Expressing Application and Network Adaptivity: Time Variations and Adaptation Paths

By

Steven J. Bauer

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

February 2001

Author____
Department of Electrical Engineering and Computer Science
December 13, 1999

Certified by______________________________________
John Wroclawski
Research Scientist, Laboratory for Computer Science
Thesis Supervisor

Accepted by______________________________________
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Expressing Application and Network Adaptivity: Time Variations and Adaptation Paths

by

Steven J. Bauer

Submitted to the
Department of Electrical Engineering and Computer Science
on February 1, 2001, in partial fulfillment of the
requirements for the Degree of Master of Science in
Electrical Engineering and Computer Science

Abstract

Existing wireless networks provide a wide variety of service capabilities. Due to the inherent nature of wireless transmissions, these services are often characterized by high error rates, variable bandwidths and delays, and unpredictable interruptions. Users and applications are somewhat adaptive in their ability to handle these variable service conditions. However applications are not completely flexible nor does the user perceived quality vary in uniform fashion with the changes in network service. By characterizing flexibility, network service variations and application behaviors can be correlated to improve the QoS provided. To this end, this thesis argues that two new concepts, adaptation paths and time constraints, are important. Adaptation paths specify the ways in which network services and traffic can or do change with time. Time constraints capture aspects of QoS requirements related to time. In particular, two time constraints are introduced. First, a Discernible Service Time (DST) captures the duration for which a level of service must or will be provided before it is changed. Second, Interrupt Time (IT) captures durations for which a particular service may be interrupted for whatever reason. To demonstrate the utility of theses constructs this thesis provides a number of examples for how these extensions can be employed in wireless networks to improve QoS.

Thesis Supervisor: John Wroclawski
Title: Research Scientist, Laboratory for Computer Science

Acknowledgements

I would like to thank everyone who has helped me, provided advice, or lent support as I have worked on my Master's Thesis. Most particularly I would like to thank Anne and my parents. Many thanks go to my advisor John Wroclawski for showing me what a thesis should actually look like and teaching me a great deal about the research process. For advice their advice and friendship I want to acknowledge Jo and Chuck.

1 Introduction

Applications that rely upon communication over wireless networks are becoming increasingly prevalent. Initial wireless applications have made fairly low demands upon network resources. Users check their email, view simple web pages, or engage in similar low intensity network activities. Applications and users are fairly tolerant of changes in the network service so if quality drops below an acceptable threshold they usually are willing to retry their request later.

Increasingly there is a need for applications with more intense network service requirements. Both streaming audio and video applications benefit from additional wireless service capabilities. Users require more robust services as wireless networks are used to support more critical applications. As more users start competing for the shared wireless channels, the capacity available to all users is diminished and the variability of the service increases. Coupled with newer wireless users who are more accustomed to higher speed and more reliable wired network connections, demands for improved wireless service capabilities increase. Numerous research groups and standards bodies are addressing these issues for wireless networks (e.g. [IEEE802.11] or [VCM00]).

The service that a wireless network can provide to users varies widely as load adjusts and as environmental factors change. While this is true of wired networks, the time scales on which this variability occurs is far smaller for wireless networks and in a pattern that is far more disconcerting to the user experience. The variability of wireless service results from traditional link layer impairments such as fast fading, multipath and environmental interferences, coupled with the dynamic congestion of numerous users competing for channel access. Link layer impairments have very little effect in wired networks. In wired networks, resources are not as scare. Indeed, wireless networks always will lag behind the quality of service capabilities of wired networks [CG97].

Even more challenging to application designers is the fact that different wireless technologies provide diverse capabilities. The problem becomes particularly challenging if users can dynamically choose between different types of wireless networks. In the future, users may be able to choose their wireless network service provider on a continuous basis as they balance service requirements with costs [CW00]. Users may dynamically switch between wireless networks as the move through different coverage areas. Inside a building they might use higher capacity 802.11 systems while outside they would rely upon a CDPD system or another wide area wireless network. In each case multiple service providers likely would compete for a user's traffic. Thus, users and applications are faced with the problem of identifying wireless networks that can support their application requirements most effectively in addition to balancing tradeoffs between service quality and cost.

These factors combine to considerably affect the ability of programmers to design applications that provide an acceptable user experience over wireless networks with widely varying capabilities. Wireless applications can be designed with the ability to adapt to some degree of network service variance, thus hiding from the user the underlying instability of network service. Users themselves also can tolerate some variability in application experience. However both applications and users have limits as to how much variability they can endure. Even with the adaptability of applications the quality of the user experience degrades as the service provided varies.

The goal of this thesis is to express the adaptive characteristics of application and network behaviors in a simple and general manner. It is important to emphasize that the goal is to describe both application *and* network adaptive behaviors. Two parameter sets are proposed for capturing this adaptability: adaptation paths and time constraints. Adaptation paths specify the ways in which network services and traffic can or do change with time. We propose two specific time constraints related to dual aspects of QoS requirements. First, a Discernible Service Time (DST) captures the duration for which a level of service must or will be provided before it can be adjusted. Second, Interrupt Time (IT) captures durations for which interruptions in service may occur.

Adaptation paths and time variations can be used to specify *requested* or *actual* behaviors. Descriptions of requested behaviors indicate the manner in which the network or application should behave. Descriptions of actual behaviors indicate the manner in which the network or application will behave. Either can be applicable to applications or networks. The additional information provides an increased predictability of application and network behaviors as well as allows the network to have additional, though constrained, flexibility in satisfying application service requirements.

Specific applications of these ideas to networks include improving scheduling algorithms, modifying handoff procedures between wireless or cellular base stations, optimizing route selections, or adapting link layer characteristics. Similarly applications can exploit the increased predictability of networks to tune application behaviors. Examples include more appropriately selecting network QoS levels and more gracefully changing application behaviors as network service varies.

1.1 Related Work

This work is part of a larger ongoing MIT research project aimed at providing a QoS framework for future wireless environments. A goal of the larger project is to improve the quality of service of wireless applications by exploiting application profiles to improve scheduling algorithms and network adaptation functions. A second goal is to provide a mechanism for translating between QoS specifications and network service capabilities. This will provide a means to perform network service selection in environments where multiple service options exist.

Many previous research projects have addressed quality of service issues. The following sections review other QoS models and frameworks. Other techniques for improving network QoS capabilities by modifying elements of the network stack are reviewed. Most of this related work is complementary to the ideas presented in this thesis. Where appropriate our ideas could be incorporated into their research projects. Similarly our ongoing research project at MIT will likely leverage many of their results.

1.1.1 QoS Models

This section reviews relevant QoS models and discusses their contributions to how application requirements are characterized. Specifically models that capture aspects of application behaviors that are related to time and adaptation paths are presented.

Previous research projects have identified the importance of time scales upon which applications can adapt [KATZ95] [CAM97]. Of particular relevance is the Mobiware project [CAM97]. The time scales introduced in this research project are presented as four policy options: fast adaptation, smooth adaptation, handoff adaptation, and never adapting. Respectively, they represent the ability to adapt service continuously, in a damped fashion, only at times when handoffs between base stations occur, and finally to maintain the reservation at its initial level and never adapt. The time scale upon which an application adapts in their model is independent of the service being provided. It is instead a characteristic of the application.

Another component of the Mobiware QoS model is adaptation rates. In their examples they discuss adaptation rates in the context of bandwidths. For applications that can adapt, a rate is specified which indicates the maximum amount of change in bandwidth acceptable for one adaptation period. The adaptation rate is independent of the level of service being provided and is identical regardless of whether network service is improving or degrading. The idea though is similar to the adaptation paths we propose in that it expresses information about how network service must change with time.

Many researchers have identified the importance of the graceful degradation of service [SIN]. One common way of addressing this issue is by providing layers of service [RHE99], [GCFH94]. These models target streaming audio or video applications whose traffic flows can naturally be decomposed into various layers of quality [RHE99], [GAN98]. As congestion on the network varies or channel capacity changes, models drop or add layers progressively.

Numerous QoS models capture aspects of application adaptivity through specifying ranges of acceptable service levels [SSB99], [LC98]. Ranges define the bounds on QoS parameters for which an application requires service. These identify a limited amount of flexibility in the services that they utilize. Most examples of such QoS models use closed ranges, but conceptually it is easy to imagine that a QoS model could specify open ranges. Movement of the parameter within a range is typically unconstrained.

1.1.2 QoS Frameworks

Various frameworks have been proposed for improving the quality of service capabilities of networks [CAM97], [ZBS97], and [CFK98]. The frameworks implement a mapping between application requirements and network services and define admission control or resource reservation protocols. Many define flow scheduling, shaping or control algorithms. Finally the frameworks address aspects of flow monitoring, QoS alerts, and QoS maintenance.

The frameworks apply different approaches in specifying and translating applications needs into network control parameters. The translation is preformed by the application designer and embedded into the application itself or the translation is performed by a separate component of the system. The end result of the translation is a set of network control parameters. It is assumed that the characteristics of the network services and their relation to the application requirements are well understood by the translation entity. The translation itself is accomplished through pattern matching techniques.

QoS frameworks exploit aspects of application flexibility [BRS00]. QoS is negotiated initially and subsequent adaptation does not occur unless application requirements change or the network services can no longer support the original QoS. The QoS specifications employed to perform the selection most often include ranges of acceptable values and definitions of acceptable combinations of network services. Some frameworks are designed to support the negotiation of reservations for multiple elements including reservation of operating system resources.

A degree of complexity in the translation of application requirements to network services is added when "cost" is taken into account. Various frameworks have different notions of what these costs are, including but not limited to actual monetary costs charged by the network and costs defined in terms of the amount of network resources consumed. Value decisions are made based upon specifications of the "utility" of the requested network services. Utility is defined differently in competing frameworks.

1.1.3 Improving the Network Stack

Another approach to improving the quality of service capabilities of networks is through modification of various layers of the network stack. Techniques include modifying the link, network, and application layer. These are complementary to the work presented in this thesis and could be employed to further improve the application and network performance.

Link layer techniques for improving service quality improve the predictability, efficiency, and fairness of channel access. Medium access control strategies have been proposed to provide fair access to channel [ECK00]. Fair access techniques have been devised incorporating models of fairness based upon the utility of flows [BCL98]. Link layer reservation schemes provide predictable service under certain network models [SBM]. Slot swapping techniques allow certain predictable errors to be avoided [LBS97]. Techniques for diminishing error rates experienced include dynamically adjusting transmission power and dynamically changing the encoding schemes [LS98].

At the network layer techniques for improving the network service model have been suggested. Employing explicit reservations schemes such as RSVP have been proposed. Similarly using differentiated services within wireless networks offers an improved quality of service capability [Diffserv]. Other network level techniques for improving the service capabilities include decreasing congestion through various queue management strategies such as RED [RED]. Different scheduling algorithms can be employed to improve fairness such as weighted fair queuing [DKS90] or scheduling based upon the utility of flows [BCL98].

At the highest layer, applications cope with a range of network capabilities by adapting their behavior. Applications choose alternate representations of objects, download objects from alternate locations, postpone communications until more convenient times, or vary the amount of effort put into satisfying a flow's requirements. TCP adapts the amount of data it sends into the network in response to packet losses and measurements of round trip time. Streaming audio and video algorithms adapt the amount of buffering used and the quality and frequency of the output in response to measurements of network congestion.

1.1.4 Overprovisioning Resources

Though increases in network resources will generally benefit the quality of network service, wireless network resources will always lag behind the capacity of wired networks. Attempts to overprovision wireless network resources probably will not provide a sufficient solution. User expectations and requirements for wired and wireless environments will never diverge far enough that wireless capacity will satisfy user needs. If overprovisioning could solve the quality of service requirements for applications on one particular network, the problem of selecting between different physical networks and selecting appropriate services (taking cost vs. quality or other tradeoffs into consideration) would still remain.

1.2 Goals and Contributions

The goals of this thesis are to characterize the adaptive capabilities of networks and applications in a simple and general manner and demonstrate how this information can be exploited to improve the QoS capabilities of wireless networks. We were motivated by the need to improve the quality of service that wireless applications achieve.

In addressing our goals this thesis makes the following contributions. We identify the importance of new time constraints that control when adaptation occurs and present adaptation paths through which application and network behavior changes. These concepts are important because of the inherent variations in wireless network services and application adaptive behaviors. The thesis demonstrates how these two concepts can be expressed using two QoS parameter sets: time variations and adaptation paths. We then present examples of how both these notions can be used to improve networks and applications.

A second contribution is a demonstration of the practicality of applying adaptation paths and time constraints. We design a polling algorithm for improving the quality of service capabilities of the Point Control Function (PCF) mode in 802.11 wireless networks. Our polling algorithm improves upon the conventional round robin polling strategy by enabling polling intervals to be appropriately adapted in the face of location dependent channel errors and varying channel loads. Using our strategy flows are both gracefully degraded and upgraded according to application specification adaptation paths and adaptation occurs according to flow specific time constraints.

1.3 Roadmap

The first chapter of this thesis has provided the motivation and design goals that guided our work. Chapter 2 discusses application and network adaptivity. Chapter 3 provides a description of the concept of time constraints and adaptation. Chapter 4 describes the polling algorithm and our evaluation strategy. Chapter 5 presents conclusions and future work.

2 Application and Network Behaviors

The purpose of this chapter is to discuss adaptive applications and networks. The chapter focuses on how and when adaptation occurs, why it occurs, and who drives or controls the adaptation. The reason for examining these behaviors is to learn about the relationship between application activities and network services. Network behaviors drive application adaptations as well as application behaviors driving network adaptations.

2.1 Adaptive Applications

Applications can be classified as either non-adaptive or adaptive. Non-adaptive applications require a minimum level of performance from the network infrastructure and do not perform better if more resources are provided. Adaptive applications on the other hand adjust to the capabilities of the infrastructure and make varying demands of network resources as user behaviors change. As these applications acquire additional network resources their performance improves.

Adaptive applications cope with a wide range of network behaviors by adjusting their behavior accordingly. The ability to adapt improves the quality of service provided by an application to users. If additional network resources become available adaptive applications can use them to improve the content delivered. Conversely as network resources become unavailable either because of additional load on the system or varying system resource capacity, applications adapt so that useful work can still be performed with the remaining resources.

Application adaptations can broadly be grouped into three categories: content adaptations, adaptive algorithms, and user motivated adaptations. Within each category there are variations on the time scales at which adaptations can occur. Similarly the effects of the adaptation can be partially or readily apparent to the application user. The manner in which the service levels vary also is important.

2.1.1 Content Adaptations

Often applications have a choice of how data is represented. By varying the quality of the transmitted data or its fidelity to the original source, applications response time or other important performance measure can be maintained around a target level. The number of options and the ways in which the options are presented varies by application and network as does the mechanisms by which a particular representation is selected.

Different data representations are the product of various encoding levels and techniques. Video can be compressed using any of the different MPEG standards producing a range of data rates. Audio can be similarly compressed to different bit rates. RealMedia files can be composed using six different bit rates appropriate for anything from 28K dial-up modems to Cable Modems [Stemm]. Similarly for Windows Media files have ten individual bit rates for video and eight individual bit rates for audio [Stemm]. The encoding method itself can result in a variable bit rate stream. Certain techniques remove or significantly compress silent periods. Encodings used for video streams result in lower bit rates if the amount of motion in the video is low.

When applications select from among a multitude of data representations, they choose a representation by balancing quality against end-to-end download time, cost, or other relevant performance measure. In some instances the selection of the data representation must be made initially and cannot be changed as the download progresses. Web browsers that employ the HTTP's Transparent Content Negotiation [RFC-2295] mechanism are one example. They provide a way for clients to select between multiple representations of web objects on a server using one HTTP request.

Other applications accommodate data representations that dynamically change as the download progresses. An example of such an application is the RealMedia player which measures the packet loss rate of the transfer in progress. If the loss rate is high due to limitations of the player, network path, or transmitting server, the server is instructed to switch mid-stream to a lower bit rate representation [Stemm]. Another mechanism for dynamically matching the available capacity in the network to an applications needs is through the use of a proxy that acts as a transcoding service. A client receiving a multimedia stream uses a transcoding service (such as Video Gateway [AMZ95]) to dynamically change the data rate of the stream. These services change the data rate according to client requests in real-time.

The policies by which applications select the appropriate content representation can be either a function of the application, which dynamically tests the network performance, or can be user specified. User preferences can be captured explicitly in configuration options or may be dynamically determined through user input at selection time. Whether users have enough information to make an informed choice about appropriate representation depends upon the application and granularity of data representation options.

2.1.2 Adaptive Algorithms

Applications can cope with variations in network performance by employing algorithms that adapt by measuring or responding to changes in the state of the network. TCP adapts the amount of data it sends into the network in response to packet losses and measurements of round trip time. Clients participating in a multicast sessions using RTCP change the rate at which they send Receiver Reports in response to the number of participants in the session. Other adaptive algorithms rely upon reports from the network components upon system capacity [LC98].

2.1.3 User adaptations

The load that an application places on a network is often highly correlated to user behaviors. In a general sense, users effect what work is performed and when it is performed. A user's personal threshold for how long a download should take results in them retrying a web request multiple times or aborting a transfer midway. Similarly the content that they generate or request for transfer over the web is highly variable.

Users application behaviors can be guided by information presented about the state of the network. Web browsers for instance have been modified to include hyperlinks indicating the expected transfer time of the linked object. Typically though it is difficult to characterize a users behavior. This does not imply that the load an application user can generate is unlimited. Users are limited by the structure of the application being used. For instance NetMeeting users can establish an audio connection with only one other person at a time and the data rate of the connection has a maximum upward bound.

2.2 Network Behaviors

This section describes the ways in which networks adapt as they provide service to applications. Network adaptations are the result of changes in the load on the network and efforts at improving or maintaining the connectivity of the network. Networks also adapt as they make changes that improve the efficiency or balance of load on the network. Network adaptations are important since they have a direct impact on the services provided to client applications. This entails the amount of resources that can be made available to a particular client and their quality.

Wireless LANs typically operate in very strong multipath fading channels that can change their characteristics in a very short time or in a very short distance. Such fading channels may make communication unreliable and may result in capture that leads to unfair access. According to channel measurements and modeling of wireless networks the channel conditions may change significantly within ten to twenty millisecond duration or any movement of one foot distance [VCM00].

In cellular and wireless networks, adaptations include handoff policies that dictate when clients are shifted to alternative base stations. These decisions are based either upon a measured metric of a connection's performance [LC98] or are the result of attempting to balance the load between base stations. Other predictable events that will cause a network level adaptation include exhausting a transmitting stations battery power and some movement of nodes in a wireless networks.

Some networks distribute content in anticipation of user requests. Numerous cache-based schemes exist for locating content closer to the user and distributing load on a system. The manner by which content is distributed and by which caches are selected varies. In some systems the application client selects the most appropriate cache, in others the decision is left to the network and conveyed through something like the DNS [Karger].

3 Adaptation Paths and Time Constraints

The goal of this chapter is to describe the central observations that this thesis makes regarding ways in which adaptive application and network behaviors can be expressed. Descriptions of behaviors can be either declarative, in that they define ways in which applications or networks do behave, or prescriptive, in that they define ways in which applications and networks would like behaviors to occur. Our observations apply to both declarative and prescriptive descriptions of adaptive behaviors.

Declarative Application: describe the way in which applications do behave	***Declarative Network***: describe the way networks do behave
Prescriptive Application: describe the way applications would like the network to behave	***Prescriptive Network***: describe the way networks would like applications to behave

Application and network behaviors can be characterized using many different metrics. For applications, existing QoS models provide a range of descriptive possibilities. Metrics include bandwidth, delay parameters, error rates, and jitter. Network services are characterized in terms of available channel capacity, maximum delay, and expected error rates (i.e. bandwidth, delay, jitter, error). Other metrics for describing networks include handoff frequencies, path stability, and mobility or location information.

While an application behavior or network service as measured by some metric may vary over large extents of the metrics range, the manner in which this variation occurs is often not unpredictable. The bandwidth that an application consumes may range from 0 Mbps to the total channel capacity. The transition from the lowest to highest bandwidth may occur along a predictable path and at a predictable adaptation rate. Completely random movement of a metric about its range is seldom tolerated for either application or network behaviors. Studies have found that users find video to be highly disconcerting if picture quality varies too frequently or widely as a result of channel capacity oscillations.

The predictability of application and network behaviors results from fundamental constraints of design or policy. The algorithms employed or the system architecture limits the variations in behavior. Therefore it becomes possible to characterize the adaptation paths and the characteristics of the time domain of adaptations. It is these predictable patterns of changes in behavior that are explored in the following sections.

3.1 Adaptation Paths

The importance of the graceful degradation of services has been identified previously in the context of reservation systems. Providing for graceful degradation indicates both that service levels can change and that they should change in a prescribed graceful fashion. This allows the level at which a reservation was initially supported at to be adapted to an alternative lower quality level if the initial service cannot be maintained. Further degradations to lower quality levels may follow. The dual of graceful degradation of service occurs when service quality can improve as extra capacity becomes available. In these cases higher quality layers are specified which provide additional utility to the application.

Understanding the ways in which movement occurs between these layers is important. Multi-layer video applications for example add a discrete amount of new load for every additional layer of video. Video layers usually are added sequentially however more than one layer of video can be removed at a time as service degrades abruptly. A depiction of the changes in service from one layer of video to another is shown in Figure 1. These are simple linear adaptation paths.

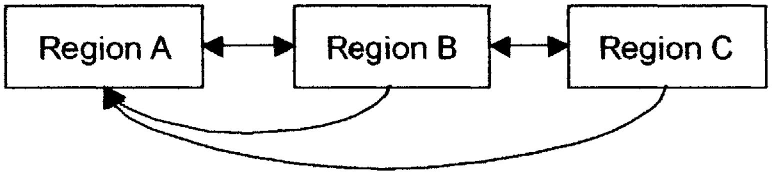

Figure 1: Linear adaptation paths

Multiple options for adaptation often exist where service can transition from one operating region to a constrained set of other regions. An application operating at 1

Mbps might be able to handle upgrading to either 2 Mbps or 3 Mbps. In Figure 2 there can be a transition from Region A to either Region B or Region C.

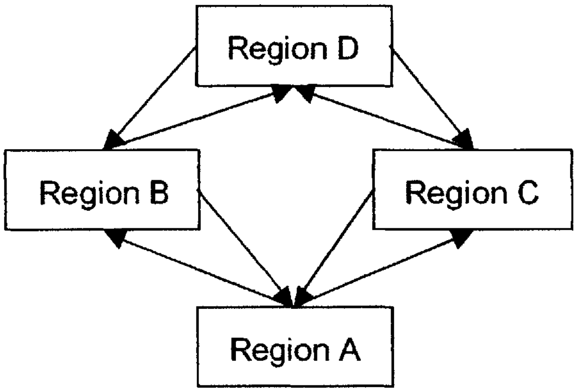

Figure 2: Multipath Adaptations

Adaptation paths are not always symmetric. Just because an application can transition from one operating region to another does not necessarily entail that it able to transition in the other direction. In Figure 3 the adaptation paths indicate that the adaptation can occur between Region C and Region D but not the other way. This may correspond to applications that can increase at a predictable rate but stop abruptly and have to be resumed gradually.

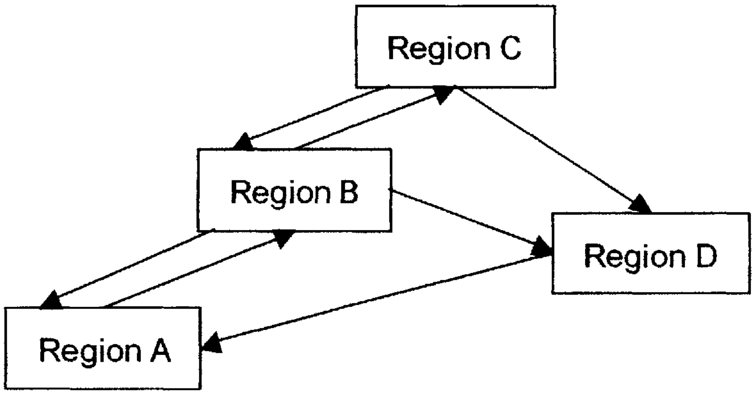

Figure 3: Directed Adaptations

3.2 Time Constraints

An important aspect related to adaptation paths is the time scale over which application and network behaviors change. Previous research work has discussed the importance of adaptation time scales [CAM97]. This is a single parameter that specifies one time scale upon which an application can adapt and is not dependent upon the level of service being provided. Our work expands on this by considering additional ways in which time information can be useful.

A single adaptation time scale may be appropriate if it expresses the granularity at which additional capacity is probed for or discovered. End systems may send QoS reports at periodic intervals prompting service adaptations. Networks may, at a fixed period, reassign network resources as a form of load balancing. Other network or applications behaviors can be expressed by a single adaptation time including the rate at which routing updates are computed or the rate at which time intervals between when an application switches servers. Figure 4 represents the ability to adapt at a fixed interval (or continuously) independent of the level of service provided.

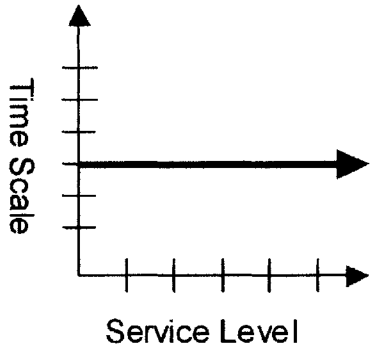

Figure 4: Single Adaptation Parameter

However the time scale upon which an application or network service can adapt may be dependent upon the level of service being provided. For instance a 4 Mbps service may only need to be provided to an application for 10 seconds before it can be changed while a 2 Mbps service may have to be provided for at least 30 seconds before it can be changed. This might reflect the need of the application to download or transmit a certain quantity of information. Figure 5 abstractly represents such an application.

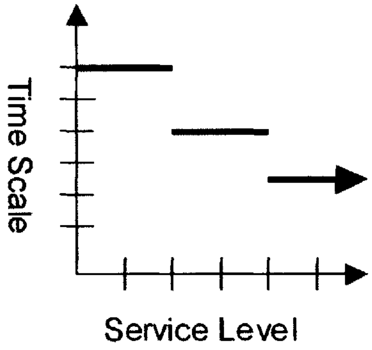

Figure 5: Service Level Dependant Adaptation Times

Another time constraint important to applications traffic and network service is the length of interrupts in service. Many applications are capable of tolerating limited durations in which service drops below a certain threshold. Similarly networks may be able to indicate that they are liable to interrupt service for certain durations. These would occur when a network consumes channel capacity through overhead traffic or when handoffs occur between base stations. Applications can tolerate interrupts potentially because they have certain buffer sizes. Interrupt times also may capture users level tolerances for interruptions in service. In Figure 6, Interrupt A may be of an acceptably short duration but Interrupt B may cost the application or user an unacceptable interruption in service.

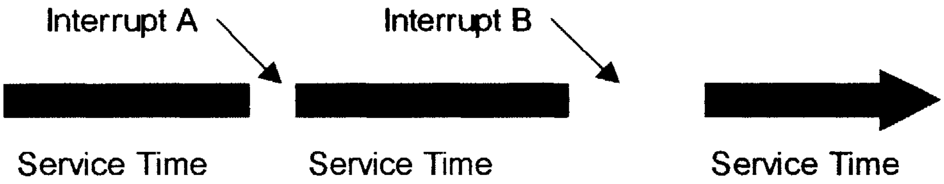

Figure 6: Service Interrupt Times

Acceptable interrupt durations may be an application constant or may be a function of the level of service being provided. For instance it may be acceptable to interrupt network service to a video application for five seconds if it is providing a lower level of video while it is only acceptable to interrupt a higher quality video stream for one-second durations. Just as there is a growing realization that it may be important to capture loss distributions when characterizing traffic flows it may be important to characterize interrupt intervals in terms of distributions. It may be important to be able to say that no more then two interrupts in service will occur in a certain duration of time or perhaps that if service is interrupted than it will not be interrupted again for a period of time.

3.3 Example Usage of Adaptation Paths and Time Constraints

This section presents brief examples of ways in which adaptation paths and time constraints can be used.

3.3.1 Prescriptive Application Profiles

Adaptation paths and time constraints can be used to extend resource reservation requests. An application time constraint can indicate a need to receive a particular service for at least a minimum duration of time. Similarly an application might specify that it could tolerate interruptions in service below some threshold time. If the maximum interrupt time for the network was longer then the requested maximum interrupt time then the reservation might be rejected.

Networks could also employ the extended resource reservation requests to provide tailored reservations. For instance the network might postpone accepting other flows until existing flows could be gracefully degraded. Extra effort might be expended to ensure that flows were serviced for at least the requested amount of service time. Interruptions in service, perhaps to redistribute client stations among base stations, might be avoided to satisfy an application that could not tolerate the interruption in service.

3.3.2 Declarative Application Profiles

Applications might inform the network of the manner in which they do behave. By describing the ways in which applications will behave, networks could tailor the services they provide. Typical traffic patterns could be conveyed to the network in the form of probable adaptation paths and time information. The network might be able to readjust its configuration if it knew that a particular user was likely to require additional services in the near future. If it was known that an application that suspended service would not resume service for a duration of time then the network could temporarily schedule other work.

3.3.3 Prescriptive Network Profiles

Networks could similarly indicate the way require applications to behave using adaptation paths and time constraints. Networks could advertise that they require traffic flows to conform to certain traffic specifications. These might indicate that upon receiving network feedback applications must adapt their traffic service levels within certain time scales and along certain paths. Networks could indicate that certain interrupts in service will happen for particular durations of time. These interrupts may be either related to environmental effects or network behaviors such as handoff times or overhead traffic. Networks could employ components which police application traffic flows and ensure that they do adopt the prescribed sending rates in the fashion dictated by the adaptation paths.

3.3.4 Declarative Network Profiles

A network might indicate to applications that the network service behaves in a certain fashion. Networks might for instance dictate that under load they will decrease service quality along certain paths to individual flows. The time scales upon which this adaptations occur could be specified. For instance periodic intervals over which the network performs load-balancing operations could be specified. Over the duration of time involved in interrupted a service to hand it off to a neighboring cell could be specified.

4 Design Example

This chapter presents the design for a polling algorithm that employs adaptation paths and time constraints. The objective of the polling algorithm is to maximize the number of supportable concurrent flows while maintaining flow specific service requirements. Our polling algorithm improves upon the conventional round robin polling strategy by enabling polling intervals to be appropriately adapted in the face of location dependent channel errors and varying channel loads. Using our strategy flows are both gracefully degraded and upgraded according to application specification adaptation paths and adaptation occurs according to flow specific adaptation time constraints.

The context for our proposed polling algorithm is an 802.11 Point Control Function (PCF) mode wireless network. This is an appropriate network for employing time constraints and adaptation paths since it is centrally administrated, designed for applications with better then best effort service requirements, and unconstrained by a specification in terms of polling algorithm or admission policies. As a practical matter 802.11 networks are widely deployed and increasingly will be relied upon to support critical applications.

This chapter is laid out as follows. The first section presents a brief background of the operating modes of the 802.11 Specification. The background is presented to familiarize the reader with the conventional functionality and limitations of 802.11 networks. The next section presents the system architecture. This contains a description of applications' flow profiles that define adaptation paths and time constraints as well as a description of the algorithms employed by both the polling and polled stations. The final section describes our proposed evaluation methodology.

4.1 802.11 Background

802.11 Specification defines two modes of operation for controlling channel access: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). Most commercially available 802.11 radios implement the Distributed Coordination Function. The Distributed Coordination Function mode channel access scheme is a carrier sense multiple access with collision avoidance (CSMA/CA) based approach. The Distributed Coordination Function mode is considered most appropriate for bulk data transfers and low load environment since it presents potentially large and an unbounded channel access delays. The 802.11 Specification also defines the Point Control Function mode designed for providing "connection-oriented" services to real-time applications. This mode provides delay guarantees through a centralized node that controls channel access. Complete details of the two modes of channel access can be found in the 802.11 Specification [IEEE802.11].

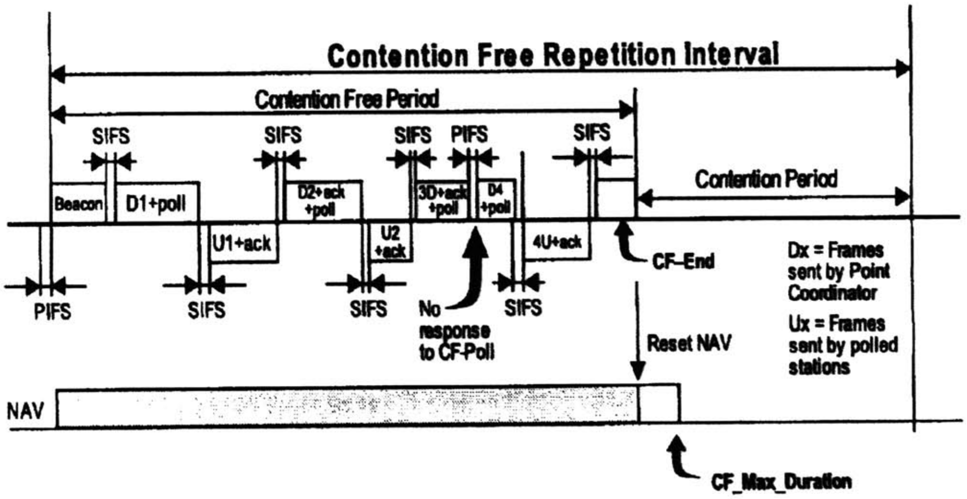

Figure 7: PCF Framing Structure [802.11]

Channel access in the Point Control Function mode of operation is divided into two time periods, the Contention Period (CP) and the Contention Free Period (CFP). A beacon frame sent by the Point Controller (PC) signals the beginning of a contention free period. During the contention free period the point controller is free to transmit any of its queued packets to a station in the network. The point controller polls any associated station to determine if it has packets to transmit. No station can transmit during the contention free period unless the pointer controller has polled it. The maximum duration of the contention free period is indicated in the initial beacon marking the beginning of the contention free period. The point controller can end the contention free period early with the transmission of a CF_End frame. During the contention period all nodes compete for channel access using the distributed coordination function methodology.

The stations polled during the Contention Free Period are determined by a polling list maintained at the Point Controller. The 802.11 Specification does not mandate a mechanism for creating or maintaining the polling list at the Point Controller. This is considered out of scope for the standard so manufacturers are free to implement one of their choosing. The method by which an access point utilizes the polling list to perform polling of station associated with it is similarly undefined. A simple polling strategy commonly employed is a round robin policy that polls each node on the list sequentially [VCM00]. The round robin policy can poll each node on the list exactly once, less then once, or more then once per contention free period. The polling policy employed depends on the type of service that the point controller is attempting to provide.

A drawback of the point control function is that it is not particularly scalable. The point controller needs to control media access by polling all stations, which can be ineffective in large networks. In a distributed coordination function network configuration multiple nodes can transmit at the same time if they are not in the same coverage area. However there is currently no means for guaranteeing channel access times in the distributed coordination function. Therefore for services that require delay bounds, the point control function mode is most appropriate given the current standard.

4.2 System Architecture

Our system architecture is composed of an 802.11 wireless network consisting of mobile stations and access points whose interaction is governed by point control function mode of the 802.11 Specification. Our architecture includes an *Admission Control Plane* that specifies an interface for admitting flows to the polling list. The *Adaptation Plane* consists of an interface and algorithms for controlling flow adaptation. The *Polling Plane* is the basic polling procedures specified by 802.11 modified slightly so that adaptation information can be included in the poll and ACK packets. The architecture is depicted in the following figure.

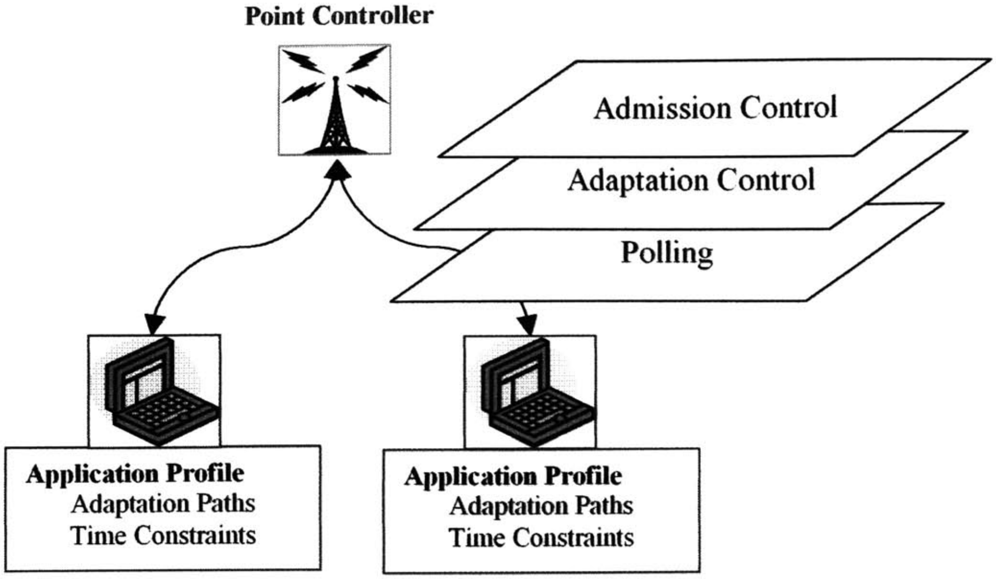

Figure 7: System Architecture

4.2.1 Mobile Nodes and Applications

This section describes the behavior of nodes and their associated applications. As indicated the network functions according to the 802.11 Point Control Function. The main components we specify are the ways in which applications define their service requirements and the algorithms that employ these specifications. Designers specify applications flow requirements using *application profiles*. These are communicated to the point controller during the admission protocol.

We assume that only one application is "active" per station at a time. This implies that a user is concerned only with the quality of service that the active application is receiving. This may correspond to the application that has the user focus on the desktop or may be indicated through some other mechanism. We justify this simplifying assumption through the observation that users often do focus on one application at a time particularly ones which require service guarantees. Streaming audio and video applications for example usually occupy a users attention entirely. Users may do multiple tasks at once such as listening to an audio stream and using email, but only one requires service guarantees. This assumption will no longer be required once we determine how application profiles can be meaningful combined to describe node adaptation profiles.

4.2.1.1 Application Profile

An *Application Profile* defines the service requirements for an application. It is a prescriptive description of the service needs of an application and can be used to describe either uplink or downlink reservations. Application profiles consist of sets of 6-tuples each containing the information presented in the figure below. In addition to the operating regions defined by each 6-tuple, an identifier names each profile. By naming a profile, a point controller can cache the commonly used profiles and avoid the overhead required to transmit them during the admission protocol.

Operating Region 6-Tuple:

- *Identifier*: name defining the current operating region
- *Bucket Rate*: rate at packets are generated
- *Maximum Packet_Size*: the maximum size of packets
- *Discernible_Service_Time*: the duration of time for which service region should be maintained before it can be productively or safely changed.
- *Interrupt_Time*: the maximum duration of time for which the current operating region can sustain a service interruption
- *Adaptation Paths*: the set of acceptable next operating regions

Figure 8: Operating Region 6-Tuple

The number of operating regions in each profile varies according the service requirements of an application. However we expect common applications will require less then ten operating regions. Appropriate encodings will be selected to minimize the costs of transmitting profiles to the point controller. Compression techniques could be applied if profiles became particularly large.

We have not addressed how to derive these profiles from application behaviors. Previous research projects have considered how QoS requirements can be captured for applications and we would leverage some of these results. We will have to extend this work to capture the time constraint information and adaptation paths that we include in our profiles. We note that application profiles do not necessarily have to be constructed by application designers or be integrated with an application itself. This enables applications to remain unmodified and allows them to exploit networks employing our modified polling algorithm.

4.2.1.2 Admission Protocol

An application profile is transmitted to a point controller as part of an admission protocol. The basic protocol is outlined in the figure below. The initial request packet is sent to the point controller during a contention free period. The request consists of a profile name and preferred operating region. The profile name can correspond to either a profile for a common application or be user selected. XML like namespaces are used to prevent identifier collisions. If the point controller has the named profile cached an admission decision can be made immediately. If the point controller does not have it cached a packet is sent indicating the profile should be transmitted. Transmission of a profile probably can be accomplished with one additional packet but it could be fragmented over multiple packets depending on its size. The preferred operating region is indicated to guide the point controller in determining which operating region to provide. Admission decisions and selection of the initial operating region is described in a later section.

Station		Point Controller
Request (Region i)	→	
	←	Transmit Profile or Accept (j) or Deny
Profile Part (1)	→	
	←	ACK (1)
...	...	...
Profile Part (N)	→	
	←	ACK (N)
	←	Admit (Region j) or Deny

Figure 9: Admission Protocol

Once a node has had an application admitted to the polling list the node will be either be polled during the contention free period or receive packets during the contention free period according to the service requests specified in the profile. Thus each node has to properly segregate its application flows so that those requiring service guarantees are the ones transmitted in response to the poll queries. Similarly for downlink traffic flows the point controller must have the ability to identify those flows requiring service guarantees from best effort traffic destined for the same station. Flow isolation can be accomplished through an appropriate marking mechanism.

4.2.1.3 Node Adaptation

As the service capacity varies due to environmental factors or varying loads on a network, nodes must be prepared to upgrade or downgrade their flows accordingly. We accomplish this through an explicit notification mechanism. A point controller notifies a node that its service changed through a new frame element, *Region Information*. This indicates to the application its new operating region defined by the application profile submitted during the admission phase. Adaptations are subject to the adaptation paths and time constraints specified in the profiles. By including a new element in the basic frame element, adaptation information can be sent with regular frames destined for a node.

Octets 2	Octets 2	Octets 8	Octets 8	Octets 8	Octets 2	Octets 7	Octets 2	Octets 0-2312	Octets 4
Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Region Info	Frame Body	CRC

Figure 10: MAC Frame format

Similarly, applications indicate their preferred operating regions using the *Region Information* frame element on messages they transmit to the point controller. Since applications preferences are dynamic these are not included in the application profile. Applications may preemptively notify the point controller that they can be downgraded to a lower operating region through information conveyed in this field. Applications also indicate upgrade preferences using this the Region Information field. If an application is satisfied with the current service level then Region Information can be excluded.

Region Information Field	
Packet Direction	
PC → STA	STA→PC
Current Region: Indicates to a station the level of service that it will now receive.	***Downgrade***: Indicates to point controller that the station has excess capacity and can be downgraded to the indicated operating region
	Region Preference: Indicates to point controller if the station is satisfied or would prefer upgraded service

Figure 11: Region Information

4.2.2 Point Controller

This section describes the behavior of a point controller that employs the application specified time constraint and adaptation path information to formulate a dynamic polling policy. The objective of the polling algorithm is to maximize the number of supportable concurrent flows while maintaining flow specific service requirements. The state maintained at the point controller consists of cached application profiles, channel capacity information, and node bandwidth allocations, and performance information. A summary of the state maintained is presented in the table below.

Point Controller State:	Purpose:
Application Profiles	Cached profiles are used to determine adaptation policy when a reservation must be upgraded or downgraded
Reservation Bandwidth Allocations	Set of bandwidth allocations for all accepted reservations
Flow State and Performance	Current state of all flows and statistics indicating the performance of each. Used to determine if a flow possibly should be upgraded or downgraded
Channel Capacity Information	Statistics regarding the current performance characteristics of the channel. Used to determine reservation capacity of the point control function.

Table 1: Point Controller State

A flow can be in one of four possible states, *stable*, *unstable never interrupted*, *unstable previously interrupted*, and *interrupted*. These states correspond to whether or not the flow has been serviced for the discernible service time specified in the application profile and whether it has been interrupted within the last discernible service time or is currently not receiving its appropriate service level. At the inception of a flows traffic it is in the *unstable never interrupted* state. The flow remains in this state until the specified discernible service time has passed. The point controller will attempt to service a flow at this level and will not adapt its reservation higher or lower until after the discernible service time has passed. Flows that are serviced for the discernible service time transition to *Stable Flows*.

If a flow is determined to be out of its requested operating region then it is classified as an *interrupted flow*. This occurs when the requested service level cannot be supported due to location dependent channel errors or because of a lack of application traffic. If an interrupted flow does not return to its operating region within the interrupt time then it is downgraded to a lower operating region.

If a flow is unstable and interrupted but returns to its operating region before the end of the maximum interrupt time then the flow is considered *Unstable Previously Interrupted Flow*. If a flow classified in this way is interrupted again it is immediately downgraded. This ensures that an operating region is interrupted only once if it cannot be supported for a discernible service time between interruptions. These transitions are diagrammed in Figure 13.

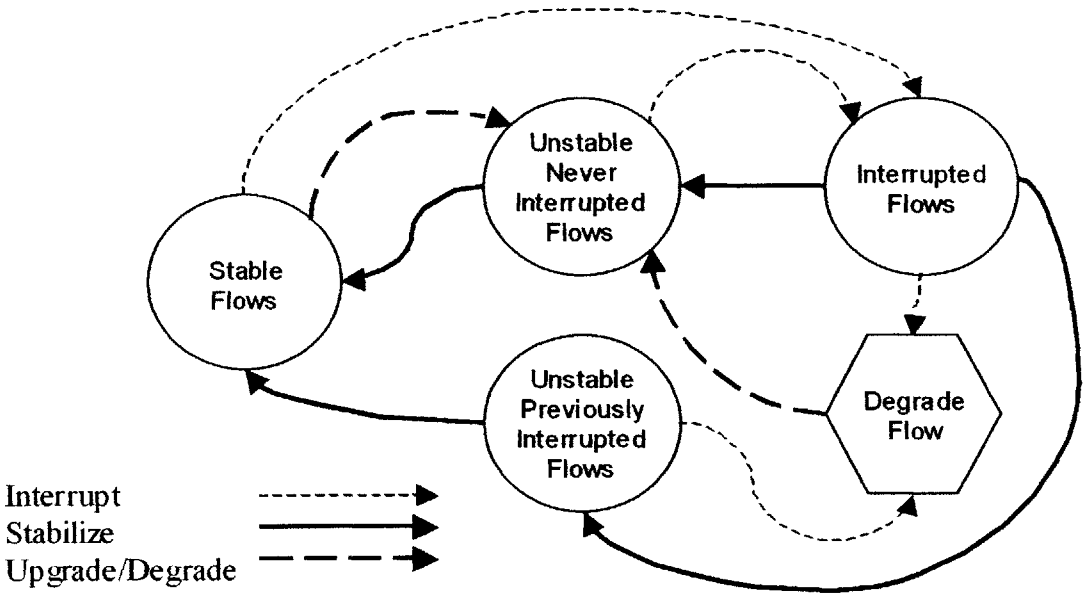

Figure 12: Flow State and Transitions

4.2.2.1 Admission Control Algorithm

Nodes requesting a reservation are accepted according to the admission policy at the point controller. Admission requests are denied if either the channel capacity is not available to support any of the requested operating regions or if the application profile does not conform to the requirements of the point controller. Policy based rejections of a reservation may include requests for too large of a discernible service time for instance. A maximum acceptable discernible service time may be specified to prevent a node from unfairly monopolizing resources. Other restrictions on application profiles could include maximum reservation bandwidths or a required minimum degree of adaptation options.

If the application profile associated with the request passes the profile-based restrictions, admissions decisions are then made based upon available capacity and the flexibility of existing flows to accommodate the new flow. Determining if the capacity exists to accept the reservation depends upon the operating regions specified in the profile, the total channel capacity, and the existing reservations and their associated adaptation restrictions. If the total bandwidth required to support the existing reservations plus the bandwidth requested to support the preferred operating region is less then the channel capacity then the flow can be accepted trivially.

```
If ($BW_{Preferred\_Operating\_Region}$ < $BW_{Free}$) Then {
    Notify(Accept);
    Allocate ($BW_{Preferred\_Operating\_Region}$);
    }
```

If the reservation request exceeds the amount of free bandwidth then the point controller runs a bandwidth allocation algorithm to determine if the application can be supported in any of the operating regions.

```
If ($BW_{Preferred\_Operating\_Region}$ > $BW_{Free}$) Then
    AcceptedBW = TestAllocate(Profile);
    If (not AcceptedBW) Then
        Notify (Deny);
        Else {
            Notify (Accept);
            Allocate (AcceptedBW);
            }
```

A flow that is accepted is initially classified as an unstable flow until the discernible service time has passed without interruption.

4.2.2.2 Bandwidth Allocation Algorithm

The bandwidth allocation algorithm is the core of the reservation system. It is used in determining if a reservation request can be satisfied and is used to rebalance bandwidth allocations between stable flows. The goal of the algorithm is to maximize the equality of bandwidth allocations among stable flows operating below their preferred service level. In other words, we attempt to achieve bandwidth fairness between all stable flows that could use additional capacity. We measure the fairness using the following formula [Jain] where each X_i represents the bandwidth allocated to a stable flow operating below its preferred operating level:

$$F(x) = \frac{(\sum_i X_i)^2}{N \times \sum_i (X_i^2)}$$

Distributing bandwidth fairly among flows is easy if each flow has a continuous spectrum of acceptable service levels. Each flow would receive an equal share:

$$BW_{Fair} = B_{Total} \;/\; \texttt{number\_flows}$$

If each flow can only productively utilize bandwidth at discrete levels then the allocation problem becomes more difficult. We allocate to each flow at a minimum the first service level below the BW_{Fair} level contained in each application profile. The appropriate service level is selected from the flows current adaptation path. In the diagram below requested service levels are depicted as solid horizontal lines. The BW_{Fair} level is depicted as the dashed horizontal line. Every node with a stable flow is guaranteed to have the first service level immediately below this fair allocation line.

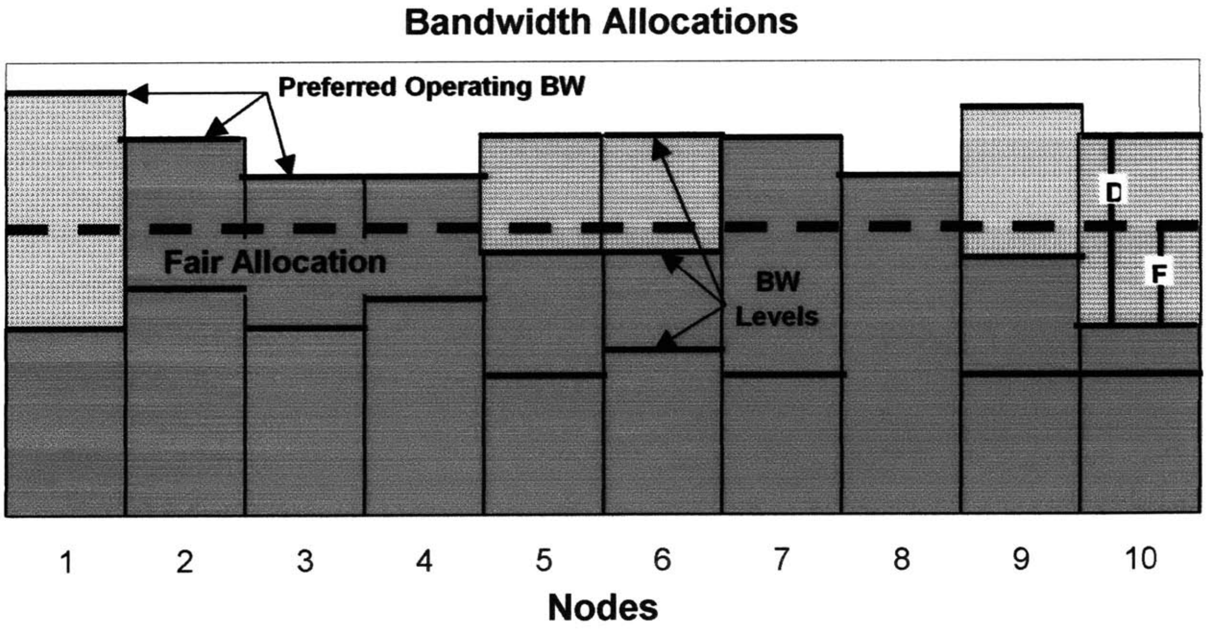

Figure 13: Bandwidth Allocations Maximizing Fairness

This allocation will leave access capacity unless each node can productively use exactly the fair bandwidth allocation. In distributing the remaining bandwidth we attempt to maximize the equality of the bandwidth allocations. For each node we determine the difference D between the bandwidth of the service level immediately below the fair allocation line and the service level immediately above the fair allocation. We then determine the difference F between the fair share allocation and the bandwidth level immediately below the fair share. We then iteratively assign bandwidth to nodes with the minimum F/D ratio. Ties are broken by satisfying flows with minimum D values first since these allocations consume fewer resources. Remaining ties are broken randomly.

This is a greedy algorithm for attempting to maximize fairness between flows with discrete service requirements. It does not necessarily achieve optimal fairness levels. Using this assignment method excess capacity might remain that cannot be assigned productively to any flows. We could have chosen to maximize the utilization of the channel link by reserved flows instead. However both optimizing the fairness and maximizing the channel utilization are computationally intense. Since the allocation algorithm is employed frequently to rebalance allocations among flows we ensure that we are not computationally bound using our approach.

```
$BW_{Total}$ : Total BW available in the channel
$BW_{Stable}$ : Total BW allocated to Stable flows
$BW_{Unstable}$: Total BW allocated to Unstable flows
$BW_{Interrupted}$: Total BW allocated to Interrupted flows
$BW_{Free}$ : Total BW unallocated
$BW_{Fair}$ : Fair Share Allocation
$\{BW_1 \ldots BW_n\}_{stable}$: Allocations to stable flows 1 .. n

AllocateBW {
   1. Calculate $BW_{Fair}$;
   2. Select initial assignment $\{BW_1,\ldots,BW_n\}_{Stable}$ such that every $BW_i$
      is the maximum BW below $BW_{Fair}$ on i's adaptation path;
   3. Calculate set $\{F/D_1,\ldots, F/D_n\}$ based on $\{BW_1,\ldots,BW_n\}_{stable}$ set
   4. $BW_{Free} = BW_{Total} - BW_{Unstable} - BW_{Interrupted} - BW_{Stable}$
   5. While ($BW_{Free} > 0$) and bandwidth allocated last iteration
         a. Select maximum ratio $\{F/D_1,\ldots, F/D_n\}$ breaking ties by
            selecting minimum D values and then by random
            selection
         b. If ($D_i < BW_{Free}$) {
                  Allocate BW $D_i$ to Node I
                  $BW_{Free}$ -= $D_i$
                  }
   6. Move flows that have changed service levels to the *Unstable*
      Flow Set
   }
```

4.2.2.3 Maintenance Algorithm:

The Maintenance Algorithm runs at periodic intervals to perform a number of tasks. First flows belonging to either of the unstable flow sets that have been serviced for their discernible service time are moved to the *Stable* flow set if they have a fair allocation of bandwidth. Unstable flows were previously exempt for being rebalanced by the bandwidth allocation algorithm. Therefore it is possible that these flows received more then their fair share allocation but could not be adapted since they had yet to be serviced for their discernible service time. Thus unstable flows may be degraded and remain unstable flows after a discernible service time period. Flows with longer discernible service time are not degraded as quickly as flows that are stable when channel capacity diminishes or load increases. However such flows also do not benefit from additional available capacity as quickly since they cannot be upgraded either.

The second function of the maintenance algorithm is to downgrade any flow that is a member of the *Interrupted* flow set for greater then the Interrupted Time specified in the flow's application profile. Third any flow that is a member of the *Interrupted* set that has been determined to have returned to its operating region is moved to the *Unstable Previously Interrupted Set.* Determining whether a flow has been interrupted or has returned to its operating region is made through examinations of the flows performance statistics. A flow is considered interrupted if it does not receive its allocated bandwidth over some interval of time. Forth any flow that is not using its currently allocated bandwidth, either from experiencing errors or a lack of application data, is moved to the *Interrupted* set.

```
Maintenance Algorithm {
  1. For every flow$_i$ in unstable never interrupted set
       a. If (ActualBW (flow$_i$) < AllocatedBW(flow$_i$)) then move
          flow$_i$ to interrupted set
       b. If (flow$_i$ service time > DST$_i$ ) then move flow$_i$ to
          stable set
  2. For every flow$_i$ in unstable previously interrupted set
       a. If (ActualBW (flow$_i$) < AllocatedBW(flow$_i$)) then
          downgrade flow to a next lower layer on adaptation
          path
       b. If (flow$_i$ service time > DST$_i$ ) then move flow$_i$ to
          stable set
  3. For every flow in interrupted flow set
       a. If (ActualBW (flow$_i$) < AllocatedBW(flow$_i$)) then
          downgrade flow to a next lower layer on adaptation
          path
       b. If (ActualBW (flow$_i$) = AllocatedBW(flow$_i$)) then move to
          unstable previously interrupted set
  4. Rerun the BW allocation algorithm since flows have
     potentially been downgraded or joined the stable set
```

4.3 Evaluation

This purpose of this section is to describe the methodology that we would employ to evaluate our polling algorithm. We have not completed an implementation so we describe the evaluation approach we will take once work has progressed. The primary evaluation will be through simulation. While an analytic approach would be better we have not yet gained enough experience with our system to understand its important behaviors. There also are not adequate analytic models of the applications we would like to evaluate. Simulation will allow us to experiment easily with a range of applications, profiles, and environments.

The objective of the polling algorithm is to maximize the number of supportable concurrent flows while maintaining flow specific service requirements. Our simulations therefore attempt to answer the following questions.

Simulation Questions:

1. How closely do service levels track available bandwidth?
2. As the number of nodes participating varies, how quickly does the network adapt the service levels of flows?
3. How does the fairness of bandwidth allocations between nodes compare?
4. Are adaptation paths and time constraints observed?
5. How much bandwidth is wasted in the transmission of late or out of profile packets?
6. How many applications (given specific profiles) can be supported in an environment?
7. How stable are bandwidth allocations as load and error rates vary?
8. How closely does the bandwidth provided to node track the bandwidth requested in an application profile (given specific profiles)?

Ideally applications will be developed that identify explicit adaptation paths and time constraints. In lieu of such applications we proceed by constructing application profiles for existing adaptive applications. Initial promising usage of adaptation paths and time constraints will be in applications that utilize adaptive video or audio codecs [MPEG], [ADPCM], [LPC]. For the purposes of evaluation an appropriate demonstrate application will be selected and modeled in simulation.

For our simulations we adopt the general models and metrics defined by the IEEE 802.11 working group in "Performance Metrics and Evaluation Criteria for the 802.11-QoS Simulation Platform" [IEEE802.11]. These models are defined for the OpNet based 802.11 simulation as modified by the IEEE 802.11 Simulation and ad hoc group. The group defines no channel error model so we adopt a 2-state continuous-time Markov chain to represent a burst error model [CWKS97] presented in Figure 9. The *good* state indicates that the channel is operating with very low bit error rates while the *bad* state indicates that the channel is operating in a fading condition with a higher error rate. The probability of transitions between the states are given by α and β. No forward error correction codes are used in this simulation so a frame is considered corrupted if it contains one or more errors.

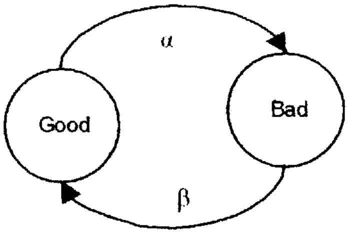

Figure 14: Markov chain burst error rate model

The environment we simulate would be a flat square topology. We are primarily limited here by models included with the simulators. Ns-2 only supplies a flat topology; OpNet does provide for more complex configurations. The benefits of using a move complex topology would be the ability to evaluate the performance under more realistic environments. We will explore this option after initial simulations have been completed.

The most direct comparison is to judge our polling algorithm against a point control function mode network employing a round robin polling strategy. The crucial element in this experiment is the selection of the admission algorithm to the polling list of the round robin PCF point controller however no standard admission policies exist. Viable options include accepting a maximum number of nodes to the polling list or deciding admission decisions based upon a single requested polling interval and available capacity.

Simulations would be geared to answering the evaluation questions presented at the beginning of the section. We would conduct a series of simulations:

Simulations:

1. Simulations over a range of traffic loads
2. Simulations over a range of location dependent error rates
3. Simulations exploring a mix of application types including differing application profiles and best effort traffic

These simulations would not need to be a full factorial experiment. As we progress on the simulations we would learn the correlation between the varied parameters and system performance.

5 Conclusions and Future Work

This thesis has introduced two new QoS parameter sets that capture aspects of application adaptivity. These parameter sets express a simple model of the time scales upon which applications can adapt and define the paths along which the application can adapt. We have provided examples of ways in which these parameters could be employed to improve the service capabilities of wireless networks.

Our proposed QoS parameters provide a simple and effective way of capturing the adaptivity of applications operating over wireless networks with varying service capabilities. More research is required to validate the proposed QoS parameters. In particular ongoing research is addressing the evaluation of the system using realistic applications and traffic profiles. Another area of investigation is focused upon capturing and specifying adaptation requirements.

The ongoing research project is examining other ways in which these parameters can be employed. Of particular interest is using these parameters in conjunction with a utility model. The goal of this work is to devise distributed algorithms that will allow for maximizing global utility under the time and adaptation constrains specified.

Another aspect of the research being pursued is whether it is useful to generalize the parameters proposed. While a simple model often is best, we are interested in examining whether more information could actually be employed usefully. As is typically the case when considering generalizations the main issue is whether the generalizations actually are justified when complexity costs are weighted against the possible improvements.

Finally we are particular interested in developing a wireless test bed where these ideas can be explored in detail. The basic physical infrastructure already exists within the lab so the main challenge is to implement the supporting software system. The benefit to this approach is that we gain experience with using actual applications within our model. Coupled with ongoing simulation work this will provide a deeper understanding of the capabilities and advantages of our approach.

6 Bibliography

[AM98] E. Amir, S. McCanne, and R. H. Katz. An Active Service Framework and its Application to Real-time Multimedia Transcoding. In *Proc. ACM Sigcomm*, September 1998.

[AMZ95] E. Amir, S. McCanne, and H. Zhang. An Application Level Video Gateway. In *Proc. ACM Multimedia*, November 1995.

[BLV94] B. Belzer, J. Liao, J.D. Villasensor, ``Adaptive Video Coding for Mobile Wireless Networks," Proc. IEEE ICIP-94, Austin Texas 1994.

[BMJ98] J. Broch, D. Maltz, D. Johnson, Y-C. Hu, J. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Routing Protocols, Proc. ACM/IEEE MOBICOM, 1998.

[BCL98] G. Bianchi, A. Campbell, and R. Liao. *On Utility-Fair Adaptive Services in Wireless Networks*. In Proceedings of International Conference on Quality of Service, IWQoS, Napa, California, 1998.

[BRS00] M. Bechler, H. Ritter, and J. Schiller. *Quality of service in mobile and wireless networks: The need for proactive and adaptive applications*. In Hawaii Int. Conf. on System Sciences (HICSS-33), Jan. 2000.

[BCL98] G. Bianchi, A. Campbell, and R. Liao. *On Utility-Fair Adaptive Services in Wireless Networks*. In Proceedings of International Conference on Quality of Service, IWQoS, Napa, California, 1998

[CG97] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol." Wireless Netowrks 3 (1997) 217-234.

[CIJ97] Crow, Brian P., Indra Kim Widjaja, Geun Jeong, and Prescott T. Sakai. "*IEEE-802.11 Wireless local Area Networks*" IEEE Communications Magazine, September 1997, vol. 35, No.9: pages 116-126.

[CW00] D. Clark, J. Wroclawski, NSF Grant Proposal, 2000.

[CAM97] Campbell, A., Mobiware: QOS-aware Middleware for Mobile Multimedia Communications, Proc. IFIP 7th Intl. Conf. on High Performance Networking, White Plains, New York, April 1997

[Diffserv] IETF "Differentiated Services" Working Group. See http://www.ietf.org/html.charters/diffserv-charter

[DiffServ EF] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999

[DKS90] A. Demers, S. Keshav, and S. Shenker, Analysis and Simulation of a Fair Queueing Algorithm, Internetworking: Research and Experience, Vol. 1, No. 1, pp. 3-26, 1990

[ES00] D. Eckhardt and P. Steenkiste. *Effort limited fair scheduling for wireless networks*. In Proceedings of IEEE INFOCOM 2000, Tel Aviv, March 2000.

[E2E] J. Saltzer, D. Reed, D. Clark, End to End Arguments in System Design, ACM Transactions in Computer Systems, November 1984. See http://www.reed.com/Papers/EndtoEnd.html

[E2E-QoS] Y.Bernet, R.Yavatkar, P.Ford, F.Baker, L.Zhang, K.Nichols, M.Speer, R. Braden, Interoperation of RSVP/Int-Serv and Diff-Serv Networks, February 1999, http://www.ietf.org/internet-drafts/draft-ietf-diffserv-rsvp-03.txt, Work in Progress

[GLA99] J.J. Garcia-Luna-Aceves. ``SOURCE TREE ADAPTIVE ROUTING (STAR) PROTOCOL," draft-ietf-manet-star-00.txt. October 1999.

[GT(%] M. Gerla, J T-C Tsai. Multicluster, mobile, multimedia radio network. In Wireless Networks 1, pages 255-265, 1995.

[GCFH94] Atanu Ghosh, Jon Crowcroft, Michael Fry, and Mark Handley, "*Integrated Layer Video Decoding and Application Layer Framed Secure Login: General Lessons from Two or Three Very Different Applications*," in First International Workshop on High PerformanceProtocol Architectures,HIPPARCH '94, Sophia Antipolis, France, December 1994, INRIA France.

[GHAN92] M. Ghanbari, "*An Adapted H.261 Two-Layer Video Codec for ATM Networks*," IEEE J. Communications, Vol. 40, pp. 1481-1490, September 1992

[IntServ] IETF "Integrated Services" Working Group. See http://www.ietf.org/html.charters/intserv-charter.html

[ISSLL] Integrated Services over Specific Link Layers, see http://www.ietf.org/html.charters/issll-charter.html

[JON99] D. Johnson. ``The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," draft-ietf-manet-dsr-03.txt. October 1999.

[K95] KATZ, R. *Adaptation and Mobility in Wireless Information Systems*. IEEE Personal Communications Magazine 1, 1 (1995), 6--17.

[KYOM99] M. Kosuga, T. Yamazaki, N. Ogino, and J. Matsuda, "An Agent-Based Adaptive QoS Management Framework and Its Applications", in M. Diaz, P. Owezarski, P. Sénac (Eds.), Interractive Distributed Multimedia Systems and Telecommunication

Services, 6th International Workshop, IDMS'99, LNCS 1718, pp.371-376, Springer Verlag, Oct. 1999.

[LBS97] S. Lu, V. Bharghavan and R. Srikant, "*Fair scheduling in wireless packet networks*," ACM SIGCOMM'97, August 1997.

[LS98] Lettieri, P., Srivastava, M.B.: "*Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency*", IEEE Infocom'98, San Francisco, USA, pp. 307-314, March 1998.

[GALL(!] Didier Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, April 1991, Vol.34, No.4, pp. 47-58

[LC98] S-B Lee. A Campbell. INSIGNIA: In-band signaling support for QoS in mobile ad hoc networks. In Proceedings of 5th Intl. Workshop on Mobile Multimedia Communication 1998.

[LLB97] Lu S., Lee K.-W. and Bhargavan V., "*Adaptive Service in Mobile Computing Environments*", Proc. 5 th International Workshop on Quality of Service (IWQOS'97), Columbia University, New York, USA, Pages 25-36.

[L95] Lee, K., *Adaptive Network Support for Mobile Multimedia*, In Proc. of the 1st Annual International Conference on Mobile Computing and Networking, pp. 62-74, November 1995.

[ME98] Pratyush Moghe and Michael Evangelista, *An Adaptive Polling Algorithm*, Proceedings of Network Operations and Management Symposium (NOMS 98), New Orleans, Feb 1998

[OYM99] Ogino, M. Kosuga, T. Yamazaki, and J. Matsuda, "A MODEL OF ADAPTIVE QOS MANAGEMENT PLATFORM BASED ON COOPERATION OF LAYERED MULTI-AGENTS", Proc. GLOBECOM'99, pp.406-413, Dec. 1999

[Partridge] C. Partridge, Gigabit Networking, Addison-Wesley, February 1994, ISBN 0-201-563339

[RIL(%] M.J. Riley, I.E.G. *Richardson: FEC and Multi-layer Video Coding for ATM Networks*, in: Performance Modelling and Evaluation of ATM Networks, Vol. 1, Chapman & Hall (1995), 450 - 457

[RFC-2295] K. Holtman and A. Mutz. *Transparent Content Negotiation in HTTP*. RFC, Mar 1998. RFC-2295

[RHE99] R. Rejaie, M. Handley, D. Estrin, Quality Adaptation for Congestion Controlled Video Playback over the Internet, Proc. ACM SIGCOMM, September 1999.

[RSVP] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997

[SBM] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks", May 1999, http://www.ietf.org/internet-drafts/draft-ietf-issll-is802-sbm-08.txt, Work in Progress

[SHE95] S. Shenker, Fundamental Design Issues for the Future Internet, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 7, pp. 1176-1188, September 1995.

[SSB99] P. Sinha R. Sivakumar V. Bharghavan. ``CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm" University of Illinois at Urbana-Champaign.1999.

[SIN] S. Singh. Quality of Service Guarantees in Mobile Computing. In Computer Communications.

[SZ99] I. Stoica, H. Zhang. ``Per Hop Behaviors Based on Dynamic Packet States"draft-stoica-diffserv-dps-00.txt. August 1999.

[SSZ98] I. Stoica, S. Shenker, H. Zhang. ``Core-Stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks." Proceedings ACM SIGCOMM 98, pages 118-130, Vancouver, September 1998.

[T99] Y.C. Tay. ``Cluster Based Routing Protocol(CBRP) Functional Specification," draft-ietf-manet-tora-spec-02.txt. August 1999.

[TOS] Almquist, P. "Type of Service in the Internet Protocol Suite", July 1992, RFC 1349

[VJ99] Bobby Vandalore, Raj Jain, Sonia Fahmy, Sudhir Dixit " AQuaFWiN: Adaptive QoS Framework for Multimedia in Wireless Networks and its Comparison with other QoS Frameworks ," Submitted to the LCN '99.

[VCM00] M. Verraraghavan, N. Cocker, T. Moors, "Support of voice services in IEEE 802.11 wireless LANS", Infocom 2000.

[VZ95] M.A.Visser and M. Zarki. *Voice and data transmission over* an 802.11 wireless network. PIMRC, pages 648--652, September 1995.

[ZBS97] J. Zinky, D. E. Bakken, and R. Schantz. *Architecture Support for Quality of Service for CORBA Objects*. Theory and Practice of Object Systems, January 1997. Also see http:/www.dist-systems.bbn.com/ tech/QuO/.