CHAPTER 2

AN ADAPTIVE CROSS-LAYER PRIORITY SCHEDULING (ACPS) ALGORITHM

Quality of Service (QoS) metrics of a connection (flow or session) include data throughput, packet error / loss rate and delay performance. Providing quality of service, in particular, meeting the data rate and packet delay constraints of real-time data users is one of the major requirements in emerging high speed data networks. This requirement is particularly challenging in networks that include wireless links. Indeed, quality of a wireless channel is typically different for different users and randomly changes in time on both slow and fast time scales. In addition, wireless link capacity is usually a scarce resource that needs to be used efficiently. Therefore, it becomes important to find efficient ways of supporting QoS for real-time data (e.g., live audio / video streams) over wireless channels, supporting as many users as possible with the desired QoS.

2.1 INTRODUCTION

Advances in digital communication system design principles at the physical layer given in Proakis (2001) have played a key role in the evolution of first and second generation wireless networks. Wireless communication systems for the next generation aim to support QoS sensitive services like streaming multimedia and high-speed data for downlink subscribers. However, bandwidth and power constraints and the unpredictable nature of wireless channels render the design of such systems a formidable task. Hence the cross-layer design methodology has attracted much interest in recent years for the design of this kind of wireless communication systems (Shakkotai et al 2003). This approach leverages the synergy existing between different layers of the communication protocol stack to achieve more efficient designs, instead of treating each layer as an individual entity. The QoS and channel-aware packet scheduling is an important illustration of the cross-layer design approach, which exploits interactions between the physical and the upper layers.

Cross-layer design seeks to enhance the capacity of wireless networks significantly through the joint optimization of multiple layers in the network, primarily the physical (PHY) and Medium Access Control (MAC) layers. Although there are advantages of such design in wire line networks as well, this approach is particularly advantageous for wireless networks due to the properties that strongly affect performance and design of higher layer protocols.

Providing guaranteed QoS over wireless channels is challenging. QoS metrics of a connection include data throughput, packet error / loss rate and delay performance.

All networks generally offer two types of services: guaranteed service and best effort service. In guaranteed service, the network provides some sort of service guarantee to individual users or group of users. These guarantees are often in the form of ensuring that the throughput for a group of users is greater than some minimum value or that the delay experienced is smaller than some threshold. In best effort services, the network makes no promises. The first category includes voice, video / audio streaming, video / audio telephony and conferencing; while applications such as web-browsing, email and File Transfer Protocol (FTP) belong to the second category.

The objective of this chapter is to design a packet scheduling algorithm that achieves good user and system throughput while satisfying the varying delay requirements of any wireless systems. This work is implemented by considering a wireless network scenario in NS-2 network simulator.

Since increased system throughput reduces the load (i.e. interference) on the system, it increases channel bit rates and consequently reduces overall delays. A cost function is defined to include the current channel qualities and delay states of the packets in the queue that negotiate between minimizing delay and maximizing throughput. Thus, improving system throughput / channel bit rates can have a positive effect on average normalized packet delay and the total number of missed packet deadlines. However, due to the heterogeneity of packet delay requirements, it is not enough to simply increase system throughput. The scheduling algorithm must adapt its system throughput improvement policies to the varying delay requirements of the packets in queue, so that it can take advantage of high bit rate channels and while doing so does not threaten packet deadlines or degrade average normalized delay.

This chapter is organized as follows. The related existing literature work is reviewed in section 2.2. Different packet schedulers and the corresponding QoS measures are dealt in section 2.3. In section 2.4, a detailed description of the proposed Adaptive Cross-layer Priority Scheduling (ACPS) algorithm is given. The simulation results and discussion are provided in section 2.5. Finally in section 2.6, the summary of the chapter is given.

2.2 RELATED WORK

There is a significant amount of prior work in finding scheduling disciplines that provide delay and fairness guarantees. Generalized Processor

Sharing (GPS) dealt in Parekh and Gallager (1993) is considered as the ideal scheduling discipline that achieves perfect fairness and isolation among competing flows. In terms of fairness and delay guarantees, GPS acts as a benchmark for other scheduling disciplines.

In the long term evolution of packet scheduling algorithms, one can identify two basic approaches in design. Time stamp-based (also called deadline-based) algorithms researched by Bennet and Zhang (1996), Demers et al (1989), Figueira and Pasquale (1995, 1997) and Stiliadis and Varma (1998) have provably good delay and fairness properties but generally need to sort packet deadlines and therefore suffer from logarithmic scale complexity in the number of flows. This sorting bottleneck makes practical implementations of these algorithms problematic and necessitates the design of simpler schemes. Round robin-based algorithms have first order complexity and support fair allocation of bandwidth, but they fail to provide good delay bounds. Thus, while fair queuing is a well-studied problem in modern computer networks, there remains a significant gap between schedulers that have provably good performance and those that are feasible to implement in high-speed routers.

It is to be emphasized here that, reduction in complexity is of paramount importance. If a separate queue is maintained for each flow, the number of queues required is potentially too large. Flow aggregation in the form of Stochastic Fair Queuing can be employed to reduce the number of queues by hashing multiple flows to a single queue.

However, in order to limit the effect of a single misbehaving flow on other flows, the number of queues required is still large. The same is true for many implementations of Differentiated Services. Thus, due to the large number of queues, even logarithmic complexity can be a significant barrier to implementation.

An overview of scheduling techniques for wireless networking can be found in Fattah and Leung (2002), where a number of desirable properties have been summarized and many classes of schedulers have been compared on the basis of these properties. A challenge to scheduler designs is predicting all three aspects of QoS, namely, throughput, loss and delay.

Johnsson and Cos (2001) proposed QoS scheduling of mixed priority non real-time traffic for wireless data systems. They focussed on either minimizing packet delay or maximizing user throughput. In general, satisfying one measure sacrifices the other. There are some algorithms that attempt to satisfy both measures. However, they tend to perform worse for one or both measures compared with algorithms that focus on any one only. This research work proposes a cross-layer priority scheduling algorithm which minimizes a prescribed cost function, given the current channel qualities and delay states of the packets in the queue.

2.3 DIFFERENT PACKET SCHEDULERS AND QOS MEASURES

The packet scheduler is one of the important components used by the routers in the layered architecture of the wireless networks. It determines the order in which packets of various independent flows are forwarded on a shared output link. The commonly used simplest packet scheduling algorithms is First Come First Served (FCFS), in which the order of arrival of packets is also determined along with the order in which they are forwarded over the output link. While almost trivial to implement, FCFS clearly cannot enforce QoS guarantees, as it allows misbehaving packets to capture an arbitrary fraction of the output bandwidth.

2.3.1 Properties of a Packet Scheduler

In general, a packet scheduler should have the following properties.

2.3.1.1 Fairness

The packet scheduler must provide some measure of isolation between multiple flows competing for the same shared output link. In particular, each flow should get its fair share of the available bandwidth and this share should not be affected by the presence and (mis)behavior of other flows. For example, this share may be a pre-allocated amount of bandwidth that should be available to the flow, regardless of other flow activity.

2.3.1.2 Bounded Delay

Interactive applications such as video and audio conferencing require the total delay experienced by a packet in the network to be bounded on an end-to-end basis. The packet scheduler decides the order in which packets are sent on the output link and therefore determines the queuing delay experienced by a packet at each intermediate router in the network.

2.3.1.3 Low Complexity

The time complexity of choosing the next packet to schedule should be small, and in particular, it is desirable that this complexity be a small constant, independent of the number of flows.

2.3.2 Quality of Service (QoS)

Providing high quality of service in a wired as well as in a wireless network should be the very next important task after obtaining functionality.

The term service represents a pre-defined treatment of data during their existence in the network and to guarantee requirements of service where available resources in the network are shared among traffic flows. The quality of this service is a limitation for applications that requires more resources than regular traffic.

A traffic controlling mechanism that filters flows is a scheduler with a scheduling policy to manage resource sharing dependent on traffic type and network specific requirements. A scheduler can give QoS in a wireless network, where flows can be served in an ordered manner and traffic can be handled in flow-specific time constraints to prevent latency and loss.

Five principles that build QoS are given as

Integration principle: This principle states that the service shall be configurable in all IP architectural layers to obtain quality of service in an end-to-end transfer.

Separation principle: This refers to the classification of packets to give preferential and required service.

Transparency principle: The application has to be free from underlying components that are used to obtain QoS. A user application is declaring what type of service is required, not how this shall be achieved.

Asynchronous resource management principle: Functionality of QoS is divided into architectural components consisting of controlling and management modules, which sometime collaborate asynchronous.

Performance principle: Rules and recommendations of traffic create order and structure to the communication protocols for functionality and high performance.

2.3.3 QoS Parameters

The important operating parameters considered in this thesis to measure QoS are Bandwidth, Latency, Jitter and Packet loss rate.

2.3.3.1 Bandwidth

The bandwidth of a channel is defined as the range of frequencies that is passed by a channel. In a scenario where there is fair bandwidth sharing, the quantum of bandwidth given to a flow will vary depending on number of flows sharing the bandwidth. This makes it hard to give a quality of service. To ensure a guarantee of minimum bandwidth reservation for flows and to ensure QoS in a network, it is necessary to have a bandwidth managing scheduler at the resource sharing network node.

2.3.3.2 Latency

Latency value is the time delay a packet experiences on its journey in the network from sending time until it reaches the destination. Arrival of packet needs to be in time and should not vary too much for delay sensitive applications like IP-phone or streamed multimedia data. Increased latency values can reduce the service quality to a large extent. Buffers are used in a network when nodes cannot handle packets immediately. A factor that affects the latency value is buffering at network nodes like routers and bridges as well as at incoming and outgoing queues of stations.

2.3.3.3 Jitter

Jitter is a term related to latency and represents the variation of the time delay between adjacent arriving packets. Buffering at receiving station can control jitter, which is done in most multimedia applications. This buffering is different from previous mentioned latency buffer. Jitter buffering is used for backlogged flow at sender and receiver for appropriate usage of data without arrival. Measurement of jitter values is complicated, since sending and receiving stations must have synchronized clocks.

2.3.3.3 Packet Loss Rate

Packet loss rate is a parameter that describes the reliability for a communication link. Packets can be lost due to several reasons in a wireless network. One reason is congestion at any point of a communication path which creates delay and drop if there are time constraint packets. Other reasons such as electromagnetic interference, noise or location based errors in a communication link, also will cause drop of packets if there are bit errors in received data.

2.4 A TYPICAL WIRELESS SCHEDULER

In order to meet the wide-ranging QoS requirements of various applications, traffic scheduling algorithms should be employed. Such schedulers operate across different sessions (connections or flows) to ensure reserved throughputs and bounds on delays and loss rates. As illustrated in Figure 2.1, the function of a scheduling algorithm is to select the session whose head-of-line packet is to be transmitted next. This selection process is based on the QoS requirements of each session. Each mobile station can support one or more sessions at any given time.

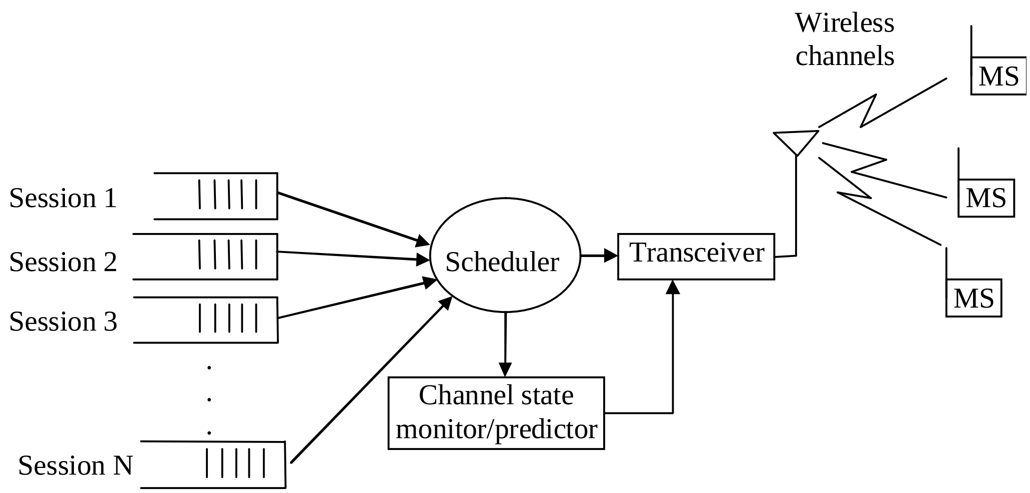

Figure 2.1 A Typical Wireless Scheduler

In order to design a scheduler that performs well with respect to both delay and throughput under a variety of system conditions, it is required to define a cost function that facilitates negotiation between minimizing delay and maximizing throughput. Scheduling the packets with the highest bit rate results in excellent throughput and reduced packet transfer times. Unfortunately, packet delay is composed of both transfer time and queuing time. Thus, in order to perform well with respect to both delay and throughput, the scheduler must try to maximize the use of good channel bit rates while adhering to packet delay requirements.

To this end, a cost function that includes the cost of delaying packets as well as the gain from maximizing throughput is defined. Then, an algorithm that schedules the packet which minimizes the cost function at every scheduling event is designed. This may be the packet with the earliest deadline or the smallest delay requirement or the highest channel bit rate. The choice depends on the mix of delay requirements, and channel bit rates of packets in queue at the time of scheduling.

2.4.1 Cost Function

The cost function J (Johnsson and Cox 2005) operates on the set V of all possible permutations of the packets in the queue. A permutation (scheduling order), $v \in V$, of the packets in the queue is a 1-to-1 mapping of the set of queued packets onto itself. If M is the total number of packets in the queue, there are M! possible permutations. Thus the cost function J is defined as follows:

$$J(v) = \sum_{i=1}^{M} \frac{\hat{d}_i}{r_i} \tag{2.1}$$

where $$\hat{d}_i = c_i + \sum_{j=1}^{i} \frac{b_j}{e_j} \tag{2.2}$$

where b_j and e_j are the packet js total bits and estimated channel bit rate, respectively. The quantity $\hat{d}_i$ is the delay estimate and r_i is the delay requirement of the i^{th} packet in the scheduling order v. The delay estimate $\hat{d}_i$ is the sum of packet i^{th} current delay c_i, and the total estimated transmit time of all packets before and including it in the scheduling order v.

The delay requirement $r(c,l)$ is a function of the packet delay class c, and packet size l in bytes and the assumed values are given in table 2.1(Johnsson et al 2001). It gives the delay requirement in terms of different priority classes such as P1, P2 and P3 for packet sizes of 128 and 1024 bytes. Some interpolation functions would be needed to determine delay requirements for packet sizes other than that given in the table.

In such cases, the delay requirement of each packet can be obtained from

$$r(c,l)=\left\{\begin{array}{ll} \dfrac{0.5l}{128}, & c=1, l \le 128 \\ 0.5+\dfrac{1.5(l-128)}{1024-128}, & c = 1, l > 128 \\ \dfrac{5l}{128}, & c=2, l \le 128 \\ 5+\dfrac{10(l-128)}{1024-128}, & c = 2, l > 128 \\ \dfrac{50l}{128}, & c=3, l \le 128 \\ 50+\dfrac{25(l-128)}{1024-128}, & c = 3, l > 128 \end{array}\right\} \tag{2.3}$$

Table 2.1 Delay Requirements for Different Priority Classes and Packet Sizes (*After Johnsson and Cox 2001*)

Priority Class, *c*	**Delay Requirement, *r* (ms)**	
	Packet Size = 128 bytes	**Packet Size = 1024 bytes**
P1	0.5	2
P2	5	15
P3	50	75

The packet class *c* in the delay requirement is based on the priority levels of the packet denoted as P1, P2 and P3, in which P1 being the highest priority in the order.

The estimated transmit time of the packet in the j^{th} position is given by b_j / e_j where b_j and e_j are the packet $i's$ total bits and estimated channel bit rate, respectively. It can be seen that the cost function, *J* is effectively a weighted estimate of the Average Packet Delay of packets in the queue when they are scheduled according to permutation *v*. Thus, minimizing

J reduces the packet delay of the packets in queue given their current estimated channel bit rates.

Since packet delay is a function of channel bit rates, the cost function reflects the impact of varying channel bit rates on the delay cost of a given scheduling permutation. In other words, the cost function indicates when scheduling high bit rate packets before more delay sensitive packets benefits or hurts delay performance. Thus, it weighs the cost of delaying urgent packets against the gains from scheduling packets with high channel bit rate.

2.4.2 ACPS Algorithm

A new packet scheduling algorithm called the Adaptive Cross-layer Priority Scheduling (ACPS) is proposed and designed to minimize the cost function in Equation (2.1). The name derives from the fact that this scheduler adapts to changes in variables across two system layers: packet delay on the link layer and channel bit rates on the physical layer. The ACPS algorithm schedules the first packet of the scheduling permutation that minimizes the cost function for the current queue conditions. Since the cost function is a weighted estimate of the average normalized packet delay of packets in the queue, this algorithm attempts to minimize the average packet delay measure at every scheduling opportunity. It also increases user throughput by taking advantage of scheduling high bit rate packets before the transmission of more urgent packets. The proposed ACPS algorithm considers the packet queuing conditions also into account while trying to maximize throughput within the confines of improving delay performance. In other words, the ACPS algorithm only tries to maximize throughput within the confines of improving delay performance. As a result, the ACPS algorithm adapts to changing channel bit rates and negotiates between minimizing delay and maximizing throughput (which reduces overall delays) at each scheduling event.

Depending on the measurement and processing capabilities of the system, the ACPS algorithm may use instantaneous or average channel bit rate estimates in the cost function. In theory, the ACPS algorithm compares the total estimated delay cost J, of all possible scheduling permutations of packets in the queue and then queues the packets according to the scheduling permutation that minimizes J. At each scheduling event, the algorithm simply schedules the first packet in the queue until the queue changes significantly.

However, scheduling permutations and their cost estimates are only valid, given the current channel and queue conditions. Due to user mobility, traffic burstiness, and the flux of users entering and leaving the system, channel bit rate estimates and the set of possible packet permutations may be invalid by the next transmission opportunity. Consequently, if a new packet enters the queue, an existing packet leaves the queue due to dropping or a queued packet's channel bit rate estimate changes prior to the next transmission opportunity. The ACPS algorithm must again determine the minimum cost scheduling permutation. Thus, in bursty traffic conditions and / or high mobility environments when channel and queue conditions change rapidly, determining the entire scheduling order at any one time is a waste of processing power since the minimum cost permutation may need to be re-determined many times before the end of the queue is reached.

Under these system conditions, it makes sense to determine only the next packet to send for each transmission opportunity, since determining the full scheduling order expends processing power on information that cannot be used.

Since the cost function is linear, the relative scheduling order of any adjacent packets in the queue can be determined without knowing the entire order. For example, consider two packets m and n. Assume that packet *m*'s original position in the queue comes before that of packet *n*. To determine which of these two packets is to be scheduled first, there is a need to compare the cost function of the original queue order with that of the queue after the packets trade positions. This can be expressed as: if $J(v_{m,n}) < J(v_{n,m})$, then schedule *m* before *n*. In this expression, $v_{m,n}$ represents the permutation that schedules packet *m* before *n*, and $v_{n,m}$ vice versa. All other packets retain their original queue positions. Due to the cost function's linearity, the inequality $J(v_{m,n}) < J(v_{n,m})$ can be simplified into $(b_n / e_n\, r_m) < (b_m / e_m\, r_n)$, where $b_m\,(b_n)$, $e_m\,(e_n)$ and $r_m\,(r_n)$ denote total bit size, estimated channel bit rate, and delay requirement of m^{th} (n^{th}) packet respectively. Thus, instead of calculating the cost of two entire scheduling orders, it is sufficient to calculate two simple ratios only.

As a result, the first packet in the minimum cost scheduling order can be determined by performing one-by-one cost comparisons of the packets in the queue. In other words, the packet to send without determining the entire scheduling order can be determined. The search is stopped after determining the packet to send, since the rest of the scheduling permutation may change by the next transmission opportunity. This significantly reduces the required processing time of the algorithm in rapidly changing traffic and channel conditions.

2.4.3 Performance Measures

The most common performance measures for real-time data services are packet delay and user throughput. The same performance measuring parameters are considered in this chapter to evaluate the proposed ACPS algorithm.

2.4.3.1 Average Packet Delay

Packet delay is often normalized with respect to the packet's delay requirement. The general formula used for Average Packet Delay is given by

$$\text{Average Packet Delay} = \frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{r_i} \qquad (2.4)$$

where d_i and r_i are packet $i's$ total packet delay and delay requirement respectively, and N is the total number of packets transmitted. The average packet delay value is expected to be as small as possible.

2.4.3.2 Average User Throughput

User throughput is measured as the average normalized packet bit rate. However, this measure does not accurately reflect the throughput a user experiences, since it does not consider overlaps in the user's packet queuing times.

In order to reflect the user's true throughput service experience, a new performance measure is defined and it is given by

$$\text{Average User Throughput} = \frac{1}{K}\sum_{n=1}^{K}\frac{P_n}{T_n} \qquad (2.5)$$

where P_n and T_n are the total number of successful packet transmissions and the total time on the system for user n respectively and K is the total number of users.

2.5 SIMULATION RESULTS AND DISCUSSION

A scheduling algorithm for data services that feature prominently in the literature is Weighted Fair Queuing (WFQ). This algorithm separates packets into queues according to their delay class. These queues are then served in weighted round robin fashion. The weights are based on the relative performance requirements among delay classes. While serving a queue, the packets within the queue are serviced in round robin fashion as well. The performance of the proposed ACPS algorithm is studied in comparison with the WFQ algorithm.

2.5.1 Simulation Environment

The Wireless Local Area Network (WLAN) environment under IEEE 802.11 b standard is simulated using NS-2 network simulator with the following features assumed in Johnsson and Cox (2005). A radio network of grid size 500m x 500m with each three numbers of sources and destinations assumed with a constant mobility speed of 3km/hr are considered for simulation. The number of data files a user receives during a session is geometrically distributed with mean 10. The data files can either be e-mail or World Wide Web (WWW) files. Since users tolerate greater delays when downloading e-mail files compared with that of WWW pages or Constant Bit Rate (CBR) traffic files, e-mail files are assigned a priority of class 2 (P2), while WWW / CBR files are assigned a higher priority of class 1 (P1). File Transfer protocol (FTP) files are assigned with class P3, the lowest priority. The distribution of e-mail file sizes is approximated by a clipped Cauchy with mean 4 kB. WWW file sizes are log-normally distributed with mean 4.1 kB and standard deviation 44 kB. FTP files are assumed to be exponentially distributed. Data file inter arrival times are Pareto distributed with mean 10 s.

The time step of the system simulator is 20 ms and is equal to the length of one Radio Link Control (RLC) block in GSM. At each time step, the SINRs of queued and active users are calculated based on a Two-ray ground propagation model with exponential path-loss, log-normally distributed large-scale (shadow) fading, and Rayleigh distributed small-scale fading. Power control is not considered in our analysis. Thus, all transmissions are assigned the same signal power, which is set high enough (20 W) to guarantee coverage throughout the cell with maximum queue size of 50. Therefore, the limiting factor on signal quality is interference, not distance, from the base station. The AODV routing protocol and the maximum of two retransmissions with packet loss in the range of [10^{-4}, 10^{-1}] are considered for simulation purposes.

Using the simulation environment and the performance measures outlined above, the performance of the WFQ and the proposed ACPS algorithms are compared with respect to average packet delay and average user throughput for P1, P2 and P3 class files.

2.5.2 Throughput Performance of ACPS Algorithm

Since the normal sizes of data packets used for simulation in Global System for Mobile Communication (GSM) are 64 (small), 128 (medium) and 192 (large) size bytes, the same number of bytes are considered in this simulation also. From Figures 2.2 to 2.4, it is understood that class P1 achieves better average user throughput compared with other priority classes P2 and P3 for a given sized packets. Since P1 holding the highest priority, always the user carrying P1 class files for transmission are provided the channel with good condition (high SNR). So P1 class files with different packet sizes are having more opportunity for transmission compared with P2 and P3 class files. It is observed that any priority class file with smaller packet sizes shows better performance compared with larger size packets. Since the large sized packets experience more delay, their maximum admissible life time in the network gets exhausted, leading to more loss of packets during transmission. The packet loss can be recovered by retransmission. So the average user throughput degrades significantly for large sized packets compared with the others.

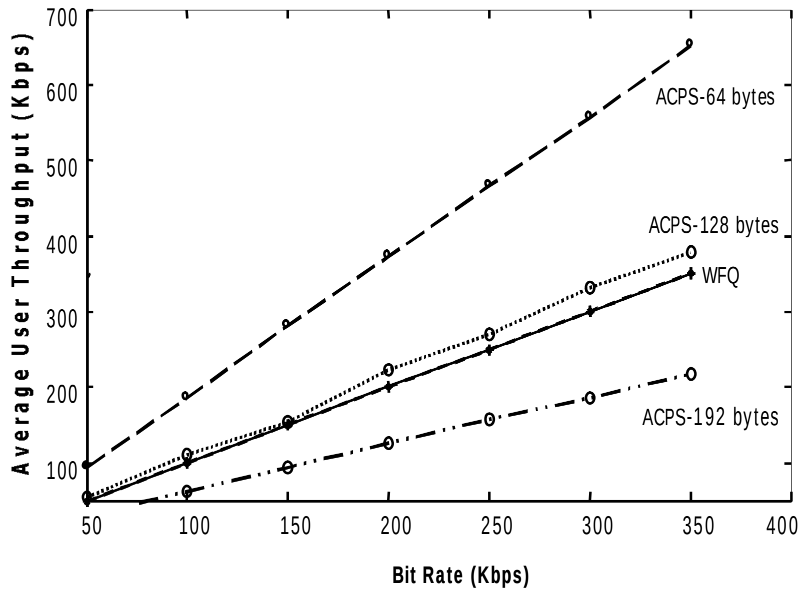

Figure 2.2 Average User Throughput Performance of the ACPS and WFQ Algorithm for P1 Priority Class File with 64, 128 and 192 Bytes Sized Packets

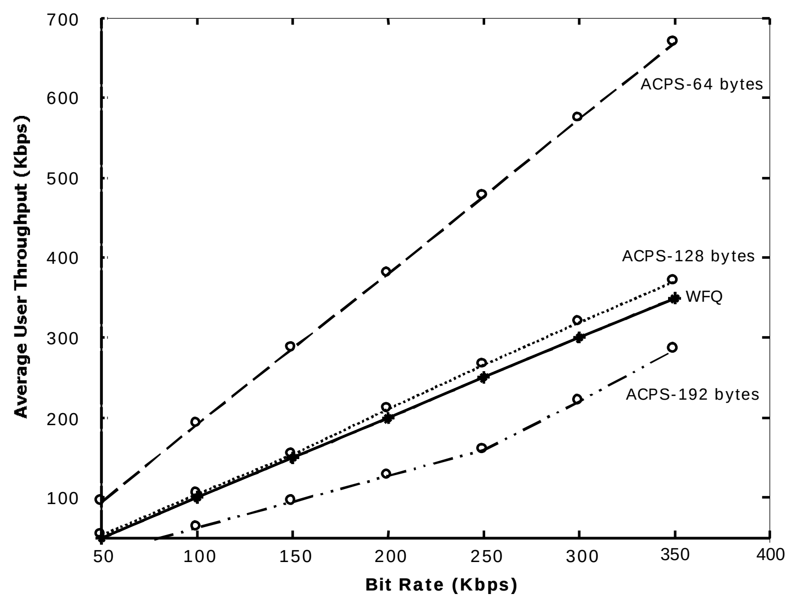

Figure 2.3 Average User Throughput Performance of the ACPS and WFQ Algorithm for P2 Priority Class File with 64, 128 and 192 Bytes Sized Packets

Since WFQ algorithm uses round-robin fashion of scheduling, it achieves a constant average user througput irrespective of packet sizes. By comparing ACPS algorithm with WFQ algorithm, it is shown that ACPS outperforms WFQ when 64 bytes sized packets are used with any priority class file. When the packet size is 128 bytes, the average throughput performance improvement in ACPS algorithm is very minimal. For 192 bytes packet sizes, WFQ algorithm is showing better average user throughput performance than that of ACPS algorithm. It is because of the larger delay experienced, completion of maximum admissible lifetime in the network and more loss of packets in ACPS algorithm.

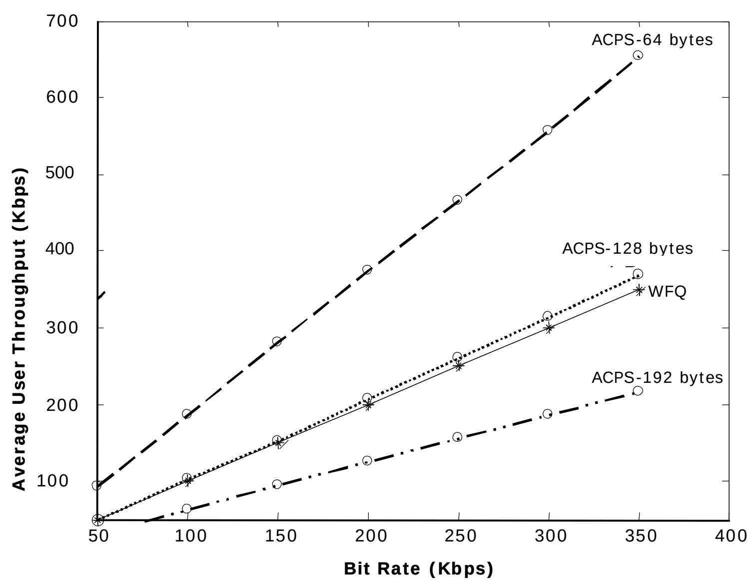

Figure 2.4 Average User Throughput Performance of the ACPS and WFQ Algorithm for P3 Priority Class File with 64, 128 and 192 Bytes Sized Packets

2.5.3 Delay Performance of ACPS Algorithm

Table 2.2 shows the average packet delay performance of the proposed ACPS algorithm for P1, P2 and P3 classes of packets sizes with 64, 128 and 192 bytes. Since P1 is holding the highest priority, it achieves very minimal delay due to negligible queuing delays. The packets of classes P2 and P3 files having the next priority levels respectively and hence their delay performance is poorer than that of P1 file packets for all packet sizes.

When the performance of ACPS and WFQ algorithms are compared, our proposed ACPS algorithm always shows better delay performance independent of packet size as well as priority classes. For example, simulation results show that the average packet delay time taken by P3 class files with 64 bytes packets using ACPS algorithm (950 μs) is nearly 21 times lower than that of the average packet delay time taken by 64 bytes packets using WFQ algorithm, (20 ms). Similarly delay of P1 file with 64 bytes packets is 21μs, which is nearly 950 times lower than that of the delay experienced by WFQ algorithm, i.e., 20 ms. Thus the outperformance of the proposed ACPS algorithm for all three priority classes than WFQ algorithm is verified.

Table 2.2 Delay Performance of ACPS and WFQ Algorithms

Packet size in bytes	Delay			
	ACPS (μs)			WFQ (ms)
	P1	P2	P3	
64	21	110	950	20
128	30	120	980	100
192	95	745	775	100

2.6 SUMMARY

In this section, we developed a new packet scheduling algorithm namely Adaptive Cross-layer Priority Scheduling (ACPS) algorithm, which minimizes a prescribed cost function, given the current channel qualities and delay states of the packets in the queue. Simulations using NS2 simulator reveal that our developed ACPS algorithm outperforms existing WFQ scheduling algorithm both in terms of packet delay and user throughput. The better performance obtained are due to the characteristics of ACPS algorithm in terms of packet delay deadlines on the link layer and channel qualities on the physical layer. In particular, cost function value of the proposed algorithm increases when packets experience decrease in channel bit rates. Consequently, since the ACPS algorithm schedules to minimize the total delay cost, it decreases packet channel bit rates and increases the probability of their being moved later in the scheduling order. This further increases probability of scheduling packets with good channel bit rates, which results in excellent user throughput. At the same time, the ACPS cost function is based on packet delay requirements and bit sizes, as well as channel bit rates. As a result, although channel bit rates influence the scheduling order, their impact depends on the difference in packet sizes and delay requirements of packets in the queue.

Ultimately, the ACPS algorithm schedules to minimize the actual delay cost, no matter which variables carry the greatest influence. Hence, it achieves excellent delay performance and good throughput performance.