

Supporting Adaptive-QOS over Multiple Time Scales in Wireless Networks

Javier Gomez and Andrew T. Campbell

ABSTRACT. For wireless channels, interference mitigation techniques are typically applied at the packet transmission level. In this paper, we present an adaptive-QOS framework 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 adaptator regulates flows taking into account the ability of wireless applications to adapt to changes in available bandwidth and channel conditions. In this paper, we argue that to effectively support QOS across wireless links there needs to be interworking or integration between the predictor, compensator and adaptator. We achieve this by deploying an arbitrator that coordinates the operation of each mechanism in response to environmental factors, scheduling state and flow semantics.

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 services over wireless networks. One school of thought believes that the radio can be engineered to provide wireline type hard-QOS assurances, (e.g., guaranteed delay or constant rate services). Another school argues that the wireless link cannot be viewed in this manner because of the inherent time-varying environmental factors found in radio communications (e.g., fading). In this case, wireless services lend themselves to more adaptive-QOS approaches [Kat94] or better than best-effort service paradigms [NJZ97].

We take our lead from the adaptive camp and propose a packet-based adaptive-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/sessions to adapt to these conditions over multiple time scales. In this paper, we argue that an adaptive-QOS paradigm is suitable for the delivery of voice, video and data to mobile devices.

The most prominent characteristics associated with wireless networks is the extraordinary premium placed on bandwidth and power efficiency as well as the use of unreliable transmission links. Existing protocols for wireline networks are limited in their ability to deal with these issues; they are generally designed to provide specific services with little ability to adapt to highly time-varying conditions associate with wireless networks. What is required is an appropriate set of adaptive protocols that pass state information across layers in an effort to cope with this variability.

In this paper, we introduce an adaptive-QOS model that is founded on the notion of exchanging state information between mechanisms capable of responding to time-varying wireless characteristics. These mechanisms operate over three distinct time scales and include a *predictor*, *compensator* and *adaptator*. An *arbitrator* monitors the state of each component coordinating their operation in an integrated manner. Channel prediction allows the arbitrator 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, an arbitrator interworks with a compensator (based on channel state dependent packet scheduling [BBKT97]) to deliver enhanced throughput to mobile devices. The compensator attempts to resolve unfairness experienced by different spatially distributed receivers and operates on the packet scheduling time scale. When persistent fading conditions exceeds the operational range of the compensator, the arbitrator activates an adaptator module. The adaptator is designed to operate over longer time scales and takes into account application profiles (e.g., packet priorities within a flow/session) in the case of severe channel conditions or variations in available bandwidth. Ideally an adaptive-QOS model should be used in conjunction with adaptive modulation/coding techniques and other interference mitigation techniques (e.g., smart antennas, multiuser detection, power control) in order to achieve optimum performance and a high degree of adaptive-QOS integration.

In this paper, we present analytical analysis of the predictor, compensator and adaptator modules operating over IP networks supporting IEEE 802.11 last hop wireless LANs. The paper is organized as follows. In Section 2, we present an overview of the adaptive-QOS model. In Section 3, we describe our channel predictor followed by a description of a compensator scheme in Section 4. In Section 5, we discuss an adaptator mechanism that supports application-specific adaptation. The adaptive-QOS model has been implemented using existing wireless LAN technology (e.g., IEEE 802.11) and the ns simulator [Ngu98].

2. Adaptive-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 operating in the order of nanoseconds, to deep fade situations or bandwidth variations occurring anywhere between hundreds of milliseconds to minutes. It is well known that several mechanisms such as modulation, forward error correction, automatic repeat request, interleaving, etc., are useful in dealing with fast radio channel impairments at the packet transmission level. It is unclear, however, which measures are the most appropriate when channel

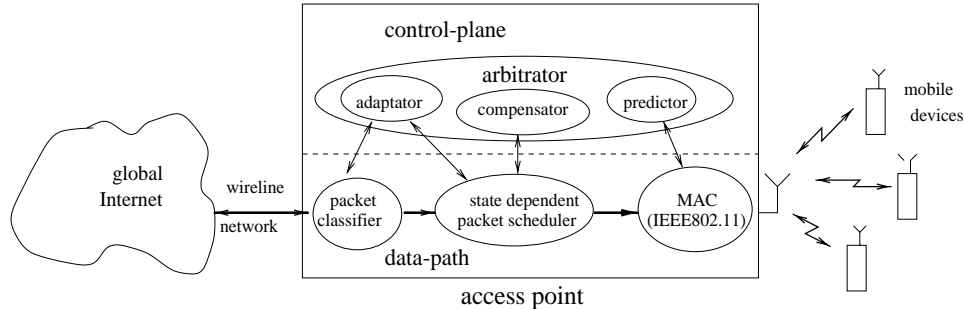


FIGURE 1. The Controlled-QOS Model

impairments become severe and go far beyond the operational range of these mechanisms. The adaptive-QOS model attempts to take this time-varying behavior into account by operating over three distinct time scales to respond to changing network conditions. The main controller of our QOS-adaptive model is an arbitrator present at each wireless access point.

Figure 1 illustrates our adaptive-QOS model. The model comprises a data path, which includes a packet classifier, state dependent packet scheduler and MAC access. In addition to the data path a control plane supports a number of QOS mechanisms that support the data path; these include:

- *an arbitrator*, which coordinates the predictor, compensator and adaptator. 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 agree to transmit the packet or hold it in a buffer and trigger the compensator to ‘credit’ the flow/session. When a flow’s buffer is about to overflow, the arbitrator will set the right filter in the adaptator to drop low priority packets;
- *a predictor*, which 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. The probing mechanism is based on the IEEE 802.11 request-to-send (RTS) and clear-to-send (CTS) packet pair but it could be implemented with any other arbitrary packet pair exchange. If an RTS-CTS probe detects the channel is in a ‘bad’ state, then the packet remains queued in the scheduler for later transmission and the flow-state is ‘credited’. If the channel is detected to be in a ‘good’ state the packet is transmitted [FSS98];
- *a compensator*, which is operational at the packet scheduling time scale. Channel prediction allows the arbitrator to defer 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 a good channel state. Channel prediction does not, however, provide mechanisms to compensate mobile devices that deferred transmission in the past. The compensator is able to ‘credit’ mobile devices experiencing fast and slow fading channel conditions and ‘compensate’ the same flows when the link becomes good. At the same time the compensator keeps packet delay variation bounded and attempts to achieve fairness among all active flows. Our compensator is built around a deficit round robin (DRR) scheduler introduced in [GCM98]; and

- *an adaptor*, which comprises two components: (i) a buffer controller which operates at a slower time scale than prediction and compensation; and (ii) a regulator, which performs end-to-end rate control [Sch92] over longer time scales. Both components are based on the insight that adaptation is application-specific. The buffer controller is suited to drop semantically less important packets while responding to changes in the available bandwidth either due to persistent channel conditions or new flows being established by mobile devices. The adaptor sets appropriate dropping marks in the buffer based on the different priorities within each flow and the long term average measurements of the channel. While the buffer controller tries to maintain ‘good’ quality over short intervals (e.g., when the buffer is about to overflow due to a deep fade), the regulator performs longer-term adaptation that reacts to long-term observed conditions that are experienced in the network.

In our framework, applications specify their flows as having a minimum bandwidth requirement and a number of enhancement layers. The base layers are treated at a higher priority than the enhancement layers by the buffer controller. Both priority and delay information can be carried in each packet using in-band fields such as the differentiated services codepoint (DSCP [KNB97]) or lightweight signaling techniques (INSIGNIA [LC98]).

While the adaptive-QOS model has been designed to operate over a variety of radios our implementation is focused on the IEEE 802.11 standard [WMSW97], [P8097] that operates between 1-20 Mbps. The IEEE 802.11 standard operates in two modes: (i) Distributed Coordination function (DCF) where mobile to mobile communication 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 [Ngu98] to support this new feature in PCF mode. The access point operates as central scheduler for both up/down link communications in this case.

3. 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, the packet remains queued in the scheduler for later transmission and the flow-state is ‘credited’. If the channel is detected to be in a ‘good’ state then the packet is transmitted [FSS98]. Previous work on channel prediction either assumes that the state of the channel or the duration of bad link periods are known in advance [FSS98], [ESZ98], [SLS97]. In practice, however, the state of wireless links cannot be entirely predicted. Our main motivation in this section is to use an analytical framework to investigate bounds and utility of the approach.

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 protocol based on the well known RTS-CTS probing mechanism. RTS-CTS as a channel predictor was proposed in [FSS98], however, no analytical or simulation results concerning the performance of such an approach have been discussed in the

literature. 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 the channel state is considered to be in a bad state. The assumption is that the RTS or CTS could have been corrupted, lost or incorrectly received because of degrading channel conditions manifest as increased bit errors and loss of signal at the receiver.

In IEEE 802.11, RTS-CTS is used in the DCF mode to compensate for the hidden terminal problem, which can lead to a very high numbers of collision in the channel for heavy traffic loads. 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. During the PCF operation, the access point is able to acquire the channel before any of the mobile devices in its coverage area. Therefore, there is no need to use RTS-CTS to prevent collisions. Any packet received in error in the PCF mode is unambiguously the result of channel conditions. The predictor we have implemented works in PCF mode to verify the state of the channel. In IEEE 802.11/PCF mode the access point always initiate transmission for both downlink (transmitting the packet) or uplink (polling a mobile) communications. 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 the downlink and request to receive (RTR) and clear to receive (CTR) for the uplink.

3.2. Analysis. A two state Markov model can be used to model the good and bad states of a wireless channel [ZR96]. 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 802.11 where no Forward Error Correction (FEC) is used [P8097]. The transition 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 the 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 average time the channel is in good and bad states, respectively. The transition matrix of the markov model [ZR96] is as follows:

$$(3.1) \quad P = \begin{pmatrix} P(0|0) & P(1|0) \\ P(0|1) & P(1|1) \end{pmatrix} = \begin{pmatrix} 1 - \lambda & \lambda \\ \gamma & 1 - \gamma \end{pmatrix}$$

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

$$(3.2) \quad \pi_1 = \lambda/(\lambda + \gamma); \quad \pi_0 = 1 - \pi_1$$

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. Legend: 0 : error-free, 1 : error, * : timeout

The probability that the channel prediction is accurate (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:

$$(3.3) \quad P_C = P(pre = 0, tra = 0) + P(pre = 1, tra = 1)$$

If the channel is currently in one of the two states, with κ being 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) [P8097]. If the speed in bytes/sec of the wireless local area network (WLAN) is C then $P_{(pre=0, tra=0)} = P(tra = 0 | pre = 0)P(pre = 0)$, where $P(pre=0)$ can be approximated by $\pi_0 e^{-(\frac{rts+cts}{C} + SIFS)\lambda}$, therefore:

$$(3.4) \quad P_{(pre=0, tra=0)} \approx \pi_0 e^{-(\frac{rts+cts+data+ack}{C} + 3SIFS)\lambda}$$

This represents the probability that the channel is in a good state at the beginning of RTS transmission and remains in a 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 a good approximation for $P_{(pre=1, tra=1)}$ under realistic conditions (e.g., where good channel periods are much longer compared to bad channel periods), is:

$$(3.5) \quad P_{(pre=1, tra=1)} \approx \pi_1 e^{-(\frac{rts+cts}{C} + SIFS)\gamma} + \pi_0 (1 - e^{-(\frac{rts+cts}{C} + SIFS)\lambda})$$

This equation is the sum of two components, the first component represents the probability that the channel is in a bad state at the beginning of the RTS transmission and remains in a bad state for a period at least longer than the beginning of the DATA packet transmission. The second term represents the probability when the channel is in a good state at the beginning of RTS transmission but changes to a bad state before the beginning of the DATA packet transmission.

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 consistently 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 channel prediction the packet scheduler operates under the assumption that the predicted channel state is accurate.

Because channel prediction can avoid unwarranted multiple retransmissions to receivers in a bad channel state, throughput is greatly enhanced. Channel prediction, however, does not provide any compensation mechanism for receivers that have deferred transmission in the past due to a bad channel state [BBKT97]. Although receivers in a good channel state can benefit from the deferred transmission of receivers in a bad channel state, they are not typically re-compensated after the state of the channel of the deferred receiver becomes good.

4. Compensator

To overcome the potential unfairness of mobile devices experiencing different channel conditions, our compensator uses a modified version of deficit round robin (DRR) [SV95] to ‘credit’ and ‘compensate’ flows. Transmission of data packets in DRR is controlled by the use of quantum (Q) and deficit counters (DC) [SV95]. The quantum accounts for the number of allocated bytes to each flow for transmission during 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 queue 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, 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 maintain the deficit value in a flow-state table for the next round and move to serve 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 [SV95].

An equal allocation of the link is achieved when the quantum size for all flows is the same. 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 [SV95]. For example, if three flows have a similar quantum (e.g., equal to 100), they all will receive $1/3$ of link bandwidth. If $Q_1 = Q_2 = 100$ and $Q_3 = 200$, the share of the link would be $\frac{1}{4}$, $\frac{1}{4}$ and $\frac{1}{2}$, respectively.

4.1. Operations. We modify the weighted round robin algorithm 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, αCC_i extra bytes (if the compensation counter for $flow_i$ is positive) are allocated, where α is a value between 0 and 1. Each time αCC bytes are used to compensate the flow, its compensation counter is decreased by the same amount. It should be noted that if a compensation counter for a receiver is positive then the session will get αCC more bytes for transmission in one round than other sessions with nonpositive compensation counters. This compensates receivers sessions which have been deferred in previous rounds. To this end, even if the channel has estimated a bad state (and hence data packets are 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 adjusted by a quantum size increase of the same amount¹. It is important to clarify that the compensation process realizes two goals:

¹The actual compensation may vary between 0 and the quantum size according to the observed load of the system as discussed later.

- it determines how many bytes to credit a flow after the channel predictor diagnoses a bad channel; and
- it 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 Q bytes every time transmission is deferred. When the system is heavily loaded this is a good solution as we discussed below. However, when the system is lightly loaded the rate at which the round robin scheduler serves a flow is faster than the worst case (e.g., under full load). Crediting by Q bytes at this rate will over-credit flows leading to unfairness for newly arriving flows over the long term. Therefore we credit flows according to the load of the system providing less credit in light loaded systems and a full quantum size credit for heavily loaded systems. If n flows are registered with the central scheduler (each $flow_i$ with a weight Q_i), the load of the system is defined as the ratio of the sum of Q_i for active flows ² (Q_i^{Act}) and the total capacity of the system in each round (denoted hereafter as G). 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 gets a 15 percent share of the link, the quantum size for that flow should be set to 150. Let $CC_i(k)$ be the compensation counter of flow i in round k then, if $flow_i$ deferred transmission in round k , the compensation counter in round $k + 1$ will be:

$$(4.1) \quad CC_i(k+1) = \begin{cases} CC_i(k) + \left((\sum_{j=1}^n Q_j^{Act} - Q_i) / (G - Q_i) \right) Q_i & \text{if } G > Q_i \\ CC_i(k) & \text{if } G = Q_i \end{cases}$$

Only when $G = \sum_{i=1}^n Q_i^{Act}$, is the system operating at full load and the compensation given to $flow_i$ is equal to Q_i . When $\sum_{i=1}^n Q_i^{Act} = Q_i$, only $flow_i$ is active with a compensation of zero.

Now we analyze the second goal discussed above; that is, how many bytes of the credit should be used for compensation in one round. It is desirable to compensate a flow that is behind schedule as soon as is possible. This means adding CC bytes to DC in one round no matter what the size of CC is. The problem with this approach is that the latency for flows is likely to be sensitive to the amount of compensation that is given to a particular flow in each round specially during loaded periods. In order to bound the latency it is necessary to bound the maximum compensation that a flow acquires in a single round. Similar to [ESZ98], we bound the maximum amount of bytes $flow_i$ transmit in one round to a constant parameter DC_i^{max} even under loaded conditions.

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

$$(4.2) \quad \alpha CC_i = \min \left[\max \left[\left(\frac{Q_i^+}{\sum_{j=1}^n Q_j^+} \right) (G - \sum_{j=1}^n Q_j^{Act}), DC_i^{max} - Q_i \right], CC_i \right]$$

The first term inside the brackets in Equation 4.2 accounts for compensation in the case where unused bandwidth is available. This can be obtained by computing

²We consider an ‘active’ flow to be one that has at least one packet in the scheduler’s queue.

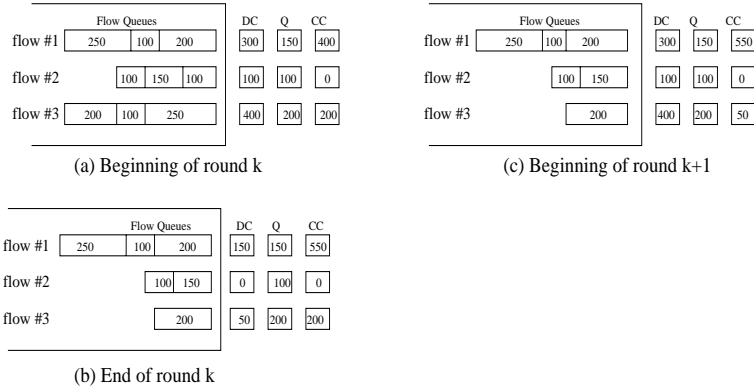


FIGURE 2. Compensator Operation

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

An illustration of the scheduler state and the operation of the compensator is shown in Figure 2. A snapshot of the scheduler at the beginning of a round (after the quantum and compensation bytes have been added) is illustrated in part 2(a). Three flows associated with three different mobile devices are active and the sum of the allocated rates is 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 round. The following events take place during the round:

- the channel prediction for flow #1 detects a bad channel state and the scheduler defers the transmission of the packet, updates the compensation counter by the quantum size and reduces the deficit counter by the same amount;
 - the channel prediction for flow #2 indicates a good channel state and the scheduler transmits the packet reducing the deficit counter by the packet size (normal weighted round robin operation); and
 - the channel prediction for flow #3 indicates a good channel state, allowing two packets to be transmitted and the deficit counter to be 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 (if the compensation counter is positive) are added to the deficit counter for each flow i . Note that only a portion of the compensation counter for $flow_1$ and $flow_3$ is added to their respective deficit counters so $DC_i \leq DC_i^{max}$.

4.2. Fairness. Now we discuss the fairness properties of our compensator mechanism under full load conditions. Using the same nomenclature defined in [SV95], let $DC_i(k)$ and $CC_i(k)$ be the value of the deficit counter and compensation counter, respectively, for $flow_i$ at the end of round k . Let $bytes_i(k)$ be the bytes sent by $flow_i$ in round k , and let $sent_i(k)$ be the sum of the bytes sent by $flow_i$ in rounds 1 through k (where $sent_{i,K} = \sum_{k=1}^K bytes_i(k)$). Based on the

protocol description in Section 4.1 it follows:

$$(4.3) \quad bytes_i(k) + DC_i(k) + CC_i(k) = Q_i + DC_i(k-1) + CC_i(k-1)$$

The fairness property of DRR is analyzed in [SV95]. In order to prove fairness for our compensator we must consider the scenario when the mobile device first defers transmission due to a bad channel state prediction in some rounds and then when the channel is predicted to be in a good state and compensation is provided to the mobile device. Assume that $DC_{i,0} = CC_{i,0} = 0$ are the initial conditions for round 1 letting the predictor diagnose a bad channel state for the next N rounds. The fact that the mobile device defers transmission for rounds $k = 1$ through $k = N$ implies that $sent_{i,N} = 0$. It immediately follows that:

$$(4.4) \quad DC_i(k+1) = DC_i(k) = DC_i^{max} - Q_i \quad ; \quad 1 \leq k \leq N$$

Using this result in Equation 4.3 we get:

$$(4.5) \quad CC_i(k+1) = Q_i + CC_i(k) \quad ; \quad 1 \leq k \leq N$$

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

$$(4.6) \quad CC_i(k-1) - CC_i(k) = (DC_i^{max} - Q_i - DC_i(k-1)); \quad k > N$$

Since $DC_i(k-1) = DC_i^{max} - bytes_i^{pred}(k-1)$, Equation 4.6 can also be written as:

$$(4.7) \quad CC_i(k-1) - CC_i(k) = bytes(pred)_i(k-1) - Q_i \quad ; \quad k > N$$

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

$$(4.8) \quad CC_i(k) = kDC_i^{max} - bytes_i^{pred}(k) - \dots - bytes_i^{pred}(N) \quad ; \quad k > N$$

or:

$$(4.9) \quad CC_i(k) = kDC_i^{max} - sent_i^{pred}(k) \quad ; \quad k > N$$

Clearly, the compensation of $flow_i$ will occur as long as $CC_i(k)$ remains positive and will stop when equal to zero. The ideal bytes allocated to flow i in WRR after k rounds under normal conditions (persistent good channel conditions) is $sent_i(k) = kQ_i$ [SV95]. Subtracting this from Equation 4.9, it follows that as soon as $CC_i(k)$ equals zero, the flow reaches its ideal bandwidth allocation (e.g., the flow has been fully compensated).

The mobile device can only transmit data after round N when the channel is predicted to be in a good channel state. Since $bytes_i^{pred}(k)$ is always smaller than DC_i^{max} , then as long as $DC_i^{max} > Q_i$ the flow will reach its bandwidth allocation. The main implication of this analysis is that even if the mobile device experiences a deep fade, fairness can be reached as long as the channel recovers in the future. Fairness in practical situations, however, does not hold when channel prediction fails and the packet is transmitted and corrupted by channel errors which cannot be anticipated by the predictor. In this case, the accuracy of the predictor plays a critical role in the operation of adaptive-QOS wireless systems.

The choice of DC^{max} is a design parameter. Choosing a small DC^{max} will reduce the latency bound but increase flow compensation time. On the other hand, choosing 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 we do not limit the maximum size of the compensation counter.

4.3. Delay Analysis. The latency bound provided by normal WRR is given by $\sum_{i=1}^n Q_i/C$ [SV95], where C represents the transmission speed when there are n flows in the scheduler³. An interpretation of this equation is that a small packet arriving at the head of the queue can be delayed by the quantum size of the other flows in the scheduler. In our case, the quantum size could be bigger than the default size (Q) when compensation bytes are added; therefore the latency bound becomes:

$$(4.10) \quad LatencyBound = \frac{\sum_{i=1}^n DC_i^{max}}{C}$$

This equation does not consider the delay associated with the RTS-CTS packet exchange. This delay is in the order 3 msec for 802.11 operating at 2 MBPS [P8097]. For small packets this delay can generate a large overhead. The value of DC^{max} , which is also translated to how fast flows recover their share of the link has a direct impact on the latency bound at which a flow can probe the state of the channel. It is important to note that this latency bound does not represent the worst case packet delay but the worst case channel prediction delay. Because it is out of the scheduler's control how long the channel is in a bad state, the best the scheduler can do is to bound the time between channel predictions for each flow.

Ideally the system should attempt to probe the channel as soon as is possible if the channel is in a bad state. Experimental results show [BBKT97], 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 may 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. The fact that WRR visits flows at discrete times (once every round) matches with this 'intuitive' probing timing of the predictor. Determining the optimal interval and time for probing the channel during a fade is still an open research issue which depends on how well the duration of bad periods can be accurately estimated.

The fairness properties of the compensator assumes that buffer space is infinite and packets can remain in the buffer indefinitely. Buffer space is a finite resource, however. If bad channel periods persist and build up the queues, then arriving packets may find the buffer full and be dropped or the application regulated. This observation calls for additional adaptation mechanisms capable of responding to these conditions over longer time scales.

³This equation is valid only when the quantum size is greater than the maximum packet length, which is a necessary condition in DRR to make the system work-conserving. Otherwise Q_i should be replaced by the maximum packet size.

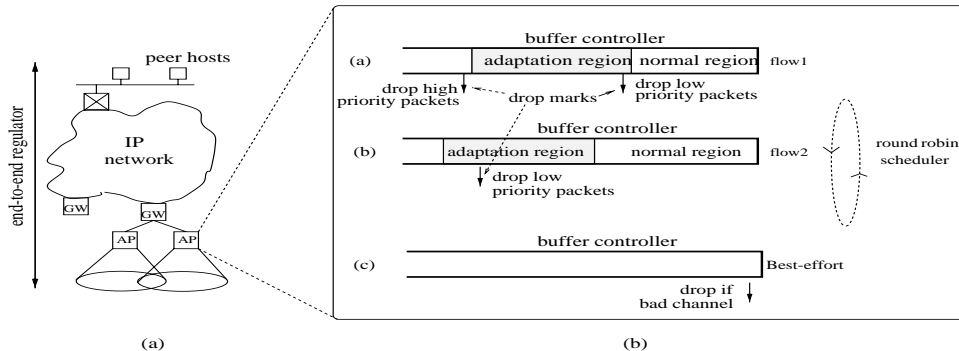


FIGURE 3. Adaptation Mechanism.

5. Adaptator

The final component of our adaptive-QOS model exploits the ability of applications to adapt to channel dependent conditions or variations in available bandwidth over longer time scales. 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 [BCL98]. 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. In either case the wireless access point can respond to these conditions by dropping low priority packets and by controlling the rate of flows over longer time scales.

In what follows, we discuss how QOS information such as delay, priority and multi-resolution semantics can be used to enhance the quality of service delivered to mobile devices. For example, layered audio/video applications can be transmitted using different layers of resolution (e.g., MPEG-2, in response to network conditions [ACKL98]). Typically, multi-resolution applications transmit a base layer plus a number of enhancement layers. These applications are capable of gracefully utilizing enhancements layers as bandwidth become available as channel conditions improve.

5.1. Adaptator Operation. While the goal of the compensator is to maintain stability of supporting adaptive real-time flows (e.g., minimum bandwidth assurances), fast time-scale dynamics are also resident. Such dynamics, which translated to application level QOS, can lead to poor performance for continuous media applications. For example, 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 is undesirable for some applications. Subjective tests suggested that many users are susceptible to such changes and a stable even lower quality is sometimes preferred. The observation that adaptation is application-specific motivates the notion of adaptation in wireless network. The adaptator includes two components that support the notion of adaptive wireless services; these are:

- a buffer controller, which operates over the wireless hop; and
- a regulator, which operates on an end-to-end basis.

A typical real time application will use a regulator to adapt the rate of a source to the average bandwidth observed by the network and use it to support the basic stream quality (e.g., base layer). Enhanced quality streams (e.g., enhancement layers) can be transmitted within the available rate seen from the network, or can be transmitted above that rate with the risk that those packets may be dropped before reaching the destination. Here we assume that if congestion or bad channel conditions occur, then enhancement layers should be dropped first. In our adaptive-QOS framework a buffer controller supports this type of operation by partitioning the mobile device's buffer allocation into two regions using dropping marks as illustrated in Figure 3:

- *a normal region.* During normal operations the buffer occupancy is likely to be small reflecting the fact that the channel is in a good state and no burst of data occur; and

- *an adaptation region.* When severe channel degradation occurs or bursty data arrives, the buffer occupancy can reach high levels where packet dropping will likely occur. When this situation occurs, the compensator notifies the adaptor which set the proper filter in the packet classifier to drop low priority packets.

In our adaptor we assume that another protocol running on an end-to-end basis regulates the rate of flows over a longer time scale according to the measured network performance (e.g., throughput, packet losses/delay). Using end-to-end regulation in this manner limits the likelihood of persistent high occupancy queues due to congestion.

Figure 3 illustrates a per-mobile buffering scenario at a wireless access point. In this example, two real-time flows are supported by per-mobile queues and buffering with all best effort flows being aggregated into a single queue. Figure 3 (a) illustrates the case when a flow consist of three different priorities; this may for example be associated with a video flow with a base layer and two enhancement layers. In Figure 3 (b), a flow with only two priorities is active; this may correspond to an audio flow with normal and enhanced qualities. Finally in Figure 3 (c), a single buffer is used to aggregate best effort traffic for all mobile devices within a wireless cell. The aggregation of several flows into a single queue leads to the head-of-line problem [BBKT97]. In order to avoid this, the adaptor drops the packet at the head of the buffer if the predictor diagnoses a bad channel state.

5.2. Setting the dropping marks. The optimal position of the drop marks illustrated in Figure 3 depends on the average queue size. Without channel prediction, the average queue size depends on several factors that relate to the mismatch between traffic load, link capacity and traffic burstiness. Assuming a regulator operates on an end-to-end basis, the source can match the available rate at the bottleneck node in the network; therefore, small queue sizes should be anticipated (mostly related to jitter in the network). When the predictor is operational, the length of the queue will increase as the length of fade periods increase. If the typical queue size is small then the drop mark should be correspondingly large. This allows the wireless link to operate at a relatively high throughput without having to drop packets. When the average queue size is large, then the drop marks must be correspondingly small. This allows the arbitrator to drop low priority packets earlier, which saves buffer space for high priority packets in case of severe network conditions. Currently, we are working on dynamic techniques for setting up and maintaining these marks.

6. Conclusion

In this paper we have discussed three adaptation components of an adaptive-QOS framework for wireless networks; that is, the predictor, compensator and adaptator mechanisms. We have argued that a systems approach needs to be taken to support the delivery of adaptive services over time-varying wireless networks where multiple time scales come into play. We believe that the predictor, compensator and adaptator mechanisms should work in unison to deliver adaptive services and not in isolation. We argue for a level of integration and interworking managed by an arbitrator that is operational in the access point. Currently we are working on an in depth evaluation of our approach.

References

- [ACKL98] 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.
- [BBKT97] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Enhancing Throughput over Wireless LANs using Channel State Dependent Packet Scheduling. *In Proceedings of the IEEE INFOCOM*, Kobe, April 1997.
- [BCL98] 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.
- [ESZ98] T. S. Eugene, I. Stoica, and H. Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. *In Proceedings of the IEEE INFOCOM*, San Francisco, March 1998.
- [FSS98] C. Fragouli, V. Sivaraman, and M. Srivastava. Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling. *In Proceedings of the IEEE INFOCOM*, San Francisco, California, March 1998.
- [GCM98] J. Gomez, A. T. Campbell, and H. Morikawa. A Systems Approach to Prediction, Compensation and Adaptation in Wireless Networks. *ACM First International Workshop on Wireless Mobile Multimedia (WOWMOM '98)*, Dallas, October 1998.
- [Kat94] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications Magazine*, Vol. 1, No.1, First Quarter, 1994.
- [KNB97] F. Baker K. Nichols, S. Blake and D. Black. Definition of the Differentiated Services Field (DS Field in the ipv4 and ipv6 headers). *RFC 2474*, 1997.
- [LC98] S. Lee and A. Campbell. INSIGNIA: In-Band Signaling Support for QOS in Mobile Ad Hoc Networks. *In Proceedings of Mobile Multimedia Communications, MoMuC*, Belin, October 1998.
- [Ngu98] G. Nguyen. Wireless Features in ns Simulator. <http://www.cs.berkeley.edu/~gn-guyen/ns/>, 1998.
- [NJZ97] 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, work in progress*, 1997.
- [P8097] P802.11. IEEE Standard For Wireless LAN Medium Access Control (MAC) and (PHY) Specifications, 802.11. *IEEE standard*, November 1997.
- [Sch92] H. Schulzrinne. RTP: The real-time Transport Protocol. *MCNC 2nd Packet Video Workshop, vol. 2*, North Carolina, December 1992.
- [SLS97] V. Bharghavan S. Lu and R. Srikant. Fair Scheduling in Wireless Packet Networks. *In Proceedings of ACM SIGCOMM*, San Jose, 1997.
- [SV95] M. Shreedhar and G. Varghese. Efficient Fair Queueing using Deficit Round Robin. *In Proceedings of ACM SIGCOMM*, Berkeley, California, 1995.
- [WMSW97] 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.

SUPPORTING ADAPTIVE-QOS OVER MULTIPLE TIME SCALES IN WIRELESS NETWORKS

- [ZR96] M. Zorzi and R. Rao. Error Control Strategies for the Wireless Channel. *IEEE International Conference on Universal Personal Communications, ICUPC*, Cambridge, Massachusetts, 1996.

DEPARTMENT OF ELECTRICAL ENGINEERING AND CENTER FOR TELECOMMUNICATIONS RESEARCH, COLUMBIA UNIVERSITY, NEW YORK, NY 10027
E-mail address: javierg,campbell@comet.columbia.edu