ABSTRACT

ENHANCING END-TO-END QUALITY OF SERVICE PROVISIONING IN WIRELESS AD HOC NETWORKS USING SERVICE VECTORS

by
Didem Gozupek

A cross-layer architecture that achieves significant power savings, while enhancing the end-to-end QoS provisioning and granularity in wireless ad hoc networks is proposed in this thesis. Recently, a new concept called *service vector* has been introduced, which enables an end host to choose different service classes along its data path. This scheme enhances the user benefits from the network services and network resource utilization, while maintaining the simplicity and scalability of the current Differentiated Services (DiffServ) network architecture. This thesis explores the application of this concept on wireless ad hoc networks and provides a cross-layer architecture based on the combination of delay-bounded wireless link level scheduling and the network layer service vector concept, which enables a wireless ad hoc network to achieve significant power savings and finer end-to-end QoS granularity. The impact of various traffic arrival distributions and flows with different QoS requirements on the performance of this cross-layer architecture is also investigated and evaluated.

ENHANCING END-TO-END QUALITY OF SERVICE PROVISIONING IN WIRELESS AD HOC NETWORKS USING SERVICE VECTORS

by
Didem Gozupek

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

August 2005

APPROVAL PAGE

ENHANCING END-TO-END QUALITY OF SERVICE PROVISIONING IN WIRELESS AD HOC NETWORKS USING SERVICE VECTORS

Didem Gozupek

Dr. Symeon Papavassiliou, Thesis Advisor Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Nirwan Ansari, Committee Member Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Jie Yang, Committee Member Date
Network Engineer, Spirent Communications

BIOGRAPHICAL SKETCH

Author: Didem Gozupek

Degree: Master of Science

Date: August 2005

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2005

- Bachelor of Science in Telecommunications Engineering,
  Sabanci University, Istanbul, Turkey, 2004

Major: Electrical Engineering

To my beloved family
for their unconditional love and support

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Symeon Papavassiliou and Dr. Jie Yang for their valuable insights and suggestions as well as continuous support throughout the thesis research.

I also would like to thank Prof. Nirwan Ansari for participating in my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

The proliferation of Internet applications and services has fostered a growing research interest in provisioning Quality of Service (QoS), which refers to the capability by which a network provides certain performance guarantees. Evolution of the Internet infrastructure from a best-effort service model to the one in which service differentiation can be provided for different users and applications, and progression of the networking environment towards wireless domain, lead to a growing need for end-to-end QoS provisioning techniques in wireless networks that provide differentiated services. At the same time ad hoc wireless networks, which consist of wireless nodes interconnected by multihop communication paths, introduce additional challenges since they have no fixed network infrastructure or administrative support.

Currently, there are two service models proposed for end-to-end QoS provisioning in the Internet: Intserv [1] and Diffserv [2]. The former provides per-flow based resource reservation and allocation, whereas the latter aggregates individual flows and provides only a number of services to the aggregated data flows. Intserv model suffers from scalability problem, while Diffserv model can only provide coarse QoS granularity. Recently, a novel distributed end-to-end QoS provisioning architecture has been proposed [3], which can optimize the user benefits from network services point of view, enhance network resource utilization and end-to-end QoS granularity, while maintaining the simplicity and scalability feature of the Diffserv network architecture.

Moreover, a new concept, *service vector*, was introduced by which an end host can choose different services at different routers along its data path [4].

The power limited and time-varying nature of the wireless domain makes most of the QoS provisioning techniques originally designed for wired environments unsuitable for wireless networks. Recently, some solutions have been proposed for offering QoS in one-hop wireless networks. Nevertheless, QoS provisioning in multi-hop wireless ad hoc network domain has not been yet addressed successfully, since these networks pose further challenges due to their infrastructureless and multi-hop nature.

In this thesis, the proposed *service vector* paradigm is extended to wireless ad hoc networks. A cross-layer architecture that combines link level scheduling and network level service vector concept is proposed and evaluated. It is demonstrated that within this architecture the *service vector* paradigm can enhance the end-to-end QoS in wireless ad hoc networks via reducing the power consumption.

The rest of the thesis is organized as follows: Chapter 2 describes some related work about Internet QoS provisioning, while Chapter 3 provides a summary of the Diffserv framework and describes the *service vector* concept. The emphasis in Chapter 4 is placed mainly on the methodology that is proposed in extending the service vector concept to wireless ad hoc networks, and on the description of the wireless scheduling policy used in the proposed architecture. Chapter 5 contains the performance evaluation of the proposed approach, while Chapter 6 concludes the thesis by summarizing the main contributions and conclusions of this work, and discussing the directions for future work.

CHAPTER 2

RELATED WORK

QoS provisioning in the Internet can be categorized into two realms: single-node and end-to-end QoS provisioning techniques. The former mainly deals with packet forwarding and resource management strategies in a single node, whereas the latter tries to guarantee QoS in an end-to-end fashion, which is the actual performance of the network from the user perspective. Internet Protocol (IP) was originally designed to provide best-effort service for delivery of data packets; i.e., there is no end-to-end performance guarantee. However, the emerging real-time communications and multimedia applications in the next generation Internet necessitate the provisioning of different levels of performance guarantees.

There are two well-known fundamental frameworks to provide end-to-end QoS in the Internet: Intserv and Diffserv. The basic approach of Intserv is to reserve network resources on a flow-by-flow basis, which requires routers to maintain state information on each flow. However, processing of every individual flow on core Internet routers introduces the scalability problem, since thousands of flows may exist simultaneously at a backbone router and hence, creating large amounts of processing overhead [6]. Within the framework of Intserv model Resource Reservation Protocol (RSVP) was proposed for signaling, in which the sender regularly transmits RSVP "Path" messages for the nodes along the path to store the path information and the receiver sends back "Resv" messages specifying the desired QoS and setting up the reservation state at each node. This way, RSVP enables the nodes to have the reservations periodically refreshed [7].

Diffserv was introduced in order to overcome the scalability problem of Intserv. Per-flow service is replaced by per-aggregate service and the complex processing is moved from the core of the network to the edge. Service Level Agreement (SLA) between the user and the service provider specifies the number of services to be provided, each of which has certain Per-Hop-Behavior (PHB)'s. Flows are aggregated at the edge router according to their service class and forwarded to the core routers, where they are served with respect to their corresponding PHB's. Although scalability problem of Intserv does not exist in Diffserv, due to the limited number of service classes, flows using the same service along the data path receive the same QoS support even if they might have quite different QoS requirements and hence, leading to the problem of coarse QoS granularity.

This trade-off between scalability and granularity is attempted to be alleviated by different hybrid approaches, one of which is Intserv over Diffserv architecture [8]. In this approach, allocation of the resources for the flows as well as the admission control procedure follow the Intserv fashion, whereas the data flows are served according to the Diffserv networks. On the other hand, authors in [3] and [4] proposed a robust QoS provisioning architecture that maintains the simplicity and scalability of Diffserv networks while increasing its QoS granularity by providing better service differentiation capability. The concept of *service vector*, in which the end host can choose different services at different routers along its data path, enables a flow to choose a certain service class at some other part of the network even though it might be congested and hence, inaccessible in one part of the network. This thesis mainly explores the implications of this *service vector* concept in wireless ad hoc networks.

CHAPTER 3

EXPLICIT ENDPOINT ADMISSION CONTROL AND SERVICE VECTOR

3.1 Introduction

Diffserv model is considered to be a more feasible and scalable solution for QoS provisioning in the Internet as compared to the Intserv model. However, in addition to the coarse QoS granularity problem, Diffserv model also has vague issues that should be resolved regarding the place and method of mapping an individual flow's QoS requirements to a specific service class, which is directly linked to issues associated with the place and time that the admission control decision for a certain flow will be made.

Currently, there are three admission control schemes in the literature hitherto proposed for Diffserv networks: static Service Level Agreement (SLA), dynamic admission control, and a recently introduced scheme called endpoint admission control (EAC). While static SLA admission control solution is inefficient, dynamic admission control has the flaw of being not scalable; therefore, EAC has lately been introduced to deal with the scalability issue [9] [10].

In EAC schemes, the end host performs the admission control decision; therefore, the scalability of Diffserv networks can be preserved. Admission control basically consists of two phases: probing phase and data transfer phase. In the probing phase, the end host sends probing packets to the network and records the resulting level of performance, and then the host determines whether or not to admit the flow. The network can be oblivious to the fact that it is being probed or it can cooperate with the process. In the former design option, long probing packet trains may be needed to obtain accurate results, although it has the benefit of being simple. On the other hand, if the network collaborates with the probing process by either marking or dropping the probing packets, more accurate results can be obtained and lower overhead can be achieved, due to the fact that long probing packet trains are no longer needed. Nevertheless, in all these EAC schemes, the data flow is permitted to use the same service along the data path; i.e., it is not allowed to use combinations of different service classes. Therefore, the probing stage specifies the performance of a particular service along the entire path rather than the performance of all services at every router.

3.2 Framework of Explicit Endpoint Admission Control and Service Vector

Explicit Endpoint Admission Control with Service Vector (EEAC-SV) scheme in essence relies on the idea of explicitly providing information about the performance of each service class at each router during the probing phase and enabling the routers to choose different service classes along the path during the data transfer phase. In other words, a *service vector* is selected at the end of the probing phase according to the QoS related information sent back by the network in the probing packets and this service vector is attached to the data packets in the data transfer phase. Assume that there are *m* routers along the path of a flow and *n* service classes at each router. The set of service classes can be denoted by $S = (S_0, S_1, ..., S_{n-1})$. A flow may choose service s_i at router i; where $s_i \in S$, which may be different from service s_j it chooses at router j. A service vector can be represented as $s = (s_0, s_1, ..., s_{m-1})$, where each element corresponds to the service chosen at the corresponding router along the path. Therefore, the end-to-end QoS granularity in the service vector approach is $O(n^m)$.

3.3 Optimization Models

Determination of the service vector is implemented via the following optimization procedure: Let us denote the end-to-end QoS performance of a certain flow by the vector $R = (R_0, R_1, ..., R_{K-1})$, where K is the number of QoS parameters to be considered, and the corresponding utility function by $U(R)$. The user cost function C depends on the service vector s of the flow as well as the set of pricing policies $p = (p_0, p_1, ..., p_{l-1})$. Then the optimization model is as follows:

$$G = \max_{s}(U(R) - C(s, p)) \tag{3.1}$$

$$s.t. \quad R(s) \in \Re \tag{3.2}$$

$$and \quad U(R) - C(s, p) \geq \delta \tag{3.3}$$

where $\Re$ is the vector space that meets the end-to-end QoS requirements of the data flow, s is the service vector that a flow may choose and δ is the minimum net benefit the user can accept. Since R is a function of the service vector s, it can be further decomposed into the performance that the flow experiences at each router, which is $(R^0, R^1, ..., R^{m-1})^T$, where $R^j = (R_0^j, R_1^j, ..., R_{K-1}^j)^T \quad (j = 0, ..., m-1)$ is the performance vector that the flow gets at router j. Since the performance at a router is a function of the service class chosen at that router, R^j is a function of s_j. Each user tries to optimize its own benefit; therefore, interaction among them can be modeled as a non-cooperative game. Similarly, the interaction among the routers at the network side can also be modeled as a non-cooperative game, since they also try to maximize their benefits. In this work the utility function $U(R)$ describes the user's level of satisfaction with the perceived

QoS and it is considered to be inelastic; i.e., the application does not care if more than required QoS is provided. In other words, the utility is the same as long as the provisioned QoS satisfies the desired bound.

The optimization model at the network side is as follows:

$$\max A_i(N^i, p) \tag{3.4}$$

$$s.t. \quad R^i(S_j, N_j) \in \Re^i(S_j),\ j = 0,1,\ldots,n-1 \tag{3.5}$$

$$and \quad N^i = \sum_{j=0}^{n-1} N_j \tag{3.6}$$

where $A_i(N^i, p)$ is the revenue function of router i, N^i is the number of flows at router i, p is its pricing policies, $R^i(S_j, N_j)$ is the performance vector of service S_j when there are N_j users of it, and $\Re^i(S_j)$ is the vector space of service S_j that meets the requirements for service differentiation at router i.

The various end-to-end service provisioning methodologies for Diffserv networks currently existing in the literature, namely the static service mapping and dynamic service mapping schemes, can be included in the afore mentioned user side optimization model. Therefore, we can evaluate and compare these two schemes together with the recently proposed scheme under the framework of three types of service vectors as follows:

Type 1: Conventional Scheme (EAC-CS) (Static Service Mapping): Users map their QoS directly to a certain class of service before the probing process and hence, the service vector s is a constant vector with $s_i = s_j, \forall\, i, j = 0,1,\ldots,m-1$. Probing packets use the same service as the data packets. If the measured QoS performance at the destination meets the QoS requirements, the flow is admitted. Otherwise, it is rejected. In other words, there is no maximization process of the optimization model; the end host only checks whether the statically mapped service class satisfies the requirements or not. The resultant G value can be denoted as G_{type1}, and the end-to-end QoS granularity is $O(1)$.

Type 2: EEAC with Single Class of Service Scheme (EEAC-SC) (Dynamic Service Mapping): The service vector is a constant vector as in EAC-CS; i.e., only one service class is used along the path. However, the flow is now dynamically mapped to an available best service class. The optimization procedure tries to find the service vector that satisfies the QoS constraint and maximizes the revenue among all the possible n service vectors. The resultant G value can be denoted as G_{type2}, and the end-to-end QoS granularity is $O(n)$.

Type 3: EEAC with Combination of Service Classes Scheme (EEAC-CSC) (Combination of Service Classes via the *Service Vector*): In this case, different service classes can be chosen at the routers along the path; therefore, there are n^m possible solutions. The user side optimization model operates on these n^m possible service vectors, which makes this problem NP-hard. The resultant G value can be denoted as G_{type3}, and the end-to-end QoS granularity is $O(n^m)$.

The highest network utilization and the finest end-to-end QoS granularity can be provided by the Type 3 solution, whereas Type 1 solution results in the worst QoS granularity and the lowest network utilization. Besides, solution space of Type 1 service vectors is a subset of the solution space of the Type 2 service vectors, which is also a subset of the solution space of Type 3 service vectors: $G_{type1} \leq G_{type2} \leq G_{type3}$

If the QoS parameter considered is the average end-to-end delay and the pricing policy is to charge each packet according to its assigned service class at each router, then the user side optimization model can be formulated as follows:

$$\min_{s} \sum_{i=0}^{m-1} C_i\ (s_i) \tag{3.7}$$

$$s.t. \sum_{i=0}^{m-1} delay(s_i) \leq Delay \tag{3.8}$$

where $delay(s_i)$ refers to the average delay of service s_i at router i and $Delay$ denotes the user's end-to-end average delay requirement.

3.4 Implementation of EEAC-SV

There are basically two issues in the implementation of this approach. The first one is how to assess and envisage the performance of each service class. The main assumption here is that the network conditions do not change considerably, so that the performances of a certain service class at a certain router during the probing phase and after the probing phase are quite similar. Besides, the route is assumed to be predetermined. To assess the performance of the service classes at a router, a predictor based on the Wiener process is used to estimate the buffer occupancy of a service class:

$$\overline{L}_{S_j}(t) = (1 - e^{-\tau_{S_j}(t)/K})L_{S_j}(t) + e^{-\tau_{S_j}(t)/K}\,\overline{L}_{S_j,old}(t) \tag{3.9}$$

where $\overline{L}_{S_j}(t)$ is the estimated buffer occupancy for service class S_j at time t, $\overline{L}_{S_j,old}(t)$ is the most recently updated average queue length before t, $\tau_{S_j}(t)$ is the interval between the arrival of the previous received packet of service class S_j and the current time t, and K is a constant. This estimator is actually the exponential moving average. Furthermore, three different congestion levels are defined and the predicted congestion level at the end of the probing period is calculated based on the estimated value of the buffer occupancy, and then it is mapped to the previously defined congestion levels. During probing, each router marks its corresponding bits in the probing packet to indicate the congestion level of each service class.

Weighted Fair Queuing (WFQ) is used as the scheduling policy, since it is a widely referred and accepted scheduling policy that can guarantee the delay bound of each service class. Therefore, the delay bound for service class S_j is found as: $D(S_j) = \frac{L_{S_j}}{R_{S_j}} + \frac{L}{R}$, where L is the maximum packet length, R is the output link rate, R_{S_j} is the guaranteed service rate of S_j, and L_{S_j} is the maximum buffer length of S_j.

It is assumed that three services are provisioned at each router: Expedited Forwarding (EF), Assured Forwarding (AF), and Best Effort (BE). In Diffserv networks, EF PHB corresponds to low loss, low latency, low jitter, and guaranteed bandwidth end-to-end service, whereas AF PHB provides a base rate while permitting the use of any available capacity. BE Forwarding does not provide any performance guarantees, which is the case in the conventional Internet infrastructure. The overall drop rate is lower in AF than pure BE due to the base capacity.

The definition of these service classes and congestion levels, each of which corresponds to a unique buffer occupancy ([3], [4]) are stated in Table 3.1, where $(d_{S_j,i}, p_{S_j,i})$ are delay and packet dropping probability bounds at each congestion level i. For EF and AF services, some random packet dropping scheme is assumed and $L_{S_j}^{\min}$ is the threshold above which this dropping algorithm starts dropping packets and $p_{S_j}^{\max}$ is the maximum packet dropping probability of service S_j at a router.

Table 3.1 Performance Bounds under Different Congestion Levels for the EF, AF, and BE Services

Congestion Level	0	1	2
EF	$(\frac{L_{EF}^{\min}}{R_{EF}}+\frac{L}{R},0)$	$(\frac{L_{EF}}{R_{EF}}+\frac{L}{R},p_{EF}^{\max})$	$(\infty,1)$
AF	$(\frac{L_{AF}^{\min}}{R_{AF}}+\frac{L}{R},0)$	$(\frac{L_{AF}}{R_{AF}}+\frac{L}{R},p_{AF}^{\max})$	$(\infty,1)$
BE	$(\frac{L_{BE}^{\min}}{R_{BE}}+\frac{L}{R},0)$	$(\infty,1)$	$(\infty,1)$

The simulation results in [4] indicate that Scheme 3 (*service vector* scheme) yields the best performance in terms of average cost per packet and request drop ratio, whereas Scheme 1 gives the worst performance. Another very important result is that Scheme 3 has the largest average end-to-end delay while still satisfying the delay bound, while Scheme 1 has the smallest delay, which is a direct result of the key underlying principle of the service vector scheme. For instance, if there is a flow with end-to-end average delay requirement of 750 ms; where class 0 service provides 100 ms, class 1 provides 200 ms, and class 2 provides 300 ms of delay bound, when this flow passes a route with three routers, Scheme 1 will map the flow to class 0 service at each router, which will result in 300 ms of delay, whereas Scheme 2 will make it use class 1 service along the entire path and hence, give rise to 600 ms. of end-to-end delay. However, Scheme 3 will map the flow to use class1-class2-class1 service, which will lead to an average delay of 700 ms that still satisfies the delay bound. In other words, Scheme 2 and Scheme 1 would waste network resources for providing better-than-required QoS guarantee, which might lead to some other flow that really needs a stringent delay requirement to be blocked.

CHAPTER 4

SERVICE VECTOR SCHEME IN WIRELESS AD HOC NETWORKS

4.1 Introduction

Providing QoS guarantees is an important objective in designing the next-generation wireless networks that need to support different applications with diverse QoS requirements. The limited battery resource at a mobile terminal coupled with the unpredictable nature of the wireless channel makes the problem of providing reliable wireless services a challenging task. Unlike wired networks power efficiency is a crucial parameter in the design of wireless networks, especially ad hoc wireless networks. In this thesis, it is demonstrated that *service vector* scheme enables significant power savings in wireless ad hoc networks when used in combination with an appropriate wireless scheduling discipline.

4.2 Problem Formulation

Assume that a flow going from its source to the destination passes through m intermediate routers, where the set of available service classes at each router is $S=(S_0,S_1,...,S_{n-1})$. After the probing phase is executed, the end host determines the service vector as $s=(s_0,s_1,...,s_{m-1})$, where the service class chosen at router i is denoted by $s_i \in (S_0,S_1,...,S_{n-1})$. The QoS parameter considered in determining the service vector is the average end-to-end delay; i.e., each service class s_i corresponds to a predetermined average delay bound $delay(s_i)$ and hence, the optimization problem that is implemented in finding the service vector is (3.7) and (3.8). Data transmission phase takes place in a time-slotted manner. To minimize the overall transmission power along the route once the service vector has been determined, the following problem is considered:

$$\begin{aligned} &\min E\{\overline{P}\} \\ s.t.\quad &E\{D_i\} \leq delay(s_i)\ \forall\, i \in (0,1,...,m-1) \end{aligned} \qquad (4.1)$$

where; $\overline{P} = \lim_{n\to\infty} \sum_{i=0}^{m-1} P_{i,n}$ $P_{i,n}$ is the power in time slot n at router i , D_i is the delay experienced at router i, and s_i is the service class chosen at router i. Apparently, the above problem is transformed to the link level scheduling problem of minimizing the average transmit power subject to the constraints on the average delay.

Delaying communication by decreasing the transmission rate to save power is commonly used in wireless systems, where usually the channel conditions are also taken into account. The strength of the *service vector* scheme is that it allows a flow to choose a service class with less stringent delay guarantees in some part of the network, even though that service class might be unavailable in some other part of the network. This way, transmission rate can be decreased at that node, which in turn reduces the power consumption. Therefore, this cross-layer approach of using the service vector scheme in combination with an appropriate scheduling discipline can enable the network to have significant power savings, which is vital for the efficient operation of wireless ad hoc networks.

4.3 Related Work on Wireless Scheduling Disciplines

Scheduling in the context of changing the key physical layer parameters like transmission rate and power is widely explored in the wireless domain, where the scheduling action usually depends on the channel state. Since packet delay is a key criterion in most real-time applications, delay constraints have also been incorporated into the design of schedulers in wireless networks. Both single-user and multi-user scheduling have been studied in the literature, where in the latter case minimizing the average transmission power also has the importance of reducing interference to other users in the system and hence, leading to higher throughput.

A well-known power control scheme, called water-filling, provides high transmission rate and consequently high power in good channel states and low transmission rates in poor channel conditions [11]. The information-theoretic concept of power and rate control has also been revealed in the INFOSTATION [12] architecture, where the nodes transmit data only when they are close to the base stations, and therefore transmission is delayed under bad channel conditions.

Results in [13] and [14] indicate that power control is useful not only in fading channels, but also in time-invariant Additive White Gaussian Noise (AWGN) channels. The authors in [13] attempt to minimize transmission energy under a pre-specified average packet delay constraint. They propose offline and online *lazy* schedulers in discrete AWGN channels, which make use of silent periods in packet arrival times by reducing the speed of transmission at these particular epochs. On the other hand, the work in [14] analyzes the relation between power and different concepts of delay as well as buffer control policies.

Furthermore, delay constrained scheduling has extensively been explored for fading channels. The authors in [15] explore transmission strategies that minimize the average power subject to the constraints on average delay as well as the peak transmitter power. Assuming the simple Gilbert-Elliott wireless channel model, they formulate a dynamic programming technique to solve this problem in a time-slotted system. Alternatively, power control schemes in independent identically distributed block-fading AWGN channels with a strict transmission delay constraint are explored in [16]. Although this work emphasized more on the power control rather than the scheduling framework, it gives insights to the relations between power minimization and delay constraints in fading channels.

In addition to the average delay constraint, some schedulers proposed in the literature also take the packet loss rate into account. For instance, the scheduler in [17] seeks to minimize the packet loss rate subject to the constraints on average delay and power, whereas the scheduler proposed in [18] considers the dual of this problem; i.e., it finds the optimal scheduling policy in finite state block fading channels under an average delay and packet loss constraint.

Delay constraints in fading channels make the traditional notion of capacity; i.e., Shannon capacity [19], insufficient to provide any actual delay guarantees. Therefore, a novel concept called *delay limited capacity* was introduced in [20], which refers to the maximum achievable rate with delay constraint, independent of how slow the fading is. On the other hand, authors in [21] investigated the average delay and average power trade-off in discrete-time block-fading AWGN channels and quantified the behavior of this trade-off in asymptotically large delay.

There have also been several works in the literature, which focus on the multi-user scheduling problem under various scenarios. The work in [22] concentrates on CDMA scheduling on the forward link channel, where the QoS parameter considered is the probabilistic packet delay bound. Downlink scheduling in CDMA networks has been addressed in [23], whereas delay bounded formulations over multiple access channels are evaluated in [24]. Scheduling in a decentralized system like ad hoc networks has also been studied in the literature: MAC layer fairness in shared channel wireless networks is explored in [25], whereas a distributed priority scheduling mechanism for ad hoc networks is proposed by [26].

4.4 Delay Bounded Power Efficient Schedulers

A scheduler which minimizes the average transmit power while satisfying the delay bounds of the three different service classes, namely EF, AF, and BE classes, is required to solve (4.1). Various optimal and suboptimal design techniques to tackle this problem have been proposed by [27] and [28], which are mainly utilized in this thesis as the power efficient scheduling mechanism.

Optimal and suboptimal design techniques for single user case [27] as well as multi user case under both TDMA and CDMA frameworks have been introduced [28]. The optimal solution for the single user scenario has been utilized in the suboptimal solution for the multi user scenario. Therefore, here the single user case also needs to be highlighted, although the emphasis of this thesis is placed on the multi user case.

4.4.1 Power Efficient Single User Scheduling

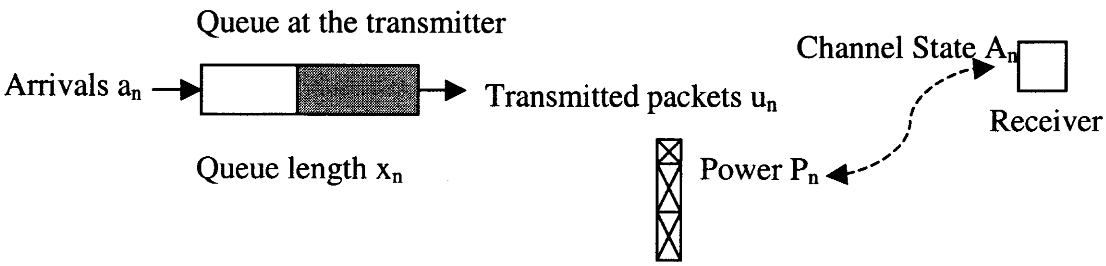

Figure 4.1 System model for single user scheduling.

A single-user time-slotted system is considered, where a_n is the number of packets that arrive at time slot n. Let us consider that at the beginning of the n^{th} time-slot, there are x_n packets in the buffer, u_n packets are chosen by the scheduler for transmission, and power P_n is used for this purpose. Lets also denote by M is the maximum number of packets that can arrive in a time slot and by L the buffer size. The arrival process $\{a_n\}$ is considered independent and identically distributed (i.i.d.) from one time slot to another and the average arrival rate is $E\{a_n\} = \lambda$ packets per time slot, where the size of each packet is S bits. The length of the time slot is T_s seconds, which is a constant value; therefore, the transmission rate is varied based on the chosen number of packets u_n. The buffer update is as follows:

$$x_{n+1} = \min(x_n + a_n - u_n, L) \tag{4.2}$$

The average packet delay is related to the average buffer occupancy via Little's theorem:

$$D_{avg} = \frac{1}{\lambda} E\{x_n\} \tag{4.3}$$

All packets arriving in time slot n can be transmitted in time slot $(n+1)$ or later. A scheduler is a mapping from the current buffer state x_n and channel state A_n to the number of packets transmitted u_n and corresponding transmission power P_n . The schedulers considered in [27] [28] as well as in this thesis are *zero-outage* schedulers. Zero-outage schedulers do not drop any packets, ensure reliable reception of the packets at the other end of the link, and avoid buffer overflows.

To prevent buffer overflow and packet dropping, either of the following two conditions must be satisfied for each state $x_n = i, \quad i \in L-M,\ldots,L$

1) Stationary probability for state i equals zero; i.e., $s_i = 0$

2) If $z = \min\{j : \alpha_{j,i} \neq 0\}$ represents the minimum number of packets transmitted in state i, then $(i-z) \leq (L-M)$

Furthermore, to guarantee a certain level of reception reliability at the receiver, the scheduler chooses the power level P_n such that u_n is equal to the Shannon capacity function for a Gaussian channel, which may be defined as follows [19]:

$$\rho(P_n, A_n) = \frac{T_s}{S}\log(1 + \frac{|A_n|^2 P_n}{\sigma^2}) \tag{4.4}$$

where for simplicity, it is assumed that the noise variance is $\frac{\sigma^2}{|A_n|^2}$, the system bandwidth is 1 Hz, and $T_s = S\log_2(e)$. Therefore, the following power control is employed:

$$P_n(u_n, A_n) = \frac{\sigma^2}{|A_n|^2}(e^{u_n} - 1) \tag{4.5}$$

Since P_n is a function of u_n, we can denote the scheduler as $\alpha : (x_n, A_n) \mapsto u_n$. This relation can be simplified even further for AWGN channels as $\alpha : x_n \mapsto u_n$, since $A_n \equiv 1$ for AWGN channels.

Let us also represent the probability of transmitting j packets given that the queue has i packets by $\alpha_{j,i}$, that is:

$$\alpha_{j,i} = \Pr ob(u_n = j/x_n = i) \tag{4.6}$$

Then every scheduler α can be symbolized by an $(L+1)\times(L+1)$ upper triangular matrix, since $\alpha_{j,i} = 0 \; for \; j > i$. Subsequently, it is easy to see that the following statement also holds true for every scheduler α:

$$\sum_{j=0}^{i} \alpha_{j,i} = 1 \tag{4.7}$$

Schedulers are stationary and memoryless; and hence, the queue state forms a first-order Markov chain. In the following let $\mathbf{C} = \lfloor C_{j,i} \rfloor$ represent the matrix of transition probabilities, where $C_{j,i}$ is the probability of transitioning from buffer state $x_n = i$ to buffer state $x_{n+1} = j$. The stationary probability of buffer state i is denoted by s_i, where $s_i = \Pr ob[x_n = i]$, and the vector of stationary probabilities is $\mathbf{s} = [s_0 \; s_1 \; ... \; s_L \;]$. Then, it follows that $\mathbf{C}\,\mathbf{s} = \mathbf{s}$. Note that $\mathbf{C}$ depends on the choice of α and therefore, $\mathbf{s}$ is a function of α.

Under this framework, the average packet delay and average power of the scheduler can be expressed as follows:

$$\begin{aligned} D_{avg}(\alpha) &= \frac{1}{\lambda} E\{x_n\} \\ &= \frac{1}{\lambda}(\sum_{i=0}^{L} i\, s_i) \end{aligned} \tag{4.8}$$

$$\begin{aligned} P_{avg}(\alpha) &= E\{P_n\} \\ &= \sum_{i=0}^{L}\sum_{j=0}^{i} s_i\, \alpha_{j,i}\, P_n(j,1) \end{aligned} \tag{4.9}$$

where P_n is the power control defined in (4.5).

The single flow optimum scheduler that minimizes the average transmit power while satisfying the bound on average delay is the solution to the following optimization problem:

$$P^*(D_0) = \min_{\substack{\theta \cap W \\ D_{avg} \le D_0}} \lim_{n \to \infty} E\{P_n\}$$

which can also be expressed as follows:

$$P^*(D_0) = \min_{\substack{\theta \cap W \\ E\{x_n\} \le \lambda D_0}} \lim_{n \to \infty} E\{P_n\} \tag{4.10}$$

A dynamic programming technique known as *Value Iteration Algorithm* is utilized to solve this problem, which can be outlined as follows:

$C(i,a)$ = Cost of doing action a in state i

$P_{ij}(a)$ =Probability of transitioning from state i to state j under action a

x = A predetermined state

ϵ = A small positive number that determines the stopping criterion.

k = k^{th} iteration of the algorithm

1. Initialize $v_0 \equiv 0, \quad \delta = 1,$ and $k = 0$

2. Evaluate $w_k(i) = \min_a\{C(i,a) + \sum_j P_{ij}(a)v_k(j)\}$

3. Set $\delta = |w_k(x) - w_{k-1}(x)|$ and $v_{k+1}(i) = w_k(i) - w_k(x)$

4. Repeat steps 2 and 3 until $\delta < \in$

5. The optimal actions for each state i are as follows:

$$a^*(i) = \arg\min_a\{\, C(i,a) + \sum_j P_{ij}(a)v_k(j)\,\}$$

In the solution for problem (4.10), $i \equiv x_n$, $a \equiv u_n$, and $P_{ij}(a)$ depends on the arrival distribution. Furthermore, for Gaussian channels, $C(i,a) = P_n(u_n, A_n, \sigma^2) + \varepsilon x_n$, where ε is the Lagrangian used in (4.10).

There are two issues with respect to the implementation complexity of this single-flow optimal scheduler. First of all, the number of possible states in the VIA increases substantially for large buffer sizes L and hence, it is computationally involved. Secondly, the algorithm depends on the knowledge about the arrival distribution, which may not be available. When the arrival distribution is measured in real-time, the optimal scheduler needs to be adapted as the measurements indicate a different arrival distribution, which can lead to an intractable design. Apart from these, we have also observed that the Lagrangian value ε used in (4.10) is mathematically very difficult to obtain, which is another drawback of this optimal scheduler.

To alleviate these drawbacks, the authors in [27] presented a suboptimal scheduler, called *log-linear* scheduler, which greatly simplifies the scheduler design at the expense of slight performance degradation when compared to the optimal scheduler:

$$u_n = \min(x_n, \lfloor \log(\kappa x_n) \rfloor) \tag{4.11}$$

where parameter κ is chosen to meet the delay bound.

4.4.2 Power Efficient Multi User Scheduling

The multi-user scheduling is treated in a similar way to the single-user scheduling problem [28]. A time-slotted system with K flows transmitted over a shared wireless channel is considered:

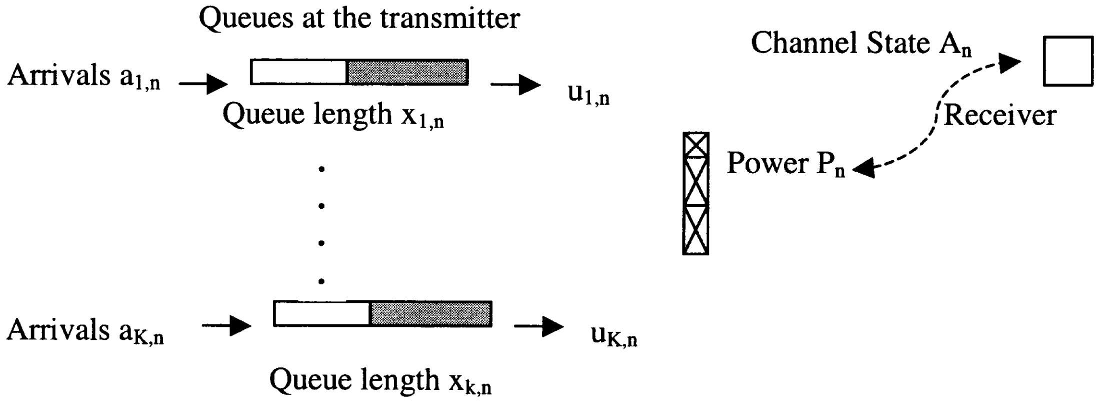

Figure 4.2 System model for multi user scheduling.

The number of packets that arrive at queue i in time slot n is denoted by $a_{i,n}$. We also consider that at the beginning of the n^{th} time-slot, there are $x_{i,n}$ packets in the buffer of the i^{th} flow, $u_{i,n}$ packets are chosen by the scheduler for transmission and power $P_{i,n}$ is used for this purpose. Furthermore, M_i is the maximum number of packets that can arrive at queue i in a time slot and L_i denotes the buffer size. The arrival process of each flow $\{a_{i,n}\}$ is independent and identically distributed (i.i.d.) from one time slot to another as well as being independent from the other flows. The average arrival rate for flow i is $E\{a_{i,n}\} = \lambda_i$ packets per time slot, and the size of each packet is S bits. As in the single user scenario, the length of the time slot is T_s seconds, which is a constant value. Therefore, the transmission rate is varied based on the chosen number of packets $u_{i,n}$. In this case the buffer update is as follows:

$$x_{k,n+1} = \min(x_{k,n} + a_{k,n} - u_{k,n}, L_k), \ \forall\, k,n \qquad (4.12)$$

In the multi user case, the set of all buffer states are represented by a $1 \times K$ vector, $\mathbf{x}_n = \lfloor x_{1,n}\ x_{2,n} \ldots x_{K,n} \rfloor$. The set of all possible buffer states $\{x_n^i\}_i$, where x_n^i is a specific value of $\mathbf{x}_n$, can be represented by Ω.

The average packet delay D_k of user k can be computed via Little's formula as follows:

$$D_k = \frac{1}{\lambda_k} \sum_{x^i \in \Omega} x_k^i s_i \qquad (4.13)$$

where x_k^i is the number of packets in the k^{th} buffer when buffer state is x^i. A scheduler is a mapping from the current buffer state $\mathbf{x_n}$ and the vector of channel states $\mathbf{h_n} = [h_1 \ldots h_K]$ to $\mathbf{u_n}$ and transmission power P_n, where $\mathbf{u_n}$ implies both the number of packets and the buffer chosen for transmission.

The actual transmission power depends on the specific multiple access scheme used. In the following we assume a TDMA system, and therefore the corresponding power can be expressed as follows:

$$P_{tdma}(\mathbf{u}) = W\, N_k\, (e^{u_{k,n} F/W} - 1) \qquad (4.14)$$

where W is the bandwidth, $F=\frac{T_s}{S}$, and k is the index of the user transmitting in time slot n. Therefore, the total transmission power P is:

$$P=\sum_{x^i\in\Omega} s_i\, P_{tdma}(u(x^i)) \qquad (4.15)$$

As in the single user case, *zero outage* schedulers are considered. Similar to the requirements for the single user case, a randomized stationary multi-user scheduler is *zero outage*, if and only if for each state $x^i \in \Omega$, one of (1) or (2) below is satisfied. In TDMA systems, only one user can transmit in a certain time slot, and condition (3) below also has to be satisfied:

(1) The stationary distribution is zero for state x^i; i.e., $s_i = 0$.

(2) If z^i_k symbolizes the minimum number of packets transmitted from the k^{th} queue in state x^i, then $(x^i_k - z^i_k) \le (L_k - M_k), \forall k$

(3) $x_k \ge (L_k - M_k)$, for at most one $k = 1,2,...,K$

The objective in the multi-user scheduler design is to minimize the expected transmission power while satisfying the average delay bounds of all users. The optimization problem can be stated as follows:

$$P^*(\mathbf{D}_0)=\min_{\substack{\theta\cap W\\ E\{x_{k,n}\}\le\lambda_k D_{k,0},\forall k}} \lim_{n\to\infty} E\{P_n\} \qquad (4.16)$$

where $D_{k,0}$ is the delay bound for the k^{th} user, and $\mathbf{D}_0 = \lfloor D_{1,0} \quad D_{2,0} \ldots D_{K,0} \rfloor$ is the vector of delay bounds. As in the single user scheduling case, *Value Iteration Algorithm* is used to solve this problem, except that in this case VIA symbols have the following association: $i \equiv x^i$, $a \equiv u(x^i)$, $C(i,a) = P_{tdma}(\mathrm{u}_n) + \varepsilon x_n$, where ε is the Lagrangian used in (4.16). Moreover, for TDMA systems, $\mathbf{a} \equiv \mathbf{u}_\mathrm{n}$ represents all $\mathbf{u}_\mathrm{n}$ that have non-zero values for at most one element of $\mathbf{u}_\mathrm{n}$. For finite state fading channels, there is a separate state i for all different $\mathbf{x}_\mathrm{n}$ and $\mathbf{h}_\mathrm{n}$ combinations.

Nevertheless, VIA in multi-user case has the same drawbacks as in the single user case. Therefore, suboptimal schedulers have been proposed to alleviate these shortcomings. The suboptimal TDMA scheduler proposed can be stated as follows:

Step 1. Flow Choice: Given the vector of buffer states $\mathbf{x}_n$, the index k of the flow chosen to transmit is:

$$k = \begin{cases} l & \mathit{if}\ x_l > L_l - M_l \\ \arg\max \dfrac{x_l}{\lambda_l D_{l,0}} & \mathit{otherwise} \end{cases} \tag{4.17}$$

where the first condition ensures zero buffer overflow, whereas the second one chooses the flow that is closest to violating its delay bound.

Step 2. Number of Packets: The number of packets transmitted is computed using the single flow optimal scheduling discipline of (4.10).

CHAPTER 5

PERFORMANCE EVALUATION AND DISCUSSIONS

5.1 Objective and Models

In this chapter, the performance of the proposed methodology is achieved via modeling and simulation, using the OPNET Modeler. Specifically, two different types of simulation scenarios have been considered, in both of which the route of a flow is assumed to be predetermined. The first scenario evaluates the performance of a single flow from source to sink, where the wireless links are assumed to be AWGN channels, whereas in the second scenario two flows with different QoS requirements are considered. In order to study the impact of traffic arrival distribution, either case is simulated under both uniformly distributed and on-off arrival distributions.

The suboptimal TDMA scheduler in [28] is modified by having the scheduler to transmit according to the suboptimal log-linear scheduler rather than the optimal single flow scheduler in step 2 of the algorithm. There are basically three reasons for this design choice. First of all, the number of possible states in the VIA grows exponentially as the buffer size and the number of queues in the system increases. Since there are only three buffers in our case because there are three service classes at each router, it does not contribute significantly to the computational complexity of the VIA. However, the buffer size limitation is still in place and finding the optimal scheduler becomes computationally intensive as the buffer sizes increase.

Secondly, the Lagrangian value ε in the VIA was found to be mathematically very difficult to obtain, even for the single user optimum scheduler.

Thirdly, VIA requires knowledge about the arrival distribution. In situations where the optimal scheduler is adapted over time, the implementation of VIA can lead to an intractable design. There are basically two options to obtain information about the arrival distribution. The first option is that the arrival distributions can be measured in real time; however, this option is not feasible not only because it introduces additional implementation complexity, but also because the measurement results might be inaccurate and can lead to erroneous scheduler decisions. The second option is that each router along the path can compute its output distribution and send this information to the next router along the path. Nevertheless, this option also introduces extra messaging overhead in the network while the computation of the output distribution is not easy. This may be illustrated for the one buffer and two buffer cases as follows:

$$\begin{aligned}\Pr ob(b_n = j) &= \sum_{k=0}^{L} \Pr ob(x_n = k) \times \Pr ob(u_n = j / x_n = k) \\ &= \sum_{k=0}^{L} s_k \times \alpha_{j,k} \qquad (5.1)\end{aligned}$$

where b_n denotes the output distribution of the router.

Since each router knows its own scheduler actions, $\alpha_{j,k}$ values are known; therefore, the router only needs to compute its vector of stationary probabilities of the buffer states: $\mathbf{s} = [s_0 \ s_1 \ ... \ s_L]$. The stationary probability of buffer state j can be expressed as: $s_j = \sum_{i=0}^{L} \sum_{t=j-i}^{j} s_i \times \Pr ob(a_n = t) \times \alpha_{t+i-j,i}$. Given the fact that all the stationary buffer probabilities sum up to 1; i.e., $\sum_{j=0}^{L} s_j = 1$, a system of $L+2$ linear equations needs to be solved.

For the two buffer case, the expression for the stationary buffer probabilities is even more complicated. The stationary probability of buffer state (x_1, x_2) can be conveyed as: $s(x_1,x_2) = \sum_{i=0}^{x_1}\sum_{j=0}^{x_2}\sum_{k=0}^{L_1}\sum_{m=0}^{L_2} s(k,m)\times\alpha_{[k-i,m-j],[k,m]}\times \Pr ob(a_n = [x_1 - i, x_2 - j])$

Considering the fact that we have three buffers in our case, computation of the arrival distribution is apparently intensive. Moreover, for *Type 3* service vector case, rather than being computationally involved, it is mathematically impossible to compute the arrival distribution for a certain buffer, since the fact that a certain number of packets leave the buffer of a certain service class at a router does not necessarily mean that they are going to use the same service class at the next router. Therefore, even the suboptimal multi-user scheduler where the optimal single user scheduler is used in the second step is impossible to be implemented for the service vector scheme.

Due to the reasons mentioned above, the suboptimal multi-user scheduler proposed by [28] was modified by having *log-linear* scheduler in the second step of the algorithm. Besides, zero buffer overflow is also ensured by guaranteeing that $x_k \geq (L_k - M_k)$, for at most one $k = 1,2,...,K$. The limit on the output rate of the scheduler was set to be equal to the maximum number of packets that can arrive at the router in a certain time slot; i.e., the maximum number of packets that the scheduler can transmit equals $M_1 + M_2 + M_3$.

First of all, since the limit on the scheduler output rate is $M_1 + M_2 + M_3$, the maximum number of packets that can arrive at any buffer at the next router along the path is $M_1 + M_2 + M_3$. To ensure zero buffer overflow at this subsequent router, the buffer sizes for each of the service classes must satisfy: $L_k \geq M_1 + M_2 + M_3$, for all $k = 1, 2, ..., K$. Similarly, since the output rate limit of this second router is $3 \times (M_1 + M_2 + M_3)$, which is in turn equal to the maximum number of packets that can arrive at any buffer at the third router along the path, the buffer sizes at the third router must satisfy: $L_k \geq 9 \times (M_1 + M_2 + M_3)$, for all $k = 1, 2, ..., K$. This exponential increase in the buffer sizes as the number of routers along the path increases may be regarded as a drawback of having zero buffer overflow on the entire path. Since buffer sizes cannot grow indefinitely, there is a limit on the path length depending on the maximum buffer size that can be allowed. One option to overcome this downside might be to allow packet dropping at the routers, which could also lead to less power consumption since the scheduler would not have to transmit at a higher rate as the buffer occupancies approach the buffer sizes. On the other hand, having larger buffer sizes has the advantage of being able to accommodate larger delays and hence, enabling less power consumption. In this thesis, the (L, M) pairs are $(20,6)$, $(60,18)$, and $(170,54)$ for the first, second and third routers, respectively, along the path. The κ parameters of the log-linear schedulers were chosen to meet the delay bounds. Time-slot length for the entire system is fixed; $T_s = 0.05s$.

The average delay and average power relation of the log-linear scheduler for $(L, M) = (20, 6)$ is as shown in Figure 5.1. The required power decreases quickly as the delay bound is relaxed from 0.1 to 0.2 seconds and the rate of decrease diminishes as the delay bound is relaxed further.

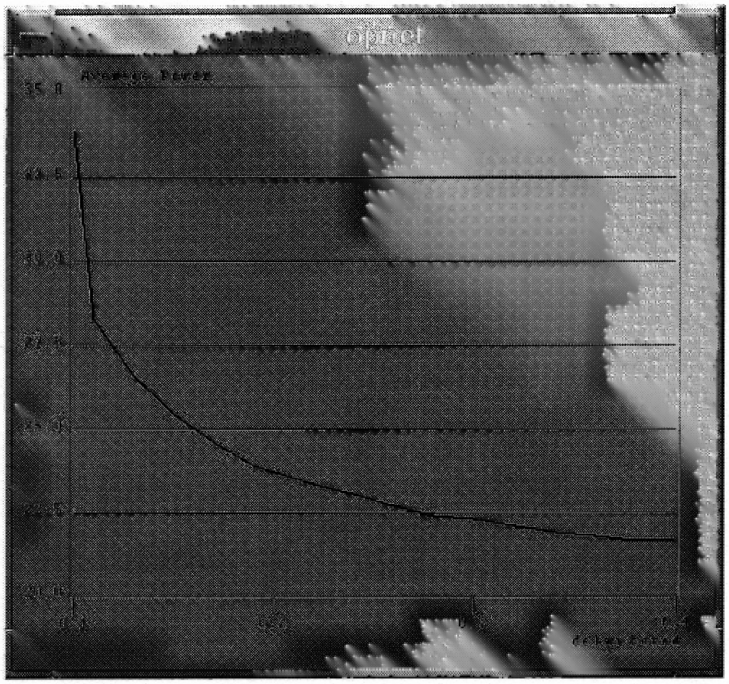

Figure 5.1 Average power and delay relation for sub-optimal log-linear scheduler with buffer size L=20 and maximum arrivals M=6.

On the other hand, Figure 5.2 illustrates the general behavior of the log-linear scheduler for an average delay bound of 150 ms. For most buffer states, the scheduler tends to transmit close to the average arrival rate $\lambda = 3$ packets/timeslot. The maximum rate of the scheduler equals M , the maximum number of packets that can arrive in a timeslot and buffer overflows are avoided by transmitting close to M packets when x_n is close to L.

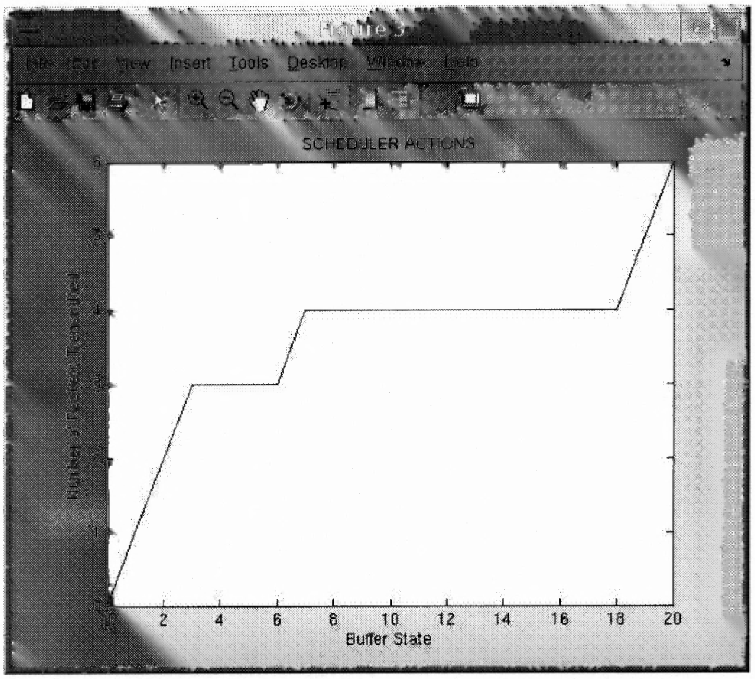

Figure 5.2 Scheduler actions for buffer size L=20, maximum arrivals M=6 and average delay bound of 150 ms.

As in [3], the source node sends a probing request to the network and gathers information about each service class at the routers. However, unlike the case in [3], the information that the routers attach to the probing acknowledgement packets is whether the service class is available or unavailable. The availability/unavailability of each service class is determined based on the arrival rate to that service class, which will be explained in more detail in Section 5.2.1. The end host then determines the best service vector among the available ones. The average delay bounds for the service classes are defined in Table 5.1.

Table 5.1 Service Class Definitions

Service Class	Average Delay Bound
EF (Class 0)	100 ms
AF (Class 1)	150 ms
BE (Class 2)	350 ms

5.2 Numerical Results and Discussions

5.2.1 Single Flow Scenario Performance Evaluation and Discussion

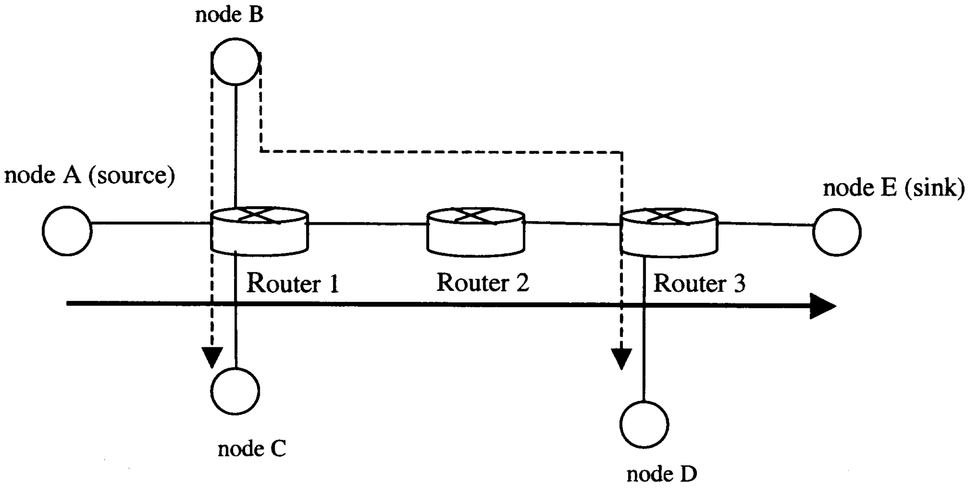

Figure 5.3 System model for single flow scenario.

The direction of the data flow whose performance was evaluated is from node A to node E. The number of packets generated by node A in one time slot is uniformly distributed with a finite support, $[0, \ldots, 4]$, i.e., the largest number of packets that arrive in a time slot is 4. Cross traffic is also uniformly distributed, and the maximum number of packets that can arrive in a time slot for the background traffic flows is summarized in Table 5.2. The buffering architecture at each router is illustrated in Figure 5.4:

Table 5.2 Summary of Background Traffic

Source	Destination	Class 0 (EF)	Class 1 (AF)	Class 2 (BE)
Node B	Node D	2 packets/slot	2 packets/slot	2 packets/slot
Node B	Node C	0 packets/slot	0 packets/slot	4 packets/slot

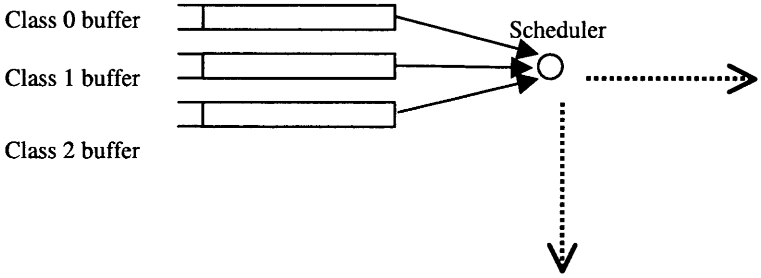

Figure 5.4 Router architecture.

Packets arriving at the router are placed in one of the three buffers depending on their service vector; i.e., the service class they are using at that router, in a FIFO manner. At the beginning of each time slot, the scheduler determines which buffer to serve and how many packets to transmit from that buffer at that time slot. Since TDMA system is used, only one buffer is allowed to transmit its packets at a certain time slot. Then the packets are taken from the head of the queue one by one in accordance with the scheduler decision and the destination of the packets is checked. The packet is then sent to the appropriate next node depending on the destination of the packet. In other words, Class 2 packets following the path from node B to node D and packets with source node B and destined to node C, which are Class 2 packets, are all placed in the same Class 2 buffer.

Figures 5.5 and 5.6 illustrate the average power consumption and delay of the service class buffers at the routers along the path before the flow from node A to node E starts sending traffic. Class 2 buffer at router 1 has significantly higher power consumption than the other buffers at router 1 as well as the buffers at the remaining routers. Besides, the average delay at this buffer is much better than its required value. Although Class 2 is the service class with the least stringent delay bound requirement, the average delay that it experiences is even less than the average delay of the service class 1. The reason for this is that the cross traffic overloads class 2 at router 1. As the arrival rate to a certain buffer increases quite considerably, the scheduler greatly increases the rate of transmission mainly in order to prevent buffer overflows as well as to meet the delay bound requirement. As a result, actual average delay of that service class becomes much better than its required value at the expense of enormous power consumption.

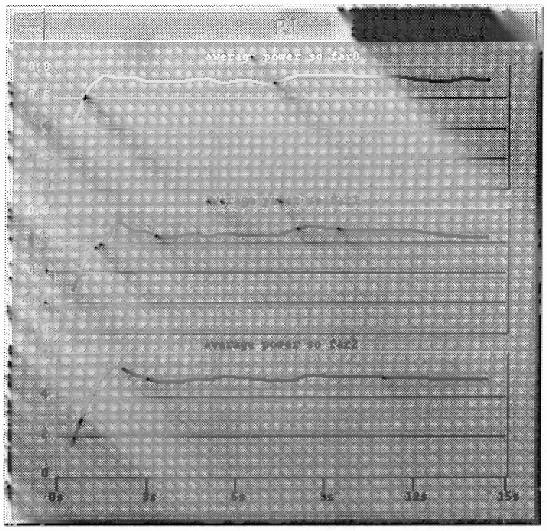

Figure 5.5 Average power of the router1 buffers before probing.

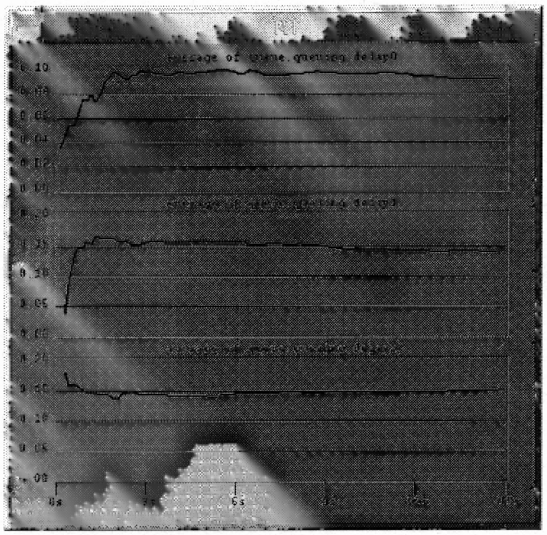
R1
average of queue.queuing delay0
0.10
0.08
0.06
0.04
0.02
0.00
average of queue.queuing delay1
0.20
0.15
0.10
0.05
0.00
average of queue.queuing delay2
0.20
0.15
0.10
0.05
0.00
0s
3s
6s

Figure 5.6 Average queuing delay of the router1 buffers before probing.

The method used in determining whether a service class is overloaded or not is another important issue. Estimated or current average values of the buffer occupancies might not reveal the congestion level of the service class. Especially if that service class is the one with loose delay requirements, its buffer occupancy might still be substantial in the routers that are not congested. In other words, there might not be a significant difference in buffer occupancies for that service class in cases where the routers are overloaded and in the ones where they are not. On the other hand, power consumption is related to the output rate of the scheduler, which is in turn directly related to the arrival rate. Therefore, estimated value of the arrival rate accurately reflects the fact that a certain service class is overloaded and can differentiate it from the buffers that are not overloaded along the path. Furthermore, schedulers have been designed according to the maximum number of packets that can arrive to the buffer in a time slot. If more packets than this value arrive at the scheduler, buffer overflow cannot be guaranteed anymore. Due to these two reasons, it is crucial to determine whether the current maximum number of packets arriving at the scheduler in a time slot is already close to its upper limit or not. Consequently, estimation of the arrival rate is used in this thesis to determine the availability/unavailability of a certain service class in the probing phase. Exponential moving average of the arrival rate is used in the estimation process. Considering that the cross traffic is uniformly distributed, if the estimated value of the arrival rate is greater than $\frac{M_{S_j}-1}{2}$, then the service class S_j is marked as *unavailable* by the corresponding router. The following expression is used in estimating the arrival rate, which is measured in terms of packets/time slot.

$$\bar{r}_{S_j}(t) = (1-e^{-\tau_{S_j}(t)/K})\frac{T_s}{\tau_{S_j}(t)} + e^{-\tau_{S_j}(t)/K}\bar{r}_{S_j,old}(t) \qquad (5.2)$$

where T_s is the time slot length in seconds, $\bar{r}_{S_j}(t)$is the estimated arrival rate for service class S_j at time t, $\bar{r}_{S_j,old}(t)$is the most recently updated arrival rate before t, $\tau_{S_j}(t)$ is the interval between the arrival of the previous received packet of service class S_j and the current time t, and K is a constant, which was selected to be 0.35 in this thesis. At each router, $\bar{r}_{S_j}$ is updated when a data packet of service class S_j is received.

Figures 5.7a, 5.7b, and 5.7c present the corresponding exponential moving average values of the arrival rates of the service classes at each router, which demonstrates that the fact that only service class 2 at router 1 is overloaded in this system is accurately reflected by the arrival rate estimates.

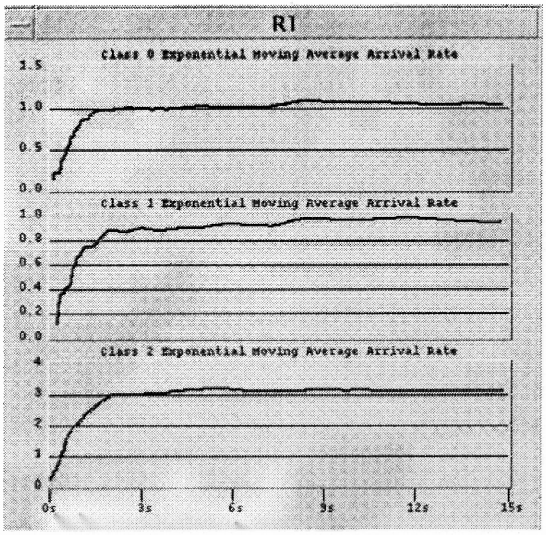

Figure 5.7.a Arrival rate estimates of the router1 buffers before probing.

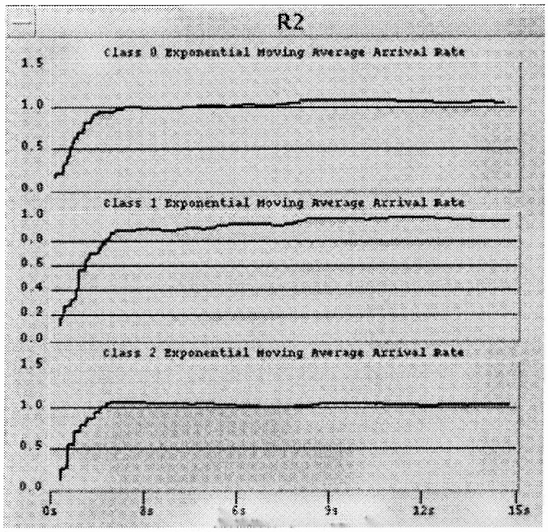

Figure 5.7.b Arrival rate estimates of the router2 buffers before probing.

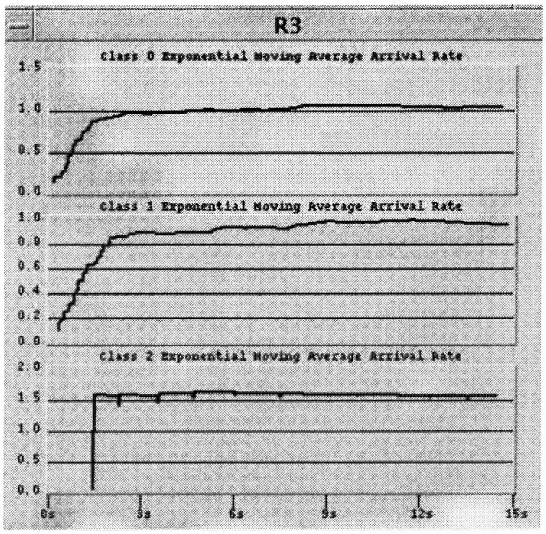

Figure 5.7.c Arrival rate estimates of the router3 buffers before probing.

The flow from node A to node E has an end-to-end average delay requirement of 950 ms. In the following the performance of three types of service vectors; i.e., *Type 1* service vector (Conventional Static Service Mapping Scheme (EAC-CS)), *Type 2* service vector (EEAC with Single Class of Service Scheme (EEAC-SC)), and *Type 3* service vector (EEAC with Combination of Service Classes Scheme (EEAC-CSC)) are evaluated. Since real-time data flows are assumed for the flows from node A to node E, they always use service class 0 in EAC-CS scheme. Figure 5.9 demonstrates that all the three schemes can satisfy the non-elastic end-to-end average delay requirement. EEAC-CSC results in longer end-to-end delay than EEAC-SC because it tries to make use of all possible combinations of service classes and makes each service class more loaded than in the EEAC-SC scheme; therefore, a packet may experience longer delay in EEAC-CSC. For the same reason, EAC-CS results in the smallest average end-to-end delay.

Figure 5.10 compares the average end-to-end power consumed by the flow from node A to node E for the three types of scheduling schemes. Scheme 3 results in the lowest power consumption, whereas Scheme 1 leads to the highest power consumption.

This demonstrates that the proposed approach of implementing the new *service vector* concept of allowing the data flow to choose different service classes at different nodes along the path in combination with the described delay bounded formulation of multi-flow wireless scheduling discipline results in significant power savings over the conventional static service mapping (EAC-CS) scheme, as well as over the single class of service (EEAC-SC) scheme. In other words, the method proposed in this thesis enables the *service vector* concept, which was originally developed for wire-line networks, to enhance the end-to-end QoS in wireless ad hoc networks.

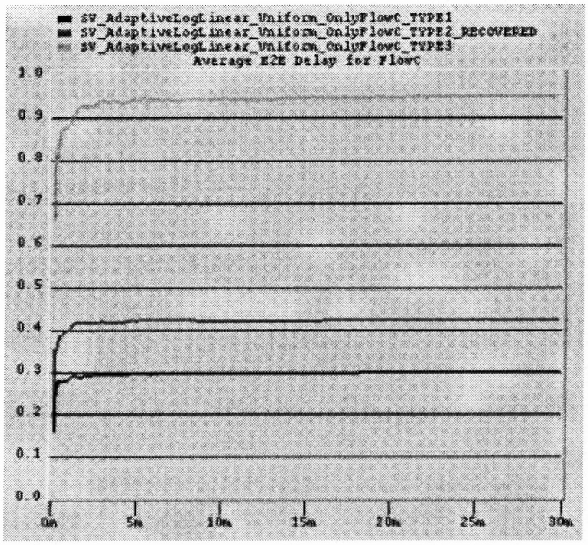

Figure 5.8 Average end-to-end delays of the three schemes for the single flow with uniformly distributed traffic.

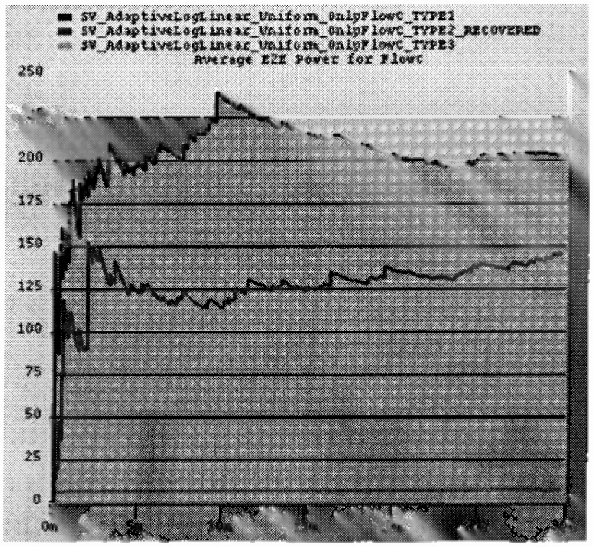

Figure 5.9 Average end-to-end power of the three schemes for the single flow with uniformly distributed traffic.

The performance of *On-Off* traffic has also been evaluated, where the probability of being in the *On* state is equal to the probability of being in the *Off* state and 4 packets are generated while the *On* state. Figures 5.11 and 5.12 exemplify the average end-to-end delay and power values of the same single flow scenario with this type of traffic. EEAC-CSC scheme clearly outperforms the other two schemes also for the *On-Off* arrivals. For all the three schemes, *On-Off* arrivals have higher power consumption than their uniform arrival distribution counterparts. The reason for that, can be attributed to the fact that the *On-Off* arrival process requires the highest transmit power at any delay in an AWGN channel among all arrival processes with the same average and finite maximum arrival rate [27].

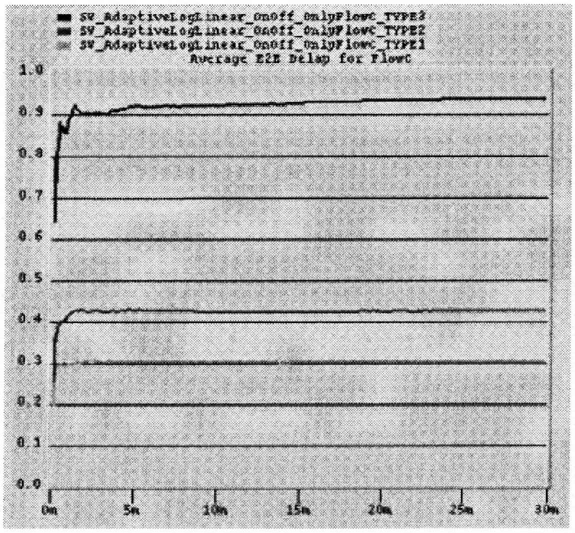

Figure 5.10 Average end-to-end delays of the three schemes for the single flow with *On-Off* arrival distribution.

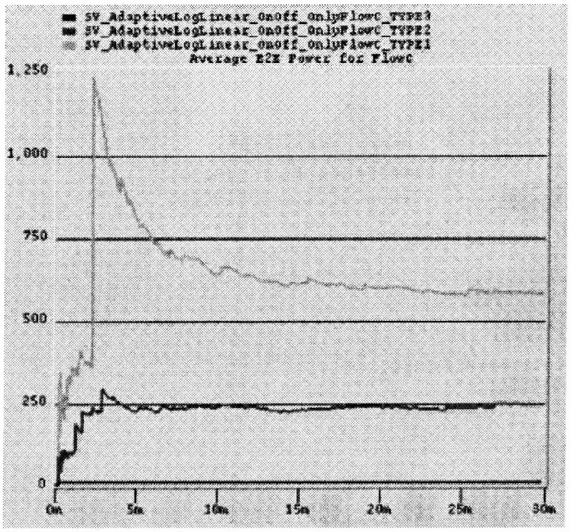

Figure 5.11 Average end-to-end power of the three schemes for the single flow with *On-Off* arrival distribution.

Furthermore, the influence of different arrival rates has also been evaluated for both uniform and *On-Off* arrival distributions. Figures 5.13 and 5.14 demonstrate that the proposed scheme outperforms the other two schemes for all of the arrival rates. The performance difference in power savings increases as the arrival rate increases. The exponential shape of the plots is due to the exponential relationship between rate and power.

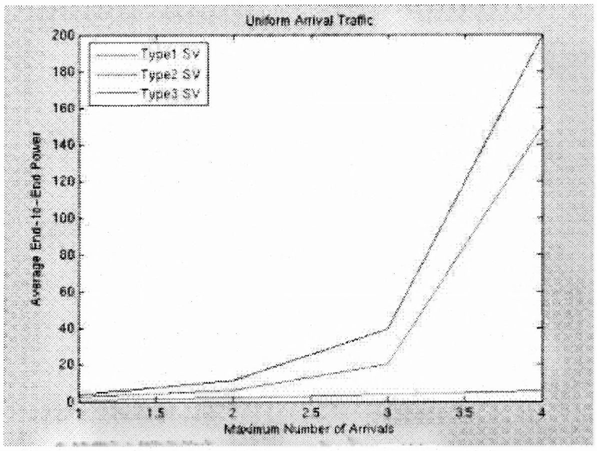

Figure 5.12 Average end-to-end power of the three schemes for the single flow with uniform arrival distribution and varying arrival rates.

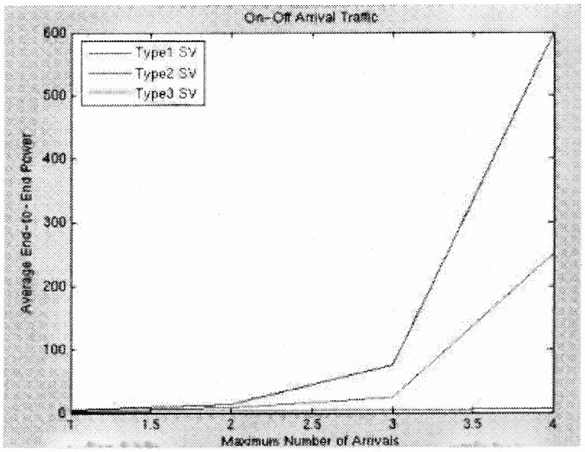

Figure 5.13 Average end-to-end power of the three schemes for the single flow with *On-Off* arrival distribution and varying arrival rates.

5.2.2 Two Flows Scenario Performance Evaluation and Discussion

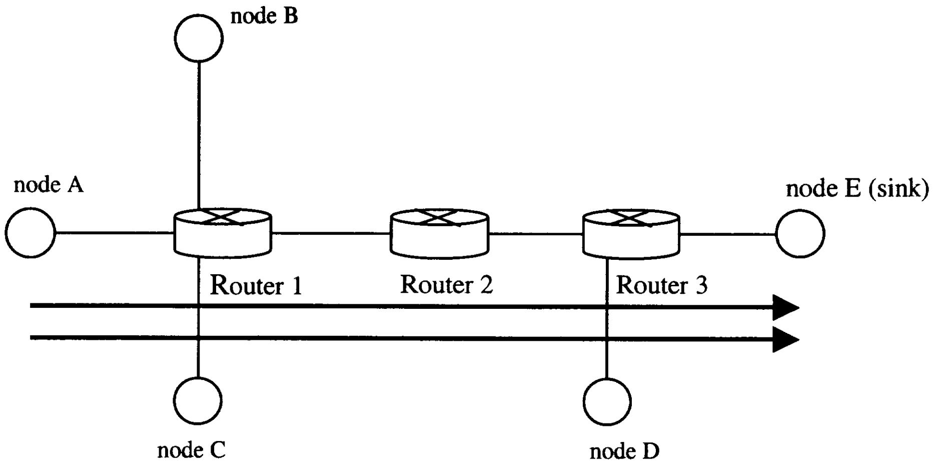

Figure 5.14 System model for two flows with different end-to-end QoS requirements.

In this scenario, there are two types of flows generated by node A; i.e., Flow C which has an end-to-end average delay bound of 950 ms. as in Section (5.2.1), and Flow D with an average end-to-end delay bound of 750 ms. 50% of the traffic generated by node A is Flow C and the remaining is Flow D. The total number of packets that can be generated by node A in a time slot is uniformly distributed with a maximum of 4 packets/time slot and cross traffic remains the same as in Section (5.2.1).

Figures 5.16 and 5.17 illustrate that EEAC-CSC scheme can provide service differentiation for these two different types of flows, whereas EEAC-SC and EAC-CS schemes are unable to provide this differentiation since they map these two flows to the same service vector. In other words, the method proposed in this thesis enables finer QoS granularity both in terms of average end-to-end power consumption and average end-to-end delay.

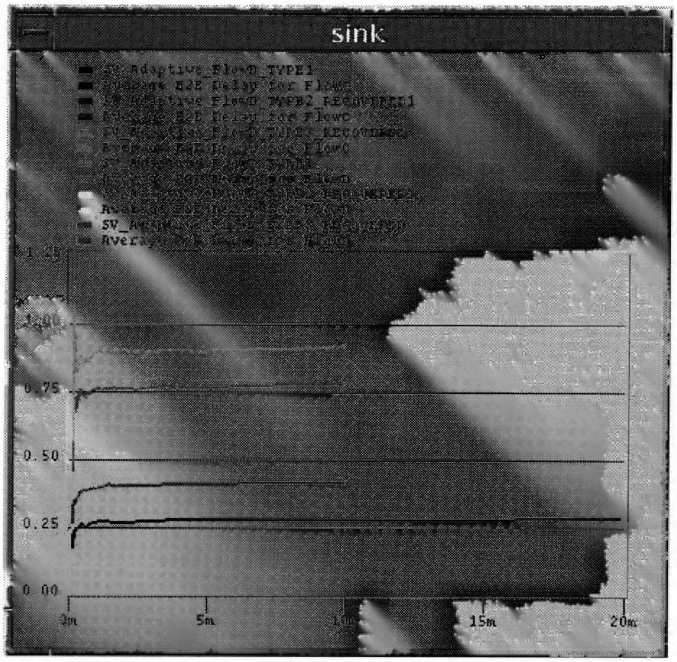

Figure 5.15 Average end-to-end delays of the three schemes for two different flows with uniform arrival distribution.

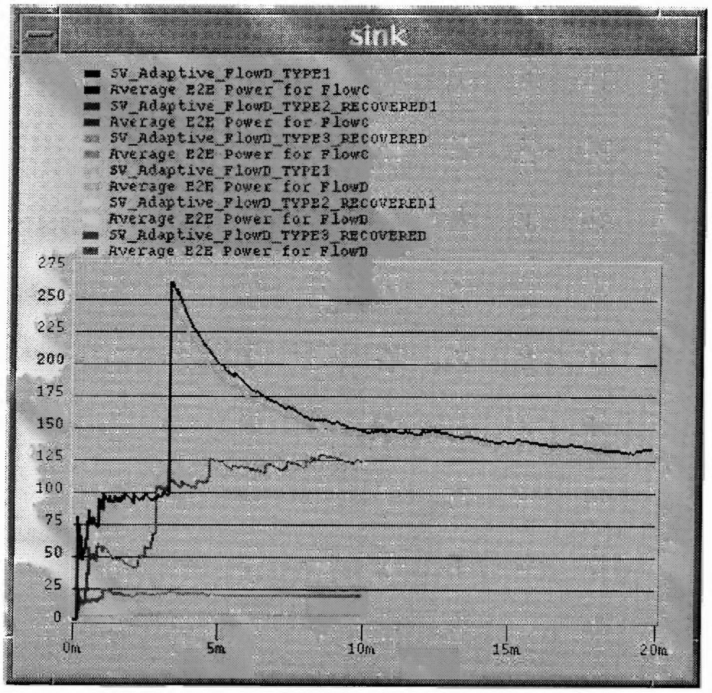

Figure 5.16 Average end-to-end power of the three schemes for two different flows with uniform arrival distribution.

Similarly, Figures 5.18 and 5.19 illustrate the end-to-end average delay and power consumption for these two flows when *On-Off* traffic is generated by node A with the same characteristics as the *On-Off* traffic in Section (5.2.1). The finer QoS granularity of the proposed scheme is again evident. Due to the same reasoning as stated before, the power consumption for each scheme with *On-Off* arrivals is more than their uniform arrival distribution counterparts.

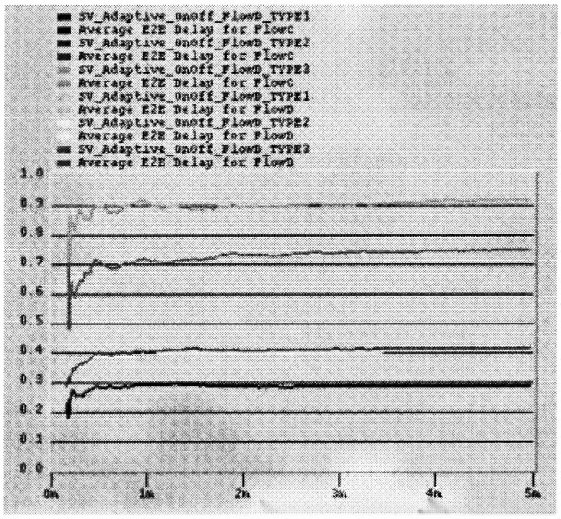

Figure 5.17 Average end-to-end delays of the three schemes for two different flows with *On-Off* arrival distribution.

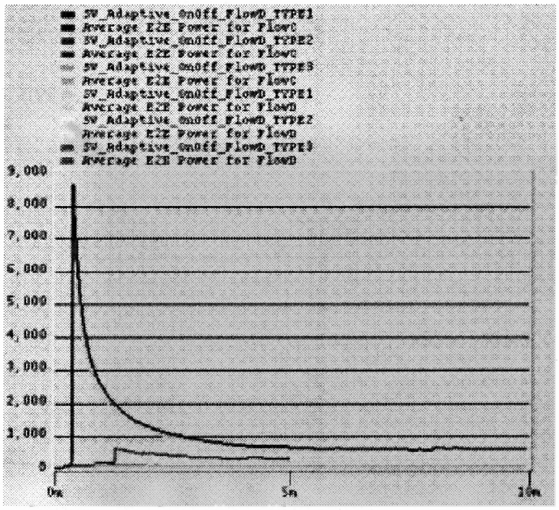

Figure 5.18 Average end-to-end power of the three schemes for two different flows with *On-Off* arrival distribution.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

A novel QoS provisioning architecture for wireless ad hoc networks is proposed in this thesis. An integrated scheme, which utilizes both link layer delay-bounded power efficient multi-user wireless scheduling and the network layer concept of *service vector* is introduced. It has been demonstrated through modeling and simulation that significant power savings as well as enhanced QoS granularity and service differentiation capability in wireless ad hoc networks can be achieved based on the proposed approach. The impact of various traffic arrival distributions as well as flows with different QoS requirements on the performance of the proposed strategy has also been investigated.

Due to the inefficiencies and implementation complexity of the optimal multi-user wireless scheduler, suboptimum scheduler which can operate only in AWGN channels has been utilized in this study. Extending this suboptimum scheduler to take fading into account would be of high practical and research importance in order to investigate the implications of fading on the performance of the *service vector* scheme. Furthermore, probing process can be utilized to gather additional information regarding the channel performance such as fading coefficients, which is usually unknown to the end user device.

Throughout this thesis, service-based pricing with constant unit price at each router was considered. Pricing schemes that encompass wireless channel performance can be taken into account in order to improve the overall network operational effectiveness and efficiency. In this way, the network may suggest its preference to the end user, which will affect the decision of the end user in determining the service vector.

REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," *RFC1633*, June 1994.

[2] S. Blake, D. Black, M. Calson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC2475, December 1998.

[3] J. Yang, J. Ye, S. Papavassiliou, and N. Ansari, "A Flexible and Distributed Architecture for Adaptive End-to-End QoS Provisioning in Next Generation Networks," IEEE Journal on Selected Areas in Communications, vol. 23, issue 2, pp. 321-333, February 2005.

[4] J. Yang, J. Ye, and S. Papavassiliou, "Enhancing End-to-End QoS Granularity in Diffserv Networks via Service Vector and Explicit Endpoint Admission Control," IEE Proceedings on Communications, vol. 151, no. 1, pp. 77-81, February 2004.

[5] *OPNET Modeler, http://www.opnet.com/products/modeler/*, 2004.

[6] J. Harju, P. Kivimaki, "Cooperation and Comparison of Diffserv and Intserv: Performance Measurements," 25th Annual IEEE Conference on Local Computer Networks, pp. 177-186, November 2000.

[7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource Reservation Protocol (RSVP)-Version 1 Functional Specification," RFC2205, September 1997.

[8] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A Framework for Integrated Services operation over Diffserv Networks," RFC2998, November 2000.

[9] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed Admission Control," IEEE Journal on Selected Areas in Communications, vol. 18, pp. 2617-2628, December 2000.

[10] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," ACM SIGCOMM Computer Communication Review, vol. 30, pp. 69-81, August 2000.

[11] A. J. Goldsmith, P. P. Varaiya, "Capacity of Fading Channels with Channel Side Information," IEEE Transactions on Information Theory, vol. 43, no. 6, pp. 1986-1992, November 1997.

[12] D. J. Goodman, J. Borras, N. B. Mandayam, R. Yates, "INFOSTATIONS: A New System Model for Data and Messaging Services," Proceedings of Vehicular Technology Conference, pp. 969-973, May 1997.

[13] B. Prabhakar, E.U. Biyikoglu, A. E. Gamal, "Energy-Efficient Transmission over A Wireless Link Via Lazy Packet Scheduling," Proceedings of IEEE INFOCOM, Anchorage, Alaska, April 2001.

[14] R. Berry, "Power and Delay Trade-offs in Fading Channels," Ph.D. thesis, Massachusetts Institute of Technology, June 2000.

[15] B. E. Collins, R. L. Cruz, "Transmission Policies for Time Varying Channels with Average Delay Constraints," Proceedings of Allerton International Conference on Communication, Control and Computing, pp. 709-717, September 1999.

[16] X. Liu, A. J. Goldsmith, "Optimal Power Allocation over Fading Channels with Stringent Delay Constraints," IEEE International Conference on Communications, vol. 3, pp. 1413-1418, May 2002.

[17] D. Rajan, A. Sabharwal, and B. Zhang, "Transmission Policies for Bursty Traffic Sources on Wireless Channels," 35th Annual Conference on Information Sciences and Systems, Baltimore, March 2001.

[18] H. Wang, N. B. Mandayam, "A Simple Packet Transmission Scheme for Wireless Data over Fading Channels," IEEE Transactions on Communications, vol. 52, issue 7, pp. 1055-1059, July 2004.

[19] C. E. Shannon, "A Mathematical Theory of Communication," Bell Systems Technical Journal, pp. 379-423, July 1948.

[20] S. Hanly, D. Tse, "Multi-access Fading Channels-Part II: Delay-Limited Capacities," IEEE Transactions on Information Theory, vol. 44, no. 7, pp. 2816-2831, November 1998.

[21] R. A. Berry, R.G. Gallager, "Communication over Fading Channels with Delay Constraints," IEEE Transactions on Information Theory, vol. 48, no. 5, pp. 1135-1149, May 2002.

[22] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, P. Whiting, "CDMA Data QoS Scheduling on the Forward Link with Variable Channel Conditions", Technical Report, Bell Laboratories, Lucent Technologies, April 2000.

[23] N. Joshi, S. R. Kadaba, S. Patel, G. S. Sundaram, "Downlink Scheduling in CDMA Data Networks," Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, August 2000.

[24] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, R. Vijayakumar, "Providing Quality of Service over A Shared Wireless Link," IEEE Communications Magazine, vol. 39, no. 2, pp. 150-154, February 2001.

[25] T. Nandagopal, T. Kim, X. Gao, V. Bhargavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, August 2000.

[26] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E. Knightly, "Distributed Multihop Scheduling and Medium Access with Delay and Throughput Constraints," Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, July 2001.

[27] D. Rajan, A. Sabharwal, B. Aazhang, "Delay-Bounded Packet Scheduling of Bursty Traffic over Wireless Channels," IEEE Transactions on Information Theory, vol. 50, no. 1, pp. 125-144, January 2004.

[28] D. Rajan, "Power Efficient Transmission Policies for Multimedia Traffic over Wireless Channels," Ph.D. thesis, Rice University, April 2002.