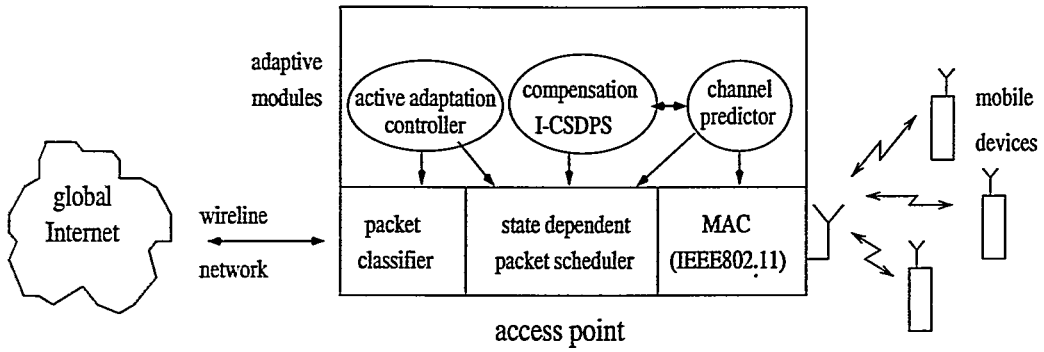

Figure 1: The Controlled-QOSControlled-QOS Model

2 The Controlled-QOS Model

Network dynamics in wireless networks are the result of several different systems interactions operating over multiple time scales. These time scales range from received signal strength variations in the order of microseconds, to available bandwidth variations occurring anywhere between hundred of milliseconds to minutes and hours. The controlled-QOS model attempts to take this time-vary behavior into account by operating over three distinct time scales to respond to changing network conditions found in wireless networks. At each time scale different components of the controlled-QOS model are operational. In Figure 1 we show an illustration of the QOS controlled model. The upper part of the diagram shows three system adaptation modules: channel prediction, improved-channel state dependent packet scheduling and active adaptation. These adaptation modules interact with packet forwarding in different ways and at different times. The controlled-QOS framework assumes a cellular Internet architecture [11] where mobile devices are connected to wireless access points connected to the global Internet. At the packet transmission time scale a channel prediction mechanism probes the wireless channel between the access point and mobile devices to determine the current state of a wireless channel before a packet can be transmitted by the scheduler over the wireless link. The probing mechanism is based on the IEEE 802.11 *request-to-send (RTS)* and *clear-to-send packet (CTS)* pair. If an RTS-CTS probe fails and the channel-state is 'bad', the packet remains in queue in the scheduler buffer for later transmission and the flow-state is 'credited'. If the channel-state is 'good' the packet is transmitted [6].

At the packet scheduling time scales I-CSDPS is operational. Channel State Dependent Packet Scheduling (CSDPS) is a technique that aims at throughput enhancement [2] by monitoring the channel. CSDPS defers scheduled transmission to a receiver in a bad channel state until the fading period is over; thus it can proceed with the transmission of packets to other receivers that are in good channel state. CSDPS does not, however, provide mechanisms to compensate mobile devices that deferred transmission in the past. Within our work we have modified CSDPS to compensate mobile devices experiencing fast and slow fading conditions using a 'deficit' and 'credit' scheme discussed in [12].

The first adaptive component of our framework operates at the lowest time scale after scheduling and prediction. Active adaptation is based on the insight that adaptation is application-specific. There is no 'one adaptation policy fits all' approach to adaptation. For example, audio and video flows may require discrete or smooth adaptation while some real-time data services may be greedy and capable of responding to any available bandwidth [3]. Some applications may be able to tolerate fast time-scale adaptation while others, conversely, may require slow adaptation to available bandwidth conditions rather than instantly reacting to any availability. To support application-specific adaptation we allow the application to interact with an active adaptation controller at the access point to determine if and when the application wants to take advantage of additional bandwidth. Such an active adaptation service is suited to drop semantically less important packets, while responding to changes in the available bandwidth either due to new flows being established at mobile devices or persistent channel degradation that can not be adequately dealt with by I-CSDPS. The semantics of the active adaptive service are as follows. Applications specify their flows as having a minimum bandwidth requirement and a number of enhancement layers. The base layers are treated as higher priority than enhancement layers by the packet scheduler. Applications also specify their adaptation interval over which a stable delivered quality is preferred. The active adaptation controller works in unison with packet scheduling and channel prediction to meet the adaptation needs of applications over wireless networks.

Both priority and delay information are carried in each packet using an in-band wireless signaling protocol called INSIGNIA [10]. By in-band we refer to the fact that control information is carried along with the data as IP options. While the controlled-QOS model has been designed to operate over a variety of radios our implementation is focused on the IEEE 802.11 standard [18] [15] that operates between 1-20 Mbps. The IEEE 802.11 standard operates in two modes: (i) Distributed Coordination function (DCF) where mobile to mobile communications is established using collision sense multiple access with collision avoidance (CSMA/CA), with or without the RTS-CTS option; and (ii) Point Coordination mode (PCF) where an access point provides a centralized controller for contention free communications. IEEE 802.11 is optimized to support best-effort IP delivery using DCF and real time flows using PCF. To support a channel predictor capability based on the RTS-CTS probe we have modified the network simulator (NS-2) IEEE 802.11 code suite [13] to support this new feature in the PCF mode. The access point operates as central scheduler for both up/down link communications.

3 Channel Prediction

Channel compensation is predicated on the assumption that either the state of the channel or the duration of bad link periods are known in advance. In practice, however, the state of wireless links cannot be entirely predicted.

3.1 Operation

In what follows, we discuss our approach to channel prediction. To estimate the channel state, we have implemented a simple hand-shake based on the well known RTS/CTS probing mechanism. RTS-CTS as a channel predictor was proposed in [6], however, no analytical or simulation results about performance of such an approach have been discussed. 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 the CTS packet as an acknowledgment to the RTS. If the CTS packet is received intact the channel state is assumed to be good. If on the other hand the CTS does not arrive after a given timeout then channel state is considered bad. The assumption is that the RTS or CTS could have been corrupted, lost or incorrectly received because degrading channel conditions manifest as increased bit errors and lost signal.

In IEEE 802.11 RTS-CTS is used in DCF operation mode to compensate for the hidden terminal problem which can lead to a very high numbers of collision in the channel for heavy traffic load. However, even if RTS-CTS fails because of channel errors, the transmitting mobile device will always assume the problem was caused by hidden terminals and will back-off before trying again. This assumption does not, however, hold when the system is light-load. In this case the rate of collisions is very small, which makes RTS-CTS in DCF mode effective in estimating the channel state. During PCF operation, the access point is able to acquire the channel before any of its mobile device neighbors in its coverage area. Therefore, there is no need to use RTS-CTS to prevent collisions. Any packet received in error in PCF mode is unambiguously the result of channel condition. The predictor we have implemented works in PCF and light-load DCF modes to verify the state of the channel. In IEEE802.11/PCF mode the access point always initiate transmission for both downlink (transmitting the packet) or uplink (polling a mobile). Therefore, RTS-CTS can be used in both downlink/uplink transmissions. As a means to differentiate between up/down link operations we use RTS-CTS for downlink and request to receive (RTR) and clear to receive (CTR) for uplink.

3.2 Analysis

A two state Markov model is used to model the good and bad states of a wireless channel [19]. Transmission of packets during good state periods assures error free delivery. On the other hand, during a bad period the 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 attached to the packets and only CRC is used [15]. The transitions between states occur 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 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 events for one transmission. Note that uplink analysis is similar using RTR-CTR pair. 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. Let $1/\lambda$ and $1/\gamma$ be the

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

Table 1: Packet Transmission; legend: 0=error-free, 1=error, blank=timeout

average time the channel is in good and bad states, respectively. The transition matrix of the markov model by [19] is as follows:

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

With the steady state probability of the channel being in Bad/Good state given by:

$$\pi_1 = \lambda/(\lambda+\gamma) \quad ; \quad \pi_0 = 1-\pi_1 \quad (2)$$

The probability that the channel prediction is correct (P_C), is equal to the probability that RTS,CTS,DATA and ACK packets are received error-free ($P(pre=0, tra=0)$) plus the probability that predictor (RTS/CTS) and transmission (DATA/ACK) are received in error ($P(pre=1, tra=1)$), see table 1, then:

$$P_C = P(pre=0, tra=0) + P(pre=1, tra=1) \quad (3)$$

If the channel is currently in one of the two states, with κ the transition rate to the other state, the probability that the channel will remain in that state for x more seconds is equal to $e^{-\kappa x}$. Now let rts, cts $data$ and ack be the size in bytes of RTS, CTS, DATA and ACK packets, respectively. Before the transmission of CTS, DATA and ACK packets in 802.11 the transmitter should wait for a short inter frame space (SIFS) respectively [15]. If the speed in bytes/sec of the wireless local area network (WLAN) is C then, the two components can be computed as:

$P_{(pre=0,tra=0)} = P(tra=0|pre=0)P(pre=0)$, where P(pre=0) can be approximated by $\pi_0 e^{-(\frac{rts+cts}{C}+SIFS)\lambda}$, therefore:

$$P_{(pre=0,tra=0)} \approx \pi_0 e^{-(\frac{rts+cts+data+ack}{C}+3SIFS)\lambda} \quad (4)$$

This represent the probability that the channel is good at the beginning of RTS and remains in good state for a period longer than the reception of the corresponding ACK. In this equation we neglected the case in which the channel changes from good to bad and from bad to good state during a SIFS interval. In the same way:

$P_{(pre=1,tra=1)} = \sum_{i=1}^{\infty} P(tra=1|pre_i=1)P(pre_i=1)$

Where the predictor packet (RTS+CTS) can be in error in many different ways. However a good approximate is:

$$P_{(pre=1,tra=1)} \approx \pi_1 e^{-(\frac{rts+cts+data+ack}{C}+3SIFS)\gamma} + \quad (5)$$
$$+(1-e^{-(\frac{rts+cts}{C}+SIFS)\lambda})(1-e^{-(\frac{data+ack}{C}+SIFS)\lambda})$$

This equation has two components, the first one represents the probability that the channel is in bad state at the beginning of RTS and remains bad for a period longer than the full transmission time. The second term represents the probability that the duration of good periods is at least smaller than the duration of prediction and also smaller than the duration of data transmission so both of them are in error.

The RTS-CTS probe introduces a small overhead in the protocol in PCF mode. For mobile devices experiencing continuous fading, the predictor will provide enhanced throughput. In contrast, mobile devices experiencing a continuous good link will receive little benefit from the use of the prediction probe; the downside being the penalty of sending the probe for each packet transmission. Based on the channel prediction the packet scheduler operates under the assumption that the predicted channel state is accurate.

4 Improved Channel State Dependent Packet Scheduling

Since channel prediction can avoid unwarranted multiple retransmissions to a receiver in bad channel state, its throughput is greatly enhanced. Channel prediction, however, does not provide any compensation for the receivers that deferred transmission in the past [2] due to a bad channel state. Although good state receivers can benefit from the deferred transmission of bad state receivers, they are not typically re-compensated after the state of the deferred receiver becomes good. Therefore a compensation scheme is necessary to achieve fairness among flows experiencing different channel conditions [12] [5].

To overcome this potential unfairness problem, we propose I-CSDPS using compensation deficit counters and a combination of CSDPS with a modified version of deficit round-robin (DRR) scheduler [17]. DRR is an implementation of Fair Queuing (FQ) which provides throughput fairness among flows. DRR, however, fails to provide tight packet delay bounds as compared with other (more complex) implementations of fair queuing e.g., weighted fair queueing (WFQ [4]) or a self-clocked fair queueing (*SCFQ* [7]). Because of fading and channel contention delays at the MAC layer, we argue that provision of tight delay bounds in wireless LANs is not feasible, which makes a simpler implementation of fair queueing a suitable choice for this environment. The worst case delay bounds in DRR change when the number of flows change which is opposite in fair queueing. When a few flows are active, which is a reasonable assumption in the pico-cell environment in which IEEE802.11 is targeted to operate, DRR provide worst case delay bounds similar to fair queueing.

A mechanism for compensation to flows in wireless networks is presented in [12]. Flows unable to be transmitted because of channel fading conditions are credited for future transmissions. This proposal, however, has the drawback that a flow coming out of a fading period will be immediately compensated in one round. Even if the maximum amount a flow is compensated is bounded, it can introduce delay in other flows having good link state [12]. These problems are solved in [5] by limiting the portion of bandwidth that 'leading flows' (e.g., flows receiving more bandwidth than the bandwidth requested) provide to 'lagging flows' (e.g., flows receiving less bandwidth than the bandwidth requested because of fading) for compensation. Therefore limiting the worst case delay bound. Our proposal is similar in that we also limit the amount of one-time compensation given. However, we do not tie the amount of compensation given based on 'leading' or 'lagging' bandwidth amounts but on the availability of unused bandwidth in the system, e.g., high/low compensation for high/low unused bandwidth respectively. Since the bandwidth used for compensation does not come from the bandwidth already reserve to flows the QOS bounds can be preserved. Finally since our scheme does not keep track of 'leading' or 'lagging' flows the complexity of the protocol is simplified.

4.1 Deficit Round Robin

Transmission of data packets in DRR is controlled by the use of a quantum size (QS) and a deficit counter (DC) [17]. Quantum size accounts for how many bytes are given to each flow for transmission in each round, whereas the deficit counter keeps track of a 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, the quantum is added to the deficit counter for each flow. The scheduler visits each flow comparing the size of 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 value, a packet will be transmitted and the deficit counter reduced by the packet size. When the packet size is bigger than the deficit counter, the scheduler will keep that deficit value in flow-state table for the next round, and 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.

In the case the quantum size for all flows is the same, an equal allocation of the link is achieved. Making the quantum size for some flows different leads to Weighted Round Robin (WRR), which allows a proportional share of the link according to the weights given to each flow. For example, if three flows have a similar QS (equal to 100), they all will get 1/3 of link bandwidth. If $QS_1 = QS_2 = 100$ but $QS_3 = 200$, the sharing of the link would be $\frac{1}{4}$ $\frac{1}{4}$ and $\frac{1}{2}$ respectively. Normally when the access point admits a new flow, it will set up a specific weight (quantum size) for packet scheduling.

4.2 Operations

We modify weighted round robin to achieve fairness in the presence of location dependent fading conditions by introducing a compensation counter (CC), that is maintained for each receiver. For each round, xCC extra bytes (if compensation counter is positive) are allocated to each flow, where x is a value between 0 and 1. Each time xCC bytes are used to compensate the flow, the compensation counter is decreased by the same amount. It should be noted that if a compensation counter for a receiver is positive, the session will get xCC more bytes for transmission than other sessions with nonpositive compensation counter. This is to compensate receiver sessions which have been deferred in previous rounds. To this end, even if the channel has estimated a bad state and hence the data packet is not transmitted, the deficit counter for the receiver is decreased by the quantum size. In return for the decrease, the compensation counter of the session is provided with a quantum size increase by the same amount[1]. Since the deferred session is compensated by the same amount as the deficit counter is reduced, fairness using I-CSDPS can be obtained.

An illustration of the scheduling state and operations is shown in Figure 2. Part 2(a) shows a snapshot of the scheduler at the beginning of a round. Three flows associated

[1] We propose that the actual compensation vary between 0 and the quantum size according to the observed load of the system.

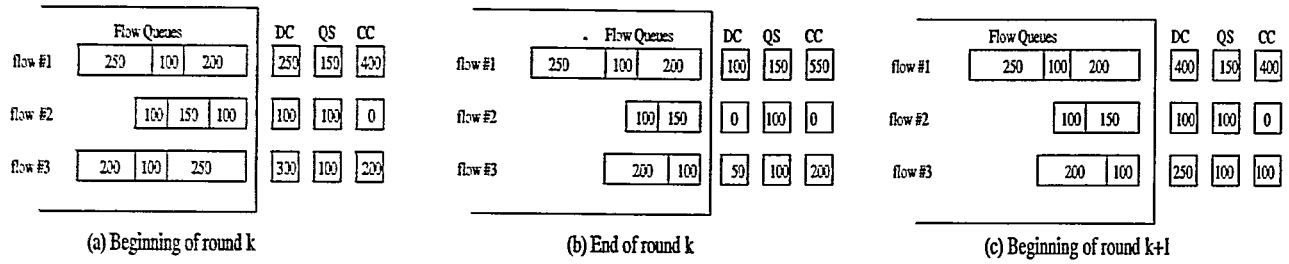

Figure 2: I-CSDPS Operation

with three different mobile devices are active and the sum of the allocated rates are equal to the system capacity, i.e., the system is fully loaded. Part 2(b) illustrates the state of the scheduler at the end of the round. The following events take place during the round are as follows: (i) channel prediction for flow #1 fails and the scheduler defers the transmission of the packet, update the compensation counter by the quantum size and reduced its 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 (normal weighted round robin operation); and (iii) channel prediction for flow #3 indicates a good channel, the packet is transmitted and the deficit counter decreased by the packet size. Part 2(c) illustrates the state of the scheduler at the beginning of next round, when QS bytes plus xCC bytes (if the compensation counter is positive, xCC=QS) are added to the deficit counter.

4.3 Compensation

It is important to clarify that the compensation process realizes two goals: (i) determines how many bytes to credit a flow after the channel predictor diagnoses a bad channel; and (ii) determines which portion of the credit is used for compensation of a flow in each round.

Considering the former goal, it is intuitive to credit by QS every time transmission is deferred. When the system is heavily loaded this is a good solution as we elaborate below. However, when the system is lightly loaded the rate at which the round robin scheduler is serving a flow is faster than the worst case, e.g. under full load. Crediting by QS at this rate will over-credit the flow leading to unfairness for newly arriving flows. Consider, for example, the case when only one flow is active. In this case if RTS-CTS fails a round robin scheduler will serve the flow continuously increasing its compensation counter. We propose to credit flows according to the load of the system with little credit in light loaded systems and a quantum size credit for heavily load systems. In this case, if n flows are registered with the central scheduler (each flow with a QS weight), the load of the system is defined as the ratio of the sum of QS for active flows [2] (QS_i^A) and G, that represent the total capacity of the system in each round. 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 requests a 15 percent share of the link, the quantum size for that flow should be set to 150. Let CC_j^B/CC_j^E be the compensation counter for the flow j at the beginning/end of a round, respectively. Then, if flow j deferred transmission in one round, the compensation counter of the flow will be credited according to:

$$CC_j^E = \begin{cases} CC_j^B + (\frac{\sum_{i=1}^{n} QS_i^A - QS_j}{G - QS_j})QS_j, & if\ G > QS_j \\ 0 & if\ G = QS_j \end{cases} \quad (6)$$

Only when $G = \sum_{i=1}^{n} QS_i^A$, is the system operating at full load and the compensation QS_j. When $\sum_{i=1}^{n} QS_i^A = QS_j$, only $flow_j$ is active with compensation zero.

Now we analyze the second issue of how many bytes of the credit should be used for compensation. It is desirable to compensate a flow that is behind schedule as soon as possible. This means adding CC_i bytes to DC_i in one operation no matter what the size of CC_i is. The problem with this approach is that the latency for the flows is likely to be sensitive to the amount of compensation that is given to a flow in each round. In order to bound the latency it is necessary to bound the maximum compensation that a flow acquires in a single round. We propose to dynamically change the value of x according to the load of the system, fast compensation when the system is lightly loaded and slow compensation for heavy load.

Let $\sum_{i=1}^{n} QS_i^{CC+}$ be the sum of QS only for flows having positive compensation counter (e.g., $QS_i^{CC+} = 0$ if $CC_i = 0$ and $QS_i^{CC+} = QS_i$ if $CC_i > 0$) then the number of bytes available for compensation to $flow_j$ in one round (β), is given by:

$$\beta = max\left[(\frac{QS_j^{CC+}}{\sum_{i=1}^{n} QS_i^{CC+}})(G - \sum_{i=1}^{n} QS_i^A), gQS_j \right] \quad (7)$$

The first term inside the brackets in equation 7 accounts for the compensation in the case when unused bandwidth is available. This can be obtained by computing the available bandwidth and the portion of that bandwidth that corresponds to each flow with a positive CC. The second term, gQS_i, where g is a positive integer, accounts for the minimum compensation given to a flow in one round in case the system is working at heavy load and there is no unused bandwidth available. Because the amount of compensation given to flow j is bounded by CC_j, then:

$$x = \begin{cases} 1 & if \quad \frac{\beta}{CC_j} \geq 1 \\ \frac{\beta}{CC_j} & if \quad \frac{\beta}{CC_j} \leq 1 \end{cases} \quad (8)$$

The choice of g is a design parameter. Choosing a small g will reduce the latency bound but increase the flow's compensation time. On the other hand, choosing a large g increases the latency bound during periods of heavy load but decreases compensation time. Since only a fraction of CC is

[2] We consider an 'active' flow to be one that has at least one packet in the scheduler's queue

used for compensation, CC can become large without affecting the latency bound of the system. Because of this we do not limit the maximum size of the compensation counter.

4.4 Fairness

The fairness properties of DRR are proved in [17]. Since we credit a flow by exactly the same amount of bandwidth the flow missed during fading, the fairness properties are preserved by I-CSDPS. Buffer space is, however, a finite resource. If bad channel periods persist and build up the queue, arriving packets to that mobile access point may find the buffer full and be dropped. For some specific applications, packet dropping can occur even before the buffer is full if the lifetime of the packets has expired. Different applications have different preferences in terms of how long their packets can be queued. If the buffer manager takes a packet timeliness into consideration and drops 'late' packets then of course fairness may not be preserved.

4.5 Delay Analysis

The latency bound provided by normal WRR is given by $\frac{\sum_{i=1}^{n} QS_i}{C}$ [17], where C represents the transmission speed when there are n flows in the scheduler[3]. A small packet arriving at the head of the queue can be delayed by a quantum's size by the other flows in the scheduler. In our case, the quantum size could be bigger than the default size (QS) when compensation bytes are added, therefore the latency bound becomes:

$$LatencyBound = \frac{\sum_{i=1}^{n}(QS_i + xCC_i)}{C} \qquad (9)$$

The value of x is bounded by the condition $xCC_i < CC_i$. It represents which percentage of CC_i will immediately be available for compensation in case the link becomes good with $0 \leq x \leq 1$. This is also translated to how fast flows recover their share of the link. The value of x has a direct impact on the latency bound at which a flow can send RTS-CTS (RTR-CTR for uplink) to test and transmit packets on the channel. It is important to mention that this latency bound does not represent the worst case packet delay, but the worst case channel prediction delay. Since it is out of the scheduler's control how long the channel is in bad state, the best the scheduler can do is to bound the time between channel predictions for each flow.

If the channel is bad and transmission for a packet deferred. Ideally the system should attempt to probe the channel as soon as is possible. Experimental results show [2], however, that fading periods are usually correlated. Therefore, waiting for some time before testing the channel again may be intuitive. On the other hand, waiting too long to test the channel can lead to poor performance. This is because the scheduler can miss periods in which the channel is in a good state and packets could have been transmitted. Determining the optimal interval and time for probing is still an open research issue which depends on how well the duration of bad periods can be accurately estimated.

In this section we have discussed how channel prediction and compensation can maintain the rate in the presence of channel fading conditions. However, when a mobile device experiences persistent fading, it cannot be compensated indefinitely; that is, at some point packets may have to be selectively dropped or the application regulated. In what follows we discuss application-level adaptation techniques which can respond to these conditions over longer adaptation time scales.

[3]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 QS_i should be replaced by the maximum packet size.

5 Active Adaptation

When mobile devices roam between cells, the resources available at each access point may differ. Even within the same cell, session dynamics (i.e. beginning/ending) or mobile devices handing-off also impacts the amount of resources made available to existing mobile devices. These time-varying conditions are visible over longer time scales than the probing of the state of a channel or the servicing of a scheduler with rate compensation. The final component of our controlled-QOS model exploits the ability of applications to adapt to changing bandwidth availability and channel dependent conditions. We call this 'active adaptation' because the application specifies and maintains the adaptation policy that drives these changes. In either case the access point can respond to these conditions by dropping low priority packets and regulate the rate of the flow over a range of applications specific time scales.

In what follows, we discuss how QOS information such as delay, priority and multi-resolution semantics support can be used to enhance the quality of service delivered to mobile devices. For example layered video/audio applications can transmit using different layers of resolution, e.g. MPEG-2 in response to network conditions [1]. Typically, multi-resolution applications transmit a basic layer plus a number of enhancement layers. A bandwidth broker [11] at the access point can be used to manage the allocation of bandwidth to mobile devices based on the services requested using a signaling reservation protocol. The applications can gracefully utilize enhancements layers as bandwidth become available at the bandwidth broker or as channel conditions improve. Conversely, an active adaptation controller [11] can selectively drop enhancement layers while attempting to maintain a 'stable' controlled-QOS by giving preference to the base layers of flows requiring minimum bandwidth assurances.

5.1 INSIGNIA: in-band Reservation.

We utilize an in-band signaling systems called INSIGNIA [10] as a means to respond to the dynamic changes in channel conditions. INSIGNIA carries control information directly in each packet transversing the network using the IP option field. This is similar to the use of the IP type of service or the differential services byte [14] driving packet-level QOS. A control field is set up by the applications and piggybacked in each data packet. This control field includes signaling type (reservation, request), class of service (real time, best effort), precedence field (priority), delay bit and minimum bandwidth. Access points process each individual data packet independently of previous packets. In this way every time flow re-routing occurs, which is the common in cellular networks, the first packet on the new path setups up resources for all other packets without any delay. When a node receives a new INSIGNIA packet carrying a bandwidth request it sets up a new queue in the scheduler with a weight according to the bandwidth request.

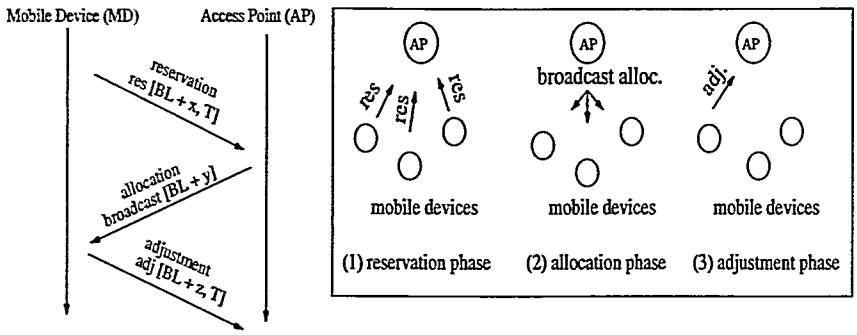

Figure 3: Active Adaptation Protocol

5.2 The Active Adaptation Protocol

While the goal of I-CSDPS is to try to maintain stability of supporting adaptive real-time flows, e.g. minimum bandwidth assurances, fast time-scale dynamics are also resident. Such dynamics translated to application level QOS can lead to poor performance for continuous media type applications. Imagine a video sequence in which the received quality is switching between high and low quality because of bandwidth variations due to new sessions or changing link conditions. Subjective tests suggested that most users are very susceptible to such changes and a stable, even lower, quality is sometimes preferred. This observation that adaptation is application-specific motivates the notion of active adaptation in wireless network where adaptation is paramount. A typical real time application will use a sustained rate service for the basic stream quality (i.e., base layer) and active adaptive services for enhance quality streams (i.e., enhancement layers). Sustained rate services suit applications requiring minimum bandwidth assurances. This is achieved by the scheduler using a weight to assures the requested bandwidth even under loaded conditions.

Applications define a specific adaptation period which specifies the interval over which the applications require 'stable QOS', e.g., consistent quality. Applications are free to define this interval. By increasing the interval applications receive a more stable or assured service. Pricing in relation to the active adaptation service is for future work. The longer the interval the more likely the application would be charged more for the service. Each application selects its adaptation service and enters into a periodic bandwidth negotiation phase with a centralized active adaptation controller at the access point at the beginning of each broadcast interval. The broadcast interval is defined as the interval between broadcast allocations by the active adaptation controller. In the following section we describe each phase of this negotiation (see figure 3). Once the negotiation phase is complete the application is assured a stable bandwidth over the interval specified. Only unexpected channel degradation (e.g. persistent fading) can degrade the mobile device allocated bandwidth and QOS. Three phases characterize the operation of our active adaptation protocol: reservation, allocation and adjustment.

5.2.1 *The Reservation Phase*

Mobiles periodically send reservation (*res*) messages (in DCF mode) to an access point requesting resources for both uplink/downlink communications as illustrated in figure 3. The format of the message contains two fields $[BL_i + x_i, T_i]$. The bandwidth field accounts for the basic layer bandwidth *(BL)* for which resources were already granted (using INSIGNIA) during session setup, plus extra resources, x, to support enhancement layers if possible. Resources for the base layer are granted for the duration of the session unless no traffic activity is detected which releases those resources for new flows. Periodic request of resources for the base layer is necessary to refresh the state of the minimum quality reservation whereas the extra resources account for the maximum quality the applications can use. The interval T is the period over which the applications request stable QOS. The mobile send *res* messages asynchronously to the access point.

5.2.2 *The Allocation Phase*

After a pre-defined interval called the *broadcast interval*, the access point collects all the *res* messages request, computes the allocation for each mobile device for the next broadcast interval, and announce the result in a broadcast message to all mobile devices in the cell. The format of the broadcast message contains the identification of each mobile followed by reservation $[BL_i + y_i]$ granted to flow i for the interval requested, where $y_i \leq x_i$.

5.2.3 *The Adjustment Phase*

The allocation provided by the reservation and allocations phases may not match the needs of a particular application. For example the *res* message may have requested the best possible quality (e.g. bandwidth for base layer + 2 enhancement layers) of a multi-resolution application and the allocated bandwidth may have been less than requested. In this case the application responds by adjusting the allocation down to the amount needed to support a lower but enhanced level of service (e.g. base layer + 1 enhancement layer). In figure 3 the application responds with an adjustment to the allocate bandwidth, e.g., $adj[BL_i + z_i]$ where $z_i \leq y_i \leq x_i$. In order to reduce the number of messages that are sent over the wireless link after the allocation broadcast message, only mobile devices having conflicts with the allocation granted will send a further message to 'adjust' the reservation. By default, if an application does not respond to an *alloc* message it is assumed its allocation was accepted.

5.3 Sharing Extra Bandwidth

A property of WRR is that it shares any available bandwidth fairly among all the flows/sessions in the system proportional to their weights. If there are currently n flows each

of them with a bandwidth reservation BW_i and C is the total capacity, the amount of available bandwidth (ABW) that flow i will obtain according to fairness is given by:

$$ABW_i = (C - \sum_{j=1}^{n} BW_j)(\frac{BW_i}{\sum_{j=1}^{n} BW_j}) \qquad (10)$$

This is a weighted portion of the total available bandwidth. The problem with this sharing allocation approach is that it completely follows the trends of $\sum_{j=1}^{n} BW_j$ in time. In an environment where new session are being created and released, fast variations in the amount of extra resources that flows obtain can be expected. While some applications such as TCP for example are willing to take any available resources in any fashion, others, e.g. video/audio flows, may not wish to take advantage of extra bandwidth unless it is reasonable stable over an application specific adaptation interval. The basic idea to provide a controlled share of available bandwidth to these applications is to filter quick variations by measuring the average available bandwidth and based on that measure, reserve bandwidth for applications over the duration of the application specific adaptation interval.

If an application requests active adaptation with T seconds into the future, where T is at least longer than the broadcast interval, the request will be accepted or denied depending on the available bandwidth measured in previous broadcast interval and the duration of T. The longer the interval (maybe multiples of broadcast interval), the less likely the allocation of available bandwidth will be shared fairly among flows. We assume that some pricing mechanism (that we do not cover in this paper) will charge applications according to the duration of T and the amount of bandwidth assured over that interval.

6 Conclusion

In this paper we have discussed three adaptation components of a controlled-QOS framework; that is, prediction, compensation and adaptation. We argue that a systems approach should be taken to support the delivery of adaptive real-time services over time-varying wireless networks. We believe that prediction, compensation and adaptation need to work in unison to deliver adaptive real-time services and not in isolation. In a companion paper [8], we have shown that our approach has merit and the interaction of these three components over different time scales provides good performance benefits. Our future work will consist of the implementation of the controlled-QOS model over a programmable mobile networking environment [1]. This phase of the work will be the subject of a future publication.

7 Acknowledgement

The authors would like to thank R. R-F. Liao, Andras Valko (Ericsson) and Daby Sow for their comments on this paper. In adition, we would like to acknowledge the support of the COMET Group industrial sponsors.

References

[1] O. Angin, A. T. Campbell, M. E. Kounavis, and R. F.-L. Liao. The Mobiware Toolkit: Programmable Support to Adaptive Mobile Networks. *IEEE Personal Communications Magazine*, August 1998.

[2] 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, April 1997.

[3] G. Bianchi, A. Campbell, and R. Liao. On Utility-Fair Adaptive Services in Wireless Networks. *In Proceedings of International Conference on Quality of Service, IWQoS, Napa, California*, 1998.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. *In Journal of Internetworking Research and Experience, pages 3-26*, 1990.

[5] 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, March 1998.

[6] 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.

[7] S. Golestani. A self-clocked Fair Queueing Scheme for Broadband Applications. *In Proceedings of IEEE INFOCOM*, Toronto, June 1994.

[8] J. Gomez, A. T. Campbell, and H. Morikawa. Supporting Applications and Channel Dependent QOS in Wireless Networks. *Submitted to IEEE INFOCOM'99*, 1998.

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

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

[11] R. F.-L. Liao and A. T. Campbell. On Programmable Universal Mobile Channels in a Cellular Internet. *Proceedings of International Conference of Mobile Computing and Communications, Mobicom, Dallas, October*, 1998.

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

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

[14] 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.

[15] P802.11. IEEE Standard For Wireless LAN Medium Access Control (MAC) and (PHY) Specifications, 802.11. *November*, 1997.

[16] D. Raychaudhuri, L. J. French, R. J. Siracusa, S. K. Biswas, R. Yuan, P. Narasimhan, and C. A. Johnton. WATMnet: A Prototype Wireless Atm System for Multimedia Personal Communications. *IEEE Journal on Selected Areas in Communications, Vol. 15, No. 1, january*, 1997.

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

[18] J. Weinmiller, A. Festag M. Schlager, and A. Wolisz. Performance Study of Access Control in Wireless LANs- IEEE802.11 DFWMAC and ETSI RES 10 HIPERLAN. *Mobile Networks and Applications journal, MONET, Volume 2*, 1997.

[19] M. Zorzi and R. Rao. Error Control Strategies for the Wireless Channel. *IEEE International Conference on Universal Personal Communications, ICUPC, Cambridge, MA*, 1996.