

Capturing 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

© Massachusetts Institute of Technology 2000. All rights reserved.

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

Capturing 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 these 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 their support as I have worked on my Master's Thesis. Most particularly I would like to thank my parents and brother. 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 both advice and for being good friends I want to acknowledge Jo and Chuck.

1	INTRODUCTION.....	7
1.1	RELATED WORK.....	10
1.1.1	<i>QoS Models</i>	10
1.1.2	<i>QoS Frameworks</i>	12
1.1.3	<i>Improving the Network Stack</i>	13
1.1.4	<i>Overprovisioning Resources</i>	14
1.2	DESIGN GOALS	14
1.3	CONTRIBUTIONS	15
1.4	ROADMAP.....	15
2	APPLICATION AND NETWORK BEHAVIORS.....	16
2.1	ADAPTIVE APPLICATIONS.....	16
2.1.1	<i>Content Adaptations</i>	17
2.1.2	<i>Adaptive Algorithms</i>	18
2.1.3	<i>User adaptations</i>	19
2.2	NETWORK BEHAVIORS	20
3	ADAPTATION PATHS AND TIME CONSTRAINTS	21
3.1	ADAPTATION PATHS	23
3.2	TIME CONSTRAINTS	26
3.3	COMBINING ADAPTATION PATHS AND TIME CONSTRAINTS	28
4	APPLICATION AND NETWORK BENEFITS	ERROR! BOOKMARK NOT DEFINED.
4.1	ENHANCED APPLICATION SERVICE REQUESTS	21
4.2	DESCRIPTIONS OF APPLICATION BEHAVIORS PATTERNS.....	22
4.3	ADVERTISING NETWORK ADAPTIVE CAPABILITIES	22
4.4	PRESCRIBED BEHAVIORS FOR NETWORK CLIENTS.....	22
5	POTENTIAL USES ADAPTATION PATHS AND TIME CONSTRAINTS....	29
5.1	FRAMEWORK FOR INTER-SERVICE SELECTION.....	29
5.2	SCHEDULING ALGORITHMS THAT EMPLOY THESE CONSTRUCTS	29
6	MODEL AND EXPERIMENTATION... ERROR! BOOKMARK NOT DEFINED.	
6.1	MODEL	45
7	CONCLUSIONS AND FUTURE WORK	45
8	BIBLIOGRAPHY	ERROR! BOOKMARK NOT DEFINED.

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 some similar low intensity or for the most part uncritical network activity. Most of these applications and users are fairly tolerant of changes in the network service. If service drops below an acceptable threshold users are often simply willing to manually retry their network activity later. Overall existing users tend to be able to find enough times where service is acceptable that they persist in using their wireless applications.

Increasingly however there is a demand for applications with more intense network service requirements. Both streaming audio and video applications would 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 there are demands for improved wireless service capabilities. Numerous research groups and standards bodies are therefore addressing these issues for wireless networks (eg. [IEEE 802.11] 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 as well, 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 [Rapport]. Link layer impairments have very little effect in wired networks. In wired networks, resources are not as scarce; wireless networks will always 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 problematic 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 they 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.

All of 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 project is to capture the adaptive characteristics of application and network behaviors that can be used to improve the quality of service provided to applications in wireless networks. 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 variations. 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.

Again *both* applications *and* networks potentially can benefit by exploiting these characteristics. The additional information provides increased predictability of application and network behaviors as well as allows the network to have additional, though constrained, flexibility in satisfying application service requirements. Adaptation paths and time variations can be used to specify requested behaviors or they can characterize actual behaviors. Either can potentially be useful to applications and networks.

The prospective applications are promising for systems utilizing adaptation paths and time variations. 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 not based upon application profiles also 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. This is not intended as a comprehensive review of ways of expressing quality of service requirements or ways of describing network traffic flows. Therefore, this section does not review the sizeable body of literature dealing with all other ways of expressing various QoS parameters. What this section does emphasize are models that capture aspects of application behaviors that are related to time and adaptation paths.

Previous research projects have identified the importance of time scales upon which applications can adapt [Katz] [Lee] [Campbell]. Of particular relevance is the Mobeware project [Campbell]. 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 Mobware 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.

Many authors have identified the importance of the graceful degradation of service [Singh]. One common way of addressing this issue is by providing of layers of service [RHE99], [GCFH94]. In particular these models often target streaming video applications whose traffic flows can naturally be decomposed into various layers of quality [RHE99], [Ghanbari]. In some of these models, service adapts in a discrete manner [Riley]. As congestion on the network varies or channel capacity changes, these models drop or add entire layers. Other applications adapt more smoothly through their layers by incrementally including additional parts of a layer. For these applications the layering model does not produce as much of a pronounced effect in terms of changes in the traffic load on the network.

Numerous QoS models capture some 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. They identify the fact that applications have a limited amount of flexibility in the services that they employ. Most examples of these QoS models use closed ranges, but conceptually it is easy to imagine that a QoS model could specify a lower bound and an unlimited upper bound. 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 [Campbell], [ZBS97], and [CFK98]. These frameworks define a mapping between application requirements and network services and define admission control or resource reservation protocols. Many also define some flow scheduling, shaping or control algorithms. Finally some aspects of flow monitoring, QoS alerts, and QoS maintenance are addressed by many of the frameworks.

These frameworks apply different approaches in specifying and translating the applications needs into network control parameters. The translation is either preformed by the application designer and embedded into the application itself or the translation is performed by a translation 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. Then the translation itself is most often simply a matter of applying basic pattern matching techniques.

QoS frameworks typically exploit some aspects of application flexibility [BRS00]. QoS is negotiated initially and adaptation occurs does not occur unless application requirements change or the network services can no longer support the reservations. The QoS models 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.

Some 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 or costs in terms of the amount of network resources consumed Value decisions are made based upon specifications of the “utility” of the requested network services. Again utility is defined differently in various frameworks.

1.1.3 Improving the Network Stack

An alternate approach to improving the quality of service capabilities of networks is to modify various layers of the network stack. This section discusses various techniques for modifying the link, network, and application layer. These techniques 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 attempt to improve the predictability, efficiency, and fairness of channel access. Various medium access control strategies have been proposed to provide fair access to channel. Other fair access techniques have been devised incorporating more complicated 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 devised allow certain predictable errors to be avoided [LBS97]. Other techniques for diminishing error rates experienced include dynamically adjusting transmission power and dynamically changing the encoding schemes [LS98].

At the network layer similar strategies have been devised. Employing explicit reservations schemes such as RSVP have been proposed [RSVP]. Similarly employing 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 can cope with a wide range of network capabilities by adapting their behavior. Applications can 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.

Further the algorithms that applications rely upon can be adaptive in nature. TCP for instance 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. Overprovisioning wireless network resources is probably not a sufficient solution. User expectations and requirements for both the wired and wireless environments will never diverge far enough that wireless capacity will satisfy user needs. Even 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 Design Goals

One goal of this thesis is to characterize and exploit the adaptive capabilities of networks and applications. We were motivated by the need to improve the quality of service that wireless applications achieve. We were wanted to explore using adaptation to not just reactively cope with variations but actually leveraging adaptation to provide enhanced QoS services. Specifically we wanted to explore correlating application's adaptive capabilities to the adaptive capabilities of networks.

A related goal is to increase the overall amount of useful work done by wireless networks, thus improving their efficiency. This goal stems from recognition of the differing value of identical network services to different applications and different users. By considering these differences intelligent choices can be made as to how network resources are assigned and how application service is adapted as network conditions vary.

1.3 Contributions

In addressing these goals this thesis makes the following contributions. First the primary intellectual contribution of this thesis is the identification of the importance of time scales upon which adaptation occurs and the articulate of the idea of adaptation paths over which application and network behavior changes. These 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 potentially improve networks and applications.

The second contribution of this thesis is the design and implementation of an algorithm for improving the capabilities of nodes participating in an 802.11 wireless network. The purpose of the algorithm is to moderate when nodes switch associations to alternate base stations. This is important since currently nodes only change associations when signal strength drops below a defined threshold regardless of application needs.

1.4 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 detailed description of the concept of time constraints and adaptation paths and describes the approach taken in this thesis for expressing these concepts. Chapter 4 discusses the potential advantages of employing these characteristics both for applications and networks. Chapter 5 proposes ways of employing these new characteristics to improve quality of service. Chapter 6 describes a specific algorithm that employs these characteristics to mediate when nodes in 802.11 networks switch base station. Chapter 7 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 network behaviors by adjusting their behavior accordingly. The ability to adapt improves the quality of service provided to an application. If additional network resources become available which an application can utilize adaptive applications can use it 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. Even if applications do not derive any direct benefits, adapting improves network efficiency.

Adaptation can be modeled as a feedback loop consisting of application behaviors and network services. The variables and driving function of the feedback loop depend upon the network and application in question. The network or the application can control either's adaptive behaviors. The probing mechanism that determines available capacity can reside within the application or the network.

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 either transparent or readily apparent to the application and application user. Finally the adaptations are driven by either the network itself or by an application controlled processes.

2.1.1 Content Adaptations

Applications often have a choice of how data transmitted through the network 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 data representation is selected.

Different data representations are the product of various encoding levels or techniques. For instance, video can be compressed using any of the different MPEG standards. The target bit rate for MPEG-1 is typically 1.5 Mb/sec though higher bit rates are supported by the standard. The MPEG-2 target bit rate is between 4 and 9 Mb/sec. Much higher bit rates are used to encode higher quality pictures; the HDTV resolution of 1920x1080 pixels at 30 frame/sec, produces at a bit rate of up to 80 Mb/sec. 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].

It is important to note that the encoding method itself can result in a variable bit rate stream. For instance, when an audio stream is encoded certain techniques can 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. Options do exist to make an encoder produce a constant bit rate stream at least for MPEG encodings.

When applications are presented with a multitude of data representations, they choose 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 only one HTTP request.

Other applications can 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 type of 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 users specified. Users 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. However 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 both the amount of resources that can be made available to a particular client as well as the quality of that service.

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.

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 connections 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 to some degree various mobility patterns for mobile nodes.

Some networks also provide the functionality of disturbing 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 application and network behaviors can be described. 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 behaviors.

These concepts can be used to characterize both application and network behaviors and can describe either the way behavior does occur or the way that the behavior should occur. Both applications and networks benefit if the behaviors of each can be correlated both in terms of time and in terms of the paths along which the behaviors change.

The main benefits to applications are that they can provide smoother transitions in service quality. Applications can have an increased reliance on the network since packets the network handles are more likely to actually be the ones needed. Applications changing their service are more likely to be accommodated since they do so in a specified way. Similarly networks can be made more robust by leveraging these ideas. Networks can be more likely to provide the services that applications actually require. Overall more useful work is performed by the both the applications and the networks.

3.1 Enhanced Application Service Requests

Adaptation paths and time constraints could be used to extend resource reservation requests. The application could indicate that it needed to receive a particular service for at least duration of time otherwise it would prefer to have the reservation rejected. Similarly an application might specify that it could tolerate interruptions in service below some threshold time. If the average 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.

3.2 Descriptions of Application Behaviors Patterns

Applications might also dictate to the network 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 Advertising Network Adaptive Capabilities

Networks could similarly describe their capabilities using adaptation paths and time constraints. Networks could advertise that they support certain service levels and that they allow applications to move between those service levels on certain time scales. Networks could also advertise that certain interrupts in service are likely for particular durations of time. These interrupts may be either related to environmental effects or network behaviors such as handoff times or overhead traffic.

3.4 Prescribed Behaviors for Network Clients

Similar to advertising the networks adaptive capabilities, a network might prescribe that applications using its service must be prepared to behave in a certain manner. Networks might for instance dictate that applications be prepared to gracefully degrade in the face of increased network load. The time scales upon which this adaptations might be forced could be specified. Similarly 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.

Application and network behaviors can be characterized using many different metrics. For applications, existing QoS models provide a wide range of descriptive possibilities. The simplest metrics are required bandwidth, delay parameters, error rates, and jitter. Network services often are characterized in similar terms of available channel capacity, maximum delay, and expected error rates. 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 entirely unpredictable. The bandwidth for instance that an application consumes might range from 0 Mbps to the total channel capacity. However the transition from the lowest to highest bandwidth occurs along a predictable path and at predictable intervals. Indeed completely random movement of a metric about its range frequently is seldom tolerated for either application or network behaviors. Video applications for instance perform poorly if the channel capacity oscillates frequently or too wildly.

Indeed, changes in a metric, either one declaring a requirement or one describing likely behaviors, often occur in a predictable manner. 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.

3.5 Adaptation Paths

Other authors have emphasized the importance of allowing for the graceful degradation of services. This is usually discussed in the context of reservation systems. An initial reservation for a certain level of service is requested. If the initial reservation must be compromised, then service is degraded to one of the alternative lower quality levels.

Further degradations to even 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 that these alternatives service levels exist is important. However the ways in which movement occurs between these layers is also important. Multi-layer video applications for example add a discreet amount of new load for every additional layer of video. In most cases these layers are added sequentially, however more then one layer of video can be removed at a time as service degrades. A graphical depiction of the changes in service from one layer of video to another is shown in Figure 1. These are simple linear adaptation paths.

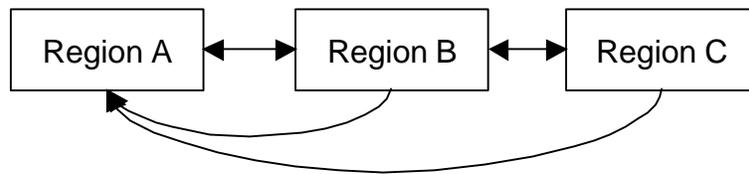


Figure 1: Linear adaptation paths

Multiple options for adaptation might exist. Service can transition from a one operating region to a constrained set of other regions. For instance an application operating at 1 Mbps might be able to handle upgrading to either 2 Mbps or 3 Mbps. In Figure 2 for instance there can be a transition from Region A to either Region B or Region C. Similarly applications may be able to handle degrading to different levels.

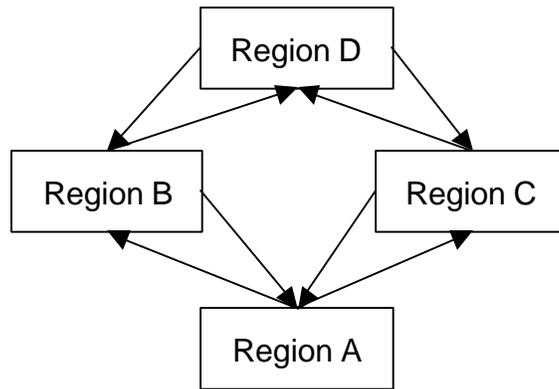


Figure 2: Multipath Adaptations

Adaptation paths might not be symmetric. Just because an application can transition from one operating region to another does not necessarily entail that it is 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 around. This may correspond to applications that can increase at a predictable rate but stop abruptly and have to be resumed gradually.

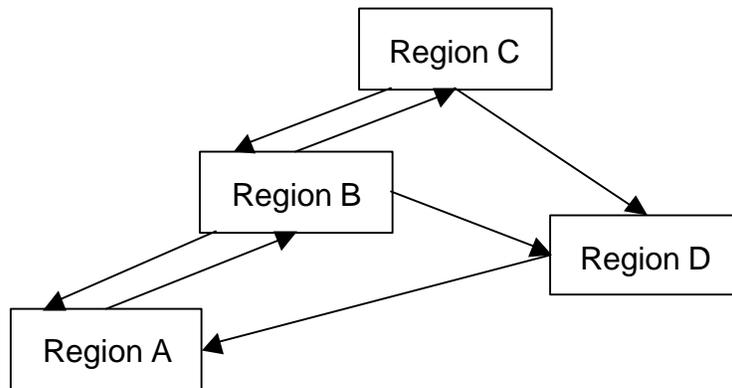


Figure 3: Directed Adaptations

3.6 Time Constraints

Another important aspect related to adaptation paths is the time scale upon which application and network behaviors change. Previous research work has discussed the importance of adaptation time scales [Campbell]. However this is generally a single parameter that specifies one time scale upon which the application can adapt. Thus it is not dependent upon the level of service being provided.

In many ways a single adaptation time scale might 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 that could be expressed by a single adaptation time include 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.

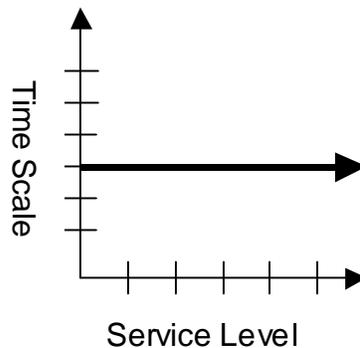


Figure 4: Single Adaptation Parameter

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.

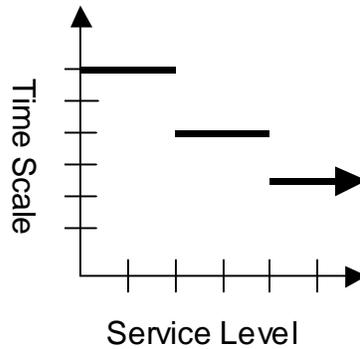


Figure 5: Service Level Dependant Adaptation Times

Another way of describing applications traffic and network service using time is by describing potential interrupts durations. Applications for instance may be able to tolerate 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.

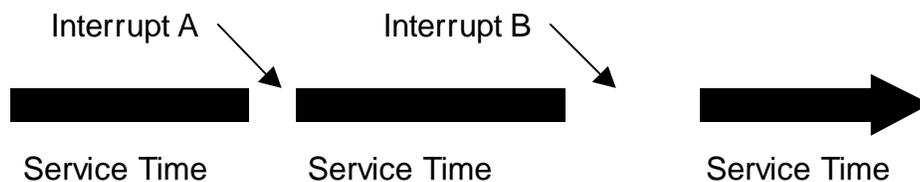


Figure 6: Service Interrupt Times

Acceptable interrupt durations 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 than 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.

3.7 Combining Adaptation Paths and Time Constraints

Time can also be used to capture information important to adaptation paths. In particular the rate of change between operating regions can be expressed. Previous research has included the notion of rate of change however it typically is a single parameter used to indicate the rate of change acceptable between all service layers. It is also important to note that the rate of change may not be symmetric and may vary depending on the level of service being provided. Rate of change may specify minimum, maximum, fixed, or ranges for the acceptable rates of change.

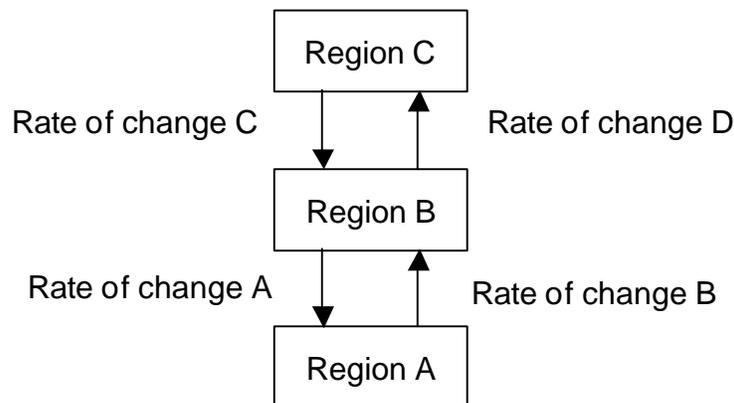


Figure 7: Adaptation Paths and Rates of Change

Potential Uses Adaptation Paths and Time Constraints

This chapter provides a number of specific examples of ways in which time constraints and adaptation paths can be used. We present these as potential advantages because while work has begun on each of the project ideas listed below we have not finished implementation or evaluating these ideas. The ideas seem promising though and future work will present the final results.

3.7.1 Scheduling algorithms that employ Discernible Service Times, Interrupt Times and Adaptation Paths

A second application of adaptation paths and time constraints being explored is using them to specify flow profiles that are used to improve scheduling algorithms. The flow profiles indicate acceptable adaptation paths and the applications time constraints. A scheduling algorithm then can employ these constraints to maintain the flows within their profile.

We have added an additional element to our flow profiles to capture the relative worth of network resources to various applications. These are simple expressions of utility that relate application performance to services provided in the network. This mapping allows the network to more efficiently allocate its resources to improve the overall level of service provided. Thus, utility profiles are used to guide resource allocation decisions. The granularity at which the profiles are employed varies according to network capabilities. On one end of the spectrum the utility profiles could be consulted only at flow admission time. On the opposite end of the spectrum the utility profiles could be used to determine each packet scheduling decision.

The approach we are exploring lies somewhere in the middle of these two endpoints. The utility profiles are used to determine flow admission and the initial parameters of the scheduling algorithm. The utility profiles are not used to make per packet scheduling decisions. The scheduling algorithm maintains the flow within the appropriate operating region if no outside perturbations disturb the flow. Whenever a disruption does occur,

the allocation algorithm will reassign resources. Such disruptions occur when a flow falls out of its current operating region or more capacity becomes available for assignment to reserved flows.

The goal in assigning resources is to maximize the global utility for reserved flows. This can be accomplished by using a distributed algorithm where each node participating allocates resources to maximize aggregate flow utility subject to the adaptation constraints of the applications sending and receiving the flows. It should be noted that the utility maximization is applied only to reserved flows at a node. Thus once a flow has been accepted at a node, it is not preempted by the arrival of a later flow that might provide a higher overall utility. Non-preemption was chosen to improve stability for reserved flows.

3.8 Framework for inter-service selection

We are employing adaptation paths and time constraints in our ongoing research project, A Wireless QoS Framework. The goal is to design a system architecture that significantly improves the quality of service that wireless applications achieved. The quality of service can be improved in a number of important ways. First, users and applications have tolerances as to how flexible they are in handling network service variations. These tolerances and the utility of wireless services vary according to application and user. By understanding these adaptation constraints and application utility differences more intelligent choices can be made to improve the aggregate utility of all applications at a node. Second, the quality of service achieved by applications can be improved by understanding the varying service capabilities of wireless networks. With this understanding more intelligent mapping between applications requirements and network resources can be achieved. In other words, scarce network resources can be used more efficiently.

In our architecture, the adaptability and utility of applications are captured by an Application Utility Specification (AUS). An AUS provides a way of expressing the user perceived utility of the variable services provided to applications. Utility functions have been discussed for many years, however it is our contention that our AUS captures many

important aspects of application adaptability that have not been adequately expressed previously. In particular our utility profiles provide ways of expressing important notions about the time scale on which applications can adapt. Further, previous utility profiles tend to express utility as a function of one network parameter. While it has been proposed many times, our scheme does allow utility to be a function of multiple network characteristics such as bandwidth, delay, and error rate. We feel this is particularly important in wireless networks since significant variability in service can occur in any of these parameters.

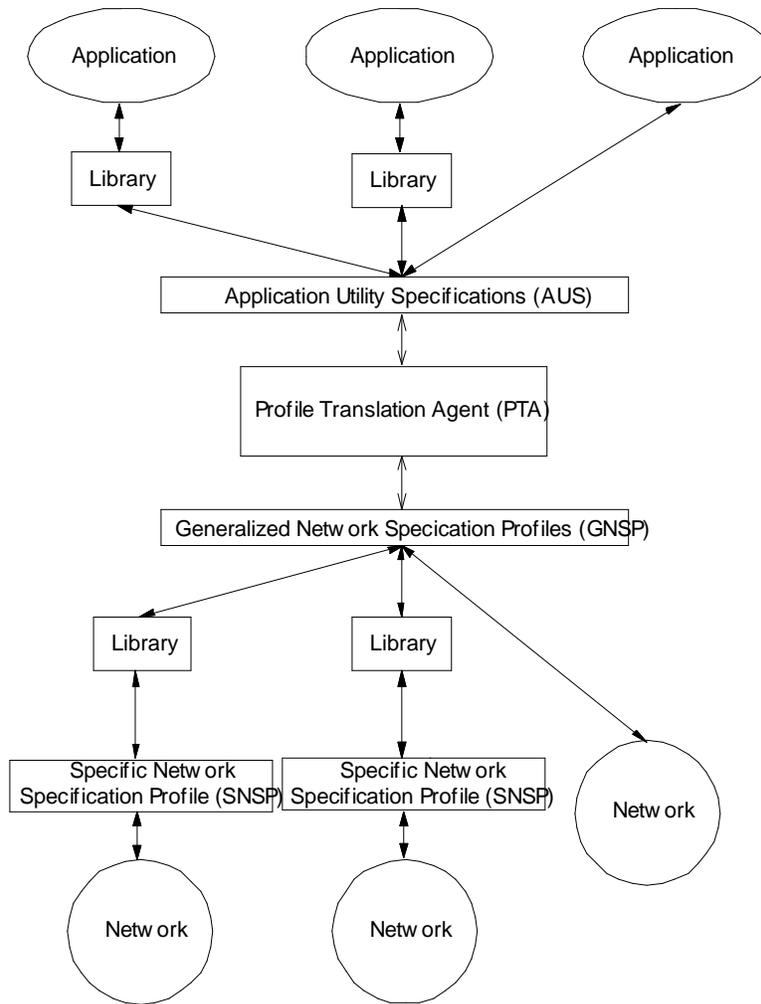


Figure 8: QoS Framework employing Adaptation paths and time constraints

The second main component of our system, Network Specification Profiles (NSP) provides characterizations of a network services. These capture the differences in the service that a wireless network can provide along with other characteristics of a particular wireless network. The purpose of expressing the service capabilities of a network is so that application service needs can be mapped to appropriate network resources. These provide a way of distinguishing wireless networks and reasoning about the differences in the services that each provides.

Finally a Profile Translation Agent (PTA) provides the functionality for actually translating between the applications service requirements specified in the AUS and appropriate network services expressed in the NSP. This mapping of application requirements to underlying network services occurs both across diverse wireless network types as well as within one particular network. This translation is performed both to initially selecting a particular network service for an application when it begins using a wireless network and later to appropriately adapt the service provided to the application as network service conditions vary with time. The time scale on which the utility profiles are employed to adapt a flow depends on the specific network adaptation mechanisms.

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. We argue that our polling algorithm improves upon the conventional round robin polling strategy by enabling flows polling intervals to be appropriately adapted in the face of location dependent channel error and varying channel loads. Using our strategy flows are both gracefully degraded and upgraded according to application specification adaptation paths. Our polling algorithm observes flow specific adaptation time constraints as well.

The context for our proposed polling algorithm is an 802.11 Point Control Function (PCF) mode wireless network. An 802.11 PCF mode network is an appropriate network for employing time constraints and adaptation paths since it is centrally administrated, designed for applications with better than 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 becoming widely deployed and will increasingly 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 DCF. The DCF mode is carriers sense multiple access with collision avoidance (CSMA/CA). It 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 PCF 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].

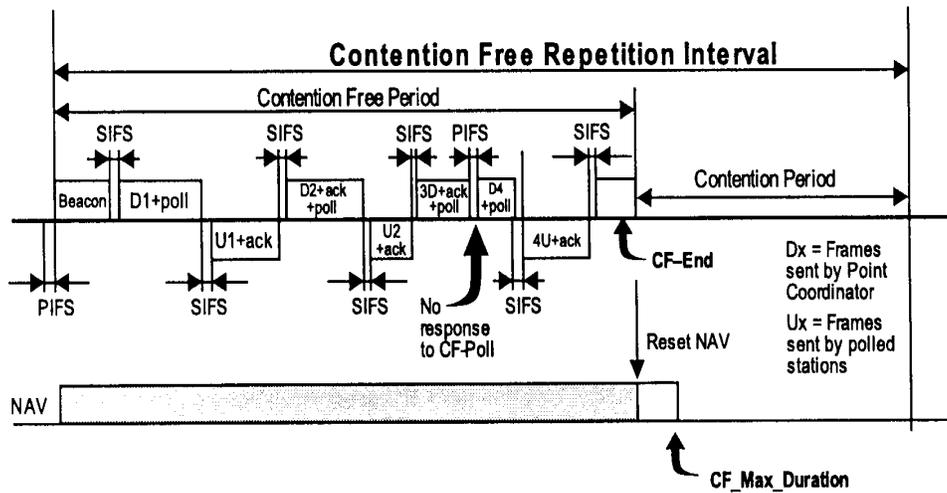


Figure 9: PCF Framing Structure [802.11]

Channel access in the PCF 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 CFP. During the contention free period the PC is free to transmit any of its queued packets to a STA in the BSS. The PC polls any associated STA to determine if it has packets to transmit. No node can transmit during the CFP unless the PC has polled it. The maximum duration of the CFP is indicated in the initial beacon marking the beginning of the CFP. The PC can end the CFP early with the transmission of a CF_End frame. During the CP all nodes compete for channel access using the DCF methodology.

The stations polled during the CFP are determined by a polling list maintained at the PC. The 802.11 Specification does not mandate a mechanism for creating or maintaining the polling list at the PC. This is considered out of scope for the standard so manufacturers are free to implement one of their choosing. The method by which an AP utilizes the polling list to perform polling of STA 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 could poll each node on the list exactly once, less than once, or more than once per CFP. The polling policy employed will depend on the type of service that the PC is attempting to provide.

A drawback of the PCF is that it is not particularly scalable. The PC needs to control media access by polling all stations, which can be ineffective in large networks. In a DCF 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 DCF. Therefore for services that require delay bounds, the PCF mode is most appropriate given the current standard.

4.2 System Architecture

Our system architecture is an 802.11 wireless network consisting of mobile stations and access points. The interaction between the access points and stations occurs according to the PCF 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 Interface* is the basic polling procedures specified by 802.11 modified slightly so that adaptation information can be included in the poll or ACK packets. The architecture is depicted in the following figure.

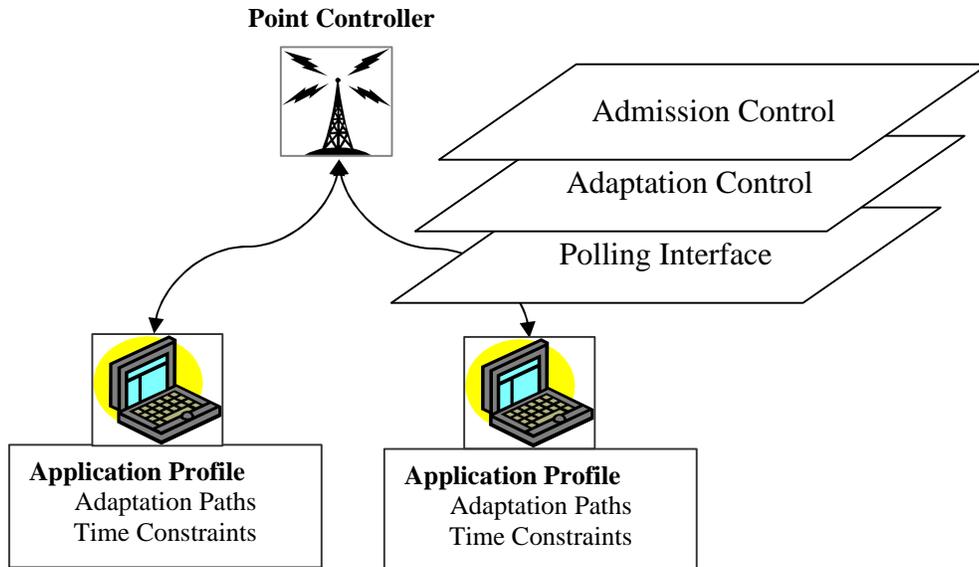


Figure 10: System Architecture

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 basic 802.11 interface. 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 PC during the *admission protocol*. A STA transmits an admission request during the contention period.

We assume that only one application is “active” per STA 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 temporary 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 be 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 PC can cache the commonly used one 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 11: 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 or twenty 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. They could be determined by some other external mechanism

and even stored separately, perhaps in an application profile database. This would enable applications to remain unmodified and allow them to work seamlessly over 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 request and reply 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. If the PC has the named profile cached an admission decision can be made immediately. If the PC 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 PC 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 12: Admission Protocol

Once a node has had an application admitted to the polling list the node will be either be polled during the CFP or receive packets during the CFP 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 PC 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 PC 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.

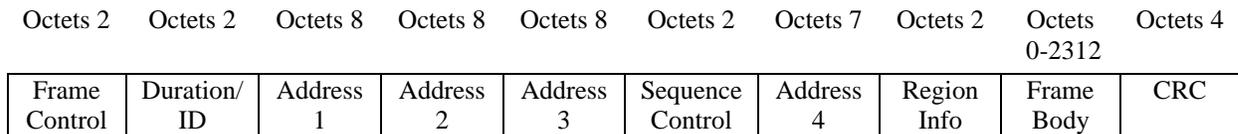


Figure 13: MAC Frame format

Similarly, applications indicate their preferred operating regions using the *Region Information* frame element on messages they transmit to the PC. Applications may preemptively notify the PC 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
<i>Current Region:</i> Indicates to a STA the level of service that it will now receive.	<i>Downgrade:</i> Indicates to PC that the STA has access capacity and can be downgraded to the indicated operating region
	<i>Region Preference:</i> Indicates to PC if the STA satisfied or would prefer upgraded service,

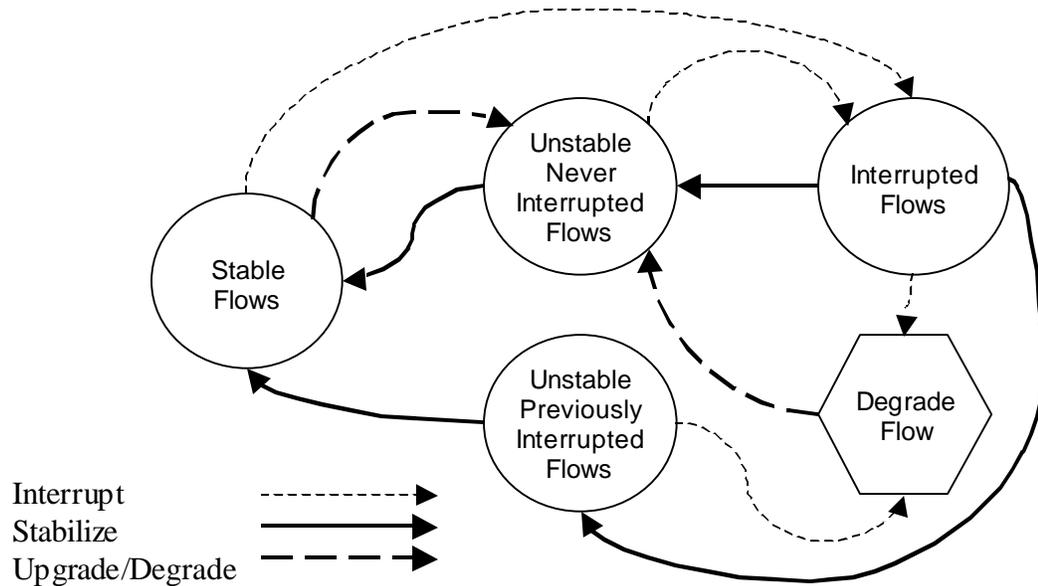
Figure 14: Region Information

4.2.2 Point Controller

This section describes the behavior of a point controller that employs 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.

Point Controller State	Purpose
Application Profiles	Cached profiles are used to determine adaptation
Node Bandwidth Allocations	
Flow Performance	
Channel Capacity Information	

4.2.2.1 Admission Control Policy



Unstable Flows

Interrupted Flows

4.2.2.2 Bandwidth Allocation Algorithm

If the requested bandwidth is less than total channel capacity then accept.

If the requested bandwidth is greater than channel capacity then we attempt to readjust all stable flows to achieve bandwidth average. $B_{total} - B_{unstable} - B_{interrupted} / n$. Give everyone the operating region below this point (some may change some may already be at this point. Then any remaining capacity distribute according to Jain's Fairness. This greedily attempts to maximize fairness of allocation. It does not maximize utilization. A more complex optimization problem.

4.2.2.3 Flow Maintenance Algorithm:

Move unstable flows to stable flows. This may require rebalancing the stable flows.

For any flows that are not using their requested resources (or are not responding to polls) move them to the interrupted list. The point of the interrupted list is to identify flows who have fallen out of their operating region either because they aren't using their resources or because they are experiencing channel error (location dependent errors).

Set bit for flows that have been interrupted. Present state diagram for what happens to interrupted flows.

4.3 Evaluation

This purpose of this chapter is to describe the methodology that would be employed to evaluate an application of adaptation paths and time constraints in a specific mechanism or scenario. We have not yet implemented anything that employs these ideas so we can only describe the evaluation approach we will take once our implementation work has progressed. We acknowledge that this is inadequate and do not suggest that this supports our contention of the utility of adaptation paths and time constraints. We present our evaluation techniques simply to demonstrate the steps that will be taken to judge the applications of our ideas.

The application of adaptation paths and time constraints requires an additional degree of complexity in the algorithms employed and potentially constitutes an additional bandwidth cost to communicate additional QoS parameters. Network architects and application designers would be faced with additional work to specific or understand our new parameters. Therefore we understand that we have to convincingly be able to demonstrate that adaptation paths and time constraints actually make a significant performance difference to justify their potential additional cost and complexity. We regret that we have not done this through a convincing implementation already. In some cases we may not be able to demonstrate that we can improve performance enough to warrant the inclusion of adaptation paths and time constraints. Either way we need a thorough way of evaluating their applicability in terms of both costs and performance improvements.

This chapter therefore is divided into two parts. The first part describes the metrics that we suggest are important in determining the costs and performance of a system incorporating our ideas. To demonstrate concretely our evaluation methodology, the second part presents an example application and describes in detail how we would evaluate it. The reader, by the end of this chapter, hopefully will understand what we consider is important in assessing our ideas and be convinced that our evaluation methodology is sound.

4.4 Evaluation Metrics

Our evaluation methodology emphasizes understanding the performance benefits as well as the associated costs of implementing our ideas. Of particular import is the ability to understand the application, traffic load, and network topology scenarios in which our extensions can be usefully employed. It is important to understand the scenarios in which our ideas apply to convincingly argue that for their deployment. The metrics presented in the following sections can be used in making these cases. While every metric will not be appropriate for every experiment, these are presented to stress the importance of a complete evaluation and demonstrate the metrics we will be using in arguing the utility of our ideas once our implementations have been completed.

4.5 Example Evaluation of a Proposed System

This part presents the evaluation methodology we would employ to determine the worth of the proposed polling algorithm. We include this part to demonstrate concretely the evaluation approach that we would have taken had we completed work on any of our ideas. Indeed we do not know whether these proposed experiments would confirm or disprove the validity of adaptation paths and time constraints in this one application of the ideas. If the experiments described below did not produce the intended results we would proceed by selecting a different application. We have confidence that the ideas themselves are sound even though we have yet to demonstrate it.

The idea we decided to describe the experiments for is the 802.11 PCF polling algorithm employing adaptation paths and time constraints. We selected this initially because it is one of the easier ideas we have suggested to quickly simulate. However most polling algorithms we found in literature are also simple in nature and seemingly could be rather easily modified to include our ideas. It also is easier to describe and understand the evaluation methodology for this idea given that we do not have any results and are therefore lacking graphics that are often useful in conveying the why a metric was employed. We first present the general nature of the models we would explore and what we hope to learn from them and then describe specific simulations that we would run.

4.5.1 Simulation Model

For the MAC layer, we adopt the general simulation 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. Parameterization details for the model are still forthcoming from the group so for the purposes of this research appropriate default values will be selected based upon other simulators, most particularly the ns-2 simulator. We employ RTS/CTS for packets above 1000 octets in size. (The default RTS threshold on both GlomoSim and ns-2 is zero bytes. However at least some 802.11 radios have a default setting of 1000 bytes.) The group defines no 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.

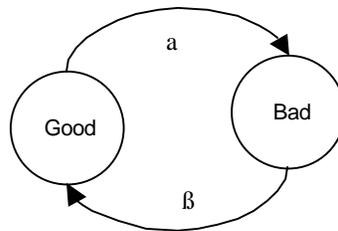


Figure 15: Markov chain burst error rate model

The environment we would simulate would be a flat square area. We are primarily limited here by models included with the simulators. Ns-2 only includes a flat environment though OpNet does provide for more complex configurations. Initial simulations would be done on a flat ground using either simulator. This provides for simple easy to understand scenarios even if they may not be terrible realistic. The number of nodes we would simulate would vary, as would the size of the environment. We do

this to experiment with different node densities. However this might not be terribly crucial given the extensive amount of ongoing work to modulate transmission power. For deployed networks with high node density we would expect power levels to be appropriately decreased.

We assume a static routing environment where any communication between applications occurs directly between nodes. No ad hoc routing algorithms are employed. It is an open research question which ad hoc routing protocols is most appropriate for different topologies, traffic patterns, and mobility models. We are interested in exploring the performance of MAC layer configurations in these experiments so the only impact excluding ad hoc routing protocols has is to limiting the possible peers that applications can communicate with.

No mobility models are employed in our simulations. Mobility is most interesting when examining a higher-level routing protocol, handoff procedures between multiple access points, or for the purpose of varying a link's performance characteristics. However in the simulations we would perform to validate the modified PCF polling algorithm we would only employ one access point so not handoffs would occur. We can vary link layer performance characteristics without a mobility model.

The applications we include in our simulation experiments are a mix of flows for which we require guaranteed delay bounds for channel access times and best effort flows. We include both types of flows to understand the effect of our modifications on traffic flows that do not employ our modifications. Since we do not have models of applications that actually employ time constraints and adaptation paths we simulate these applications using CBR applications. What we would do is change the sending rate of the CBR application in response to how many packets were in the outgoing queue. This is a cheap way of simulating an application that adapts its sending rate between multiple constant sending rates. We intend this to be somewhat representative of a RealMedia server that can select between multiple content representations. The best effort flows would be TCP based applications, primarily FTP.

4.5.2 Evaluation Metrics:

We evaluate our experimental runs according to the following metrics. We acknowledge that these are not well defined metrics but hope that they indicate the type of metrics that we believe are important in evaluating the application of time constraints and adaptation paths to the PCF polling algorithm.

1. *Packet deliver ratio*: the ratio of the number of packets originated at a source's application layer to those delivered to the destination application by the time the simulation ends
2. *Fairness*: a comparison of the throughput achieved for all flows with identical application loads over identical time spans.
3. *Latency*: the time to deliver a packet from the source application layer to the destination application layer.
4. *Polling success rate*: the percentage of polling packets that generated a responding transmission.
5. *Channel utilization*: the aggregate number of bytes that a channel was able to deliver over the duration of the simulation run

4.5.3 Simulations

To evaluate the utility of our model we compare identical application loads, topologies, and traffic patterns on networks employing both DCF and PCF mode channel access mechanisms. The reason we include DCF mode 802.11 networks is so that we can compare the channel utilization between both DCF networks and our modified PCF polling algorithm style networks. We do not include the DCF models to make any comparison claims between regular PCF and DCF. We simply wanted to be able to answer critical questions regarding the appropriateness of a centralized polling algorithm for applications that have limited adaptive capabilities.

Simulations 1 and 2 are designed to explore the throughput, latency, packet delivery ratio, fairness, and channel utilization. We expect to demonstrate that under DCF channel access times are unpredictable and that the short-term fairness is not achieved. In the second simulation the polling policy is the primary subject of investigation. We hope to

demonstrate that channel utilization is decreased as the AP spends time polling stations that do not have data packets to send. Though we expect that short-term fairness will be more favorable for the PCF. We would run these test for multiple trials and explore the variance as well.

Simulation 1	
Nodes:	20 Nodes
Channel Access Mode:	DCF
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	OFF

Simulation 2	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: round robin)
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	OFF

The second set of simulations is designed to explore the same set of experiments except with the error model in use. We would vary the error parameters to simulate harsh fading conditions present in some wireless networks. We would evaluate the same set of metrics as the previous set of experiments. If necessary to saturate the channel we would increase either the number of application flows or the bit rate of the CBR applications.

Simulation 3	
Nodes:	20 Nodes
Channel Access Mode:	DCF

Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	On a: 30 sec, β : 10 sec^{-1} Good BER: 10^{-10} Bad BER: 10^{-5}

Simulation 4	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: round robin)
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	On a: 30 sec, β : 10 sec^{-1} Good BER: 10^{-10} Bad BER: 10^{-5}

4.5.4 Theoretical Model of the PCF

For the PCF, an ideal model of the polling policy can be established. The ideal polling policy is one where errors are predicted perfectly, every station polled has a packet to send and can send it successfully, and fairness is achieved for all nodes. Fairness for the purpose of this model is defined as one where all stations on the polling list should receive an equal share of the channel access. Other possibilities include considering a model that allocates channel resources according to application requirements or different notions of fairness.

If no channel errors exist and all stations have a packet to send at all times, a theoretical ideal polling policy would be a round robin polling of all nodes on the PC polling list. If some stations occasionally do not have a packet to send, the ideal policy would be a round robin polling of the list including only stations ready to transmit a packet. If a station previously skipped acquires a packet to send there are two options for when to

poll it next. The first option would be to wait until its natural turn on the polling list came around again. The second option would be to poll a station as soon as a packet became available if it had been skipped on the past iteration. For the purpose of this model we adopt the first option and poll strictly according to the order established by the polling list. Similarly if channel errors can be predicted then stations which would experience errors either in transmitting their packet or receiving the polling signal will be skipped on the polling rotation.

We would rerun the following experiments using an error predictor and perfect knowledge of nodes with information to send. We would investigate the throughput and fairness properties of both these simulations. We would expect that throughput for all flows and thus channel utilization to improve markedly. We would vary the bit error rate and the on off durations for the CBR traffic to determine how harsh the wireless environment and how dynamic the application behaviors had to be before the perfect polling policy produced meaningful different results.

Simulation 5	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: ideal polling)
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	OFF

Simulation 6	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: ideal polling)
Simulation Time:	900 secs
FTP Applications:	10 connections

CBR Applications:	10 connections (Rate: 64 Kbps, packet size 1500 bytes.)
Error rate parameters:	On a: 30 sec, β : 10 sec ⁻¹ Good BER: 10 ⁻¹⁰ Bad BER: 10 ⁻⁵

4.5.5 Polling Algorithm utilizing time constraints and adaptation paths.

After implementing the polling algorithm described in the previous chapter we would rerun the pervious set of experiments using the new polling algorithm. In particular we would examine the percentage of times stations were successfully polled for packets. While we obviously could not achieve the success of the ideal polling algorithm we hope to improve upon the standard round robin polling policy. We would compare the results achieved using our polling algorithm to the fairness, latency, and throughput results achieved from the initial DCF mode simulations as well. We expect that fairness would be improved but overall channel utilization might still be lower.

The polling algorithms would be supplied with the following simple model (see figure 10) of the CBR behavior. It indicates that the CBR applications require polling at intervals off every 15 ms.. The interrupt interval indicates that if the application does not respond to a poll for over 300 ms then the polling algorithm can poll the station at a less aggressive interval defined by the second operating region, of every 500 ms. If the station replies to polls at the 500 ms interval then the polling algorithm may begin polling at the more frequent interval.

<ul style="list-style-type: none"> • Region A • Polling Interval: 15 ms • Interrupt Time: 300 ms • Adaptation Path: B
<ul style="list-style-type: none"> • Region B • Polling Interval: 500 ms • Interrupt Time: 0 ms • Adaptation Path: A

Figure 16: Adaptation Information

Simulation 7	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: modified polling)
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 48 bytes.)
Error rate parameters:	OFF

Simulation 8	
Nodes:	19 Nodes, 1 AP
Channel Access Mode:	PCF (polling policy: modified polling)
Simulation Time:	900 secs
FTP Applications:	10 connections
CBR Applications:	10 connections (Rate: 64 Kbps, packet size 48 bytes.)
Error rate parameters:	On a: 30 sec, β : 10 sec^{-1} Good BER: 10^{-10} Bad BER: 10^{-5}

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 a number of examples of ways in which these parameters could be employed to improve the service capabilities of wireless networks.

Our initial examples and quantitative results indicate that the 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 for a complete validation of the proposed QoS parameters. In particular ongoing research is addressing the evaluation of the system using more 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 an application utility model. The goal of this work is to devise distributed algorithms that will allow for maximizing global utility under the time and adaptation constraints 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 are justified when complexity costs are weighted against the possible improvements.

Finally we are particularly 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

[Le 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

[Amir and McCanne] 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.

[Broch, Maltz, and Johnson] 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.

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

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

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

[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*.

[Shenker] 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.

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

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

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

[Stoica, Shenker, and Zhang] 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.

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

[Vandalore and Jain] 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.

[OginoN] 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.

[Kosuga] 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. S enac (Eds.), Interactive Distributed Multimedia Systems and Telecommunication Services, 6th International Workshop, IDMS'99, LNCS 1718, pp.371-376, Springer Verlag, Oct. 1999.

[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

[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

[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>

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

[Queuing] Len Kleinrock has an extensive bibliography on traffic queuing and buffering at <http://millennium.cs.ucla.edu/LK/Bib/> Sally Floyd has information on queue

management and her research on Class-based Queuing (CBQ) at <http://www.aciri.org/floyd/cbq.html> and on RED at [../red.html](http://www.aciri.org/floyd/red.html)

[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

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

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

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

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

[Campbell] 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

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.

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

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.

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

[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 Performance Protocol Architectures, HIPPARCH '94, Sophia Antipolis, France, December 1994, INRIA France.

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

[Riley] 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

[CFK98] Prashant Chandra, Allan Fisher, Corey Kosak, T. S. Eugene Ng, Peter Steenkiste, Eduardo Takahashi, and Hui Zhang. *Darwin: Resource management for value-added customizable network service*. In Sixth IEEE International Conference on Network Protocols (ICNP'98), 1998.

[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/>.

[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

[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.

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

[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

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

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.

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

Rappaport, Theodore, "Wireless Communications, Principles and Practice", Prentice-Hall, 1996

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

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