Evaluation of Admission Policies for Probabilistic Quality of Service (QoS)

Project Report

For the Degree of Master of Science

August 11, 2000

Candidate: Sandeep S. Tamboli

Advisor: Philip Koopman

Department of Electrical and Computer Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213

Evaluation of Admission Policies for Probabilistic Quality of Service (QoS)

Sandeep S. Tamboli

Abstract

Commonly used network applications can have flexible Quality of Service (QoS) requirements. They typically have a minimum requirement for basic service, as well as an additional operating range having increased benefit in exchange for increased resource consumption. In many such cases, it is important to provide a steady, pre-arranged level of service rather than service that may fluctuate wildly from one moment to the next. This work uses a hybrid approach to providing both a hard guarantee for the basic level of service, and a probabilistic assurance of a particular service level for enhanced QoS. The main idea is to size a system to handle the worst case number of sessions of applications at the lowest acceptable service level, but at the same time support higher QoS using any available slack capacity. An admission process provides a probabilistic assurance that any session allocated more than the minimum required resources would run to completion without having to be degraded. Simulation results are used to show the effectiveness of the approach, along with tradeoff between the delivered QoS and the frequency of service degradations.

1. Introduction

1.1 Quality of Service

Quality of Service is defined in different ways in different contexts. In this paper, we use the term in the context of applications that can work at different performance levels along multiple application-quality dimensions. For example, consider a video conferencing application. At the application level, users can express the required quality of service along various dimensions such as frame rate, resolution of a frame or color. Along each dimension, the application can work at different levels of performance. For example, it can have black and white frames, gray scale frames or colored frames (along the color QoS dimension). The application/user requirements are mapped to resource requirements. The system can be built to take care of this mapping or the application profile can itself express its resource requirements for different levels of QoS along different QoS dimensions. Depending on the available resources, the system will admit the application into the system at some operating point. A single level / value for each QoS dimension defines this operating point. Thus the system provides varying levels of guaranteed service to the application. Again the guarantee can cover the whole spectrum of ‘no guarantees’ to ‘hard guarantees’. In this paper, we consider probabilistic guarantees about a service level. This is elaborated further in Section 2.2.

1.2 Terminology

Following are the definitions of some important terms used in the rest of the document.

Session: A bi-directional unicast flow of data between a sender and a receiver. For example, a telephone call can be considered as a session where data is flowing in both directions.

Admission Control: The binary decision of whether to accept or reject a new session request.

Resource Allocation: The decision of how much resource to assign to a new session.

QoS Degradation: The act of degrading the QoS provided to an existing session. The system configures the low-level mechanisms to carry this degradation out and the application adjusts its QoS dimension.

Admission control and resource allocation are actually the steps of a two-step process, that is hereafter called the *Admission Process*. In this work, the second step of resource allocation is more important than the first step of admission control because all session requests are accepted. What really makes difference is what level of resource is allocated to the session after admission. The admission process occurs before the actual data transfer. When the session is over, the resources are released.

The paper is organized as follows: Section 2 describes the problem under consideration. Section 3 gives a brief system description. The measures to evaluate the admission policies are detailed under Section 4. The next section provides some theoretical basis for this work. The simulation results are described in Section 6 whereas a brief idea of other applications of this work is given in Section 7. Related work in this area is covered in the next section. Section 9 offers the conclusions and the last section suggests some future work.

2. Problem Description

This section describes the problem scenario, the admission process, different admission policies, and different degradation criteria and hypothesizes about the results.

2.1 Problem Scenario

Even though a target application scenario is described in this section, this work is generic and other possible applications are discussed in Section 8. This section describes the problem characteristics, and narrows down the scope of the work.

Consider a surface warship. A typical cruiser is approximately 550 feet in length. The crewmembers involve different teams such as an air contact team and an under surface contact team. They might be situated in different parts of the warship. Their workstations are typically connected to a hub and the hubs are connected to each other by point-to-point links. This inter-hub link bandwidth is the resource under consideration in this work. Some of the crewmembers are typically conducting videophone sessions for their communications. In this scenario, the maximum number of sessions possible is known in advance. As the sessions are mainly conducted for war activities or maintenance/troubleshooting activities, all sessions can be critical. Hence, when someone wants to set up a session, it should not be denied resources.

The videophone sessions can work at different levels of quality of service along different dimensions. The different dimensions might be color, frame rate, frame resolution, etc. All of the three dimensions mentioned above translate directly into a bandwidth requirement. To illustrate our purpose, we have considered two levels of resource requirements, but that can be extended to multiple levels. We call the corresponding two service levels the Basic service level (for low resource requirement) and the Premium service level (for high resource requirement). For example, the Basic service level might correspond to black and white frames whereas the Premium service level might correspond to multicolor frames with the same frame rate and resolution. The inter-hub link is sized to be enough for the maximum load at Basic service level. In other words, the resource is not sized for the worst-case requirement of all the possible sessions active at Premium service level. Most of the time, all possible sessions will not be active, and many of the active sessions can enjoy Premium service. But in the rare case of all possible sessions requesting service, none of them need be denied the Basic service.

2.2 Admission Process

The admission process is as depicted in the Figure 2. At the admission time of a session, two actions are taken. One action concerns the incoming session. This action is to allocate a Basic or Premium level of resources to the newly admitted session. The second action concerns the existing sessions. One of the existing Premium sessions may or may not need to be degraded depending on the admission policy and the available resource.

Another view to look at the admission control in this specific scenario is to consider it as having two parts: the first part is admitting the Basic level of a session and the second part is admitting the difference between the Basic and Premium level. The first admission decision is defined to always be ‘yes’ whereas the second admission decision can be ‘yes’ or ‘no’.

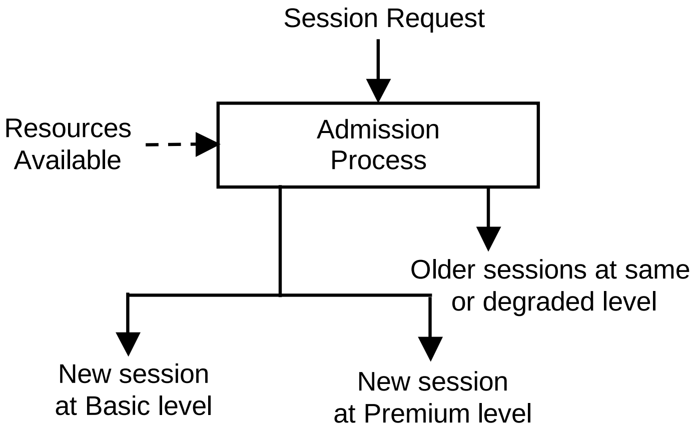
Session Request
Resources Available
Admission Process
Older sessions at same or degraded level
New session at Basic level
New session at Premium level

Figure 2: Admission Process

The resource allocation for a typical session duration can have three possibilities as shown in Figure 3. A session S_i can get admitted at Premium level and remain at that level for its entire duration. The second possibility is that S_i gets admitted at Premium level but gets degraded to Basic level at some instant in its duration. This instant is the admission instant for some incoming session S_k. The third possibility is that S_i gets admitted at Basic level and remains at that level for the entire session duration.

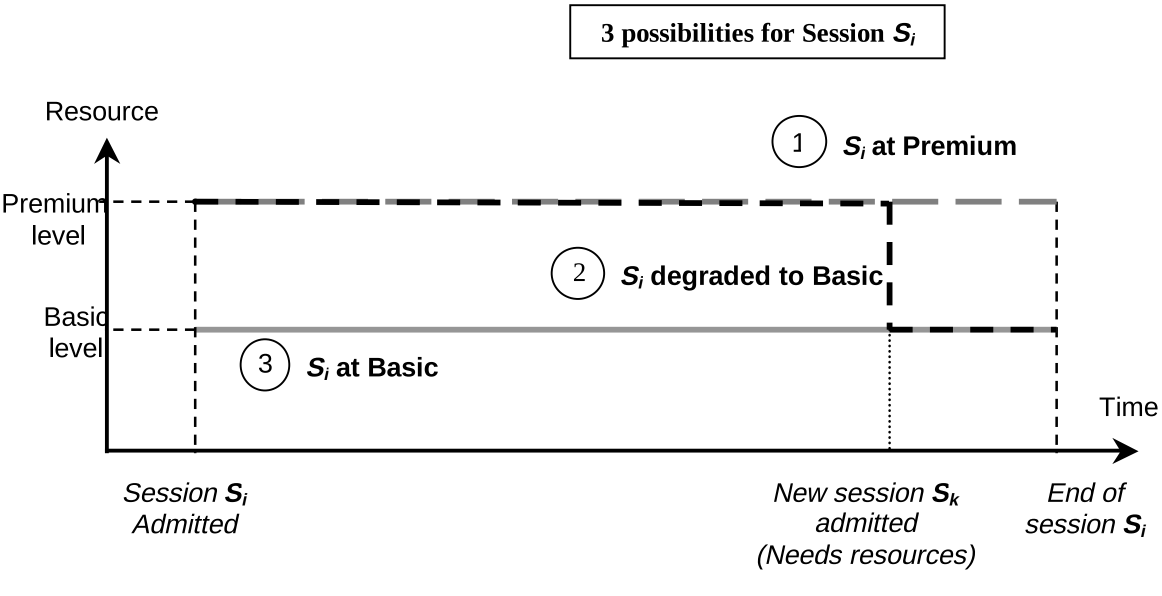

Figure 3: Resource allocation over a session duration

To summarize, the following are the ground rules for the admission process (these ideas are similar across all the admission policies):

1. Basic service level is guaranteed for the entire session duration.
2. There is a probabilistic assurance for Premium service level.
3. Admission control:
    - Always admits a session (at Premium or Basic level).
    - May involve QoS degradation: degrade an existing session from Premium to Basic level.

2.3 Admission Policies

Three admission policies are compared. The first two represent ends of a spectrum while the third policy represents a middle ground.

1. Basic-only: The policy is to admit every session at Basic service level.
2. Greedy: The policy is to admit a new session at Premium service level if enough resource capacity is available at that instant.
3. Conservative: The policy is to admit a new session at Premium service level if enough resource capacity is available after keeping some resource capacity in reserve for future session requests.

For the Conservative policy, there are two variable parameters: how long to look into the future and how much resource capacity to set aside. There are many possible values for the first parameter and we have chosen the time until the first session completion (at *t1*) from now (*t0*). That means, the future interval is (*t1 - t0)*. The following equations govern the second parameter.

- capacity = (Basic level resource) * (#sessions expected)
- #sessions expected = (future interval) * (mean arrival rate)

2.4 Degradation Selection Criterion

In case of overload, at the admission time of a session, one or more of the existing sessions at Premium service level must be degraded to the Basic service level. The following are the possible criteria considered to select a session to degrade.

1. Random: Select any Premium session at random
2. Latest to depart: Select the Premium session that has the longest remaining duration. This will release the amount of resource for longer amount of time at the cost of having one session degraded for a long time.
3. Earliest to depart: Select the Premium session that has the shortest remaining duration. The degraded session might have been on the verge of completion.

2.5 Hypothesis

From the problem description above following hypotheses can be made:

1. The Greedy policy will achieve a higher ratio of Premium admissions than Conservative policy but it will have a higher ratio of QoS degradations.
2. The Conservative policy will result in a lower ratio of QoS degradations than the Greedy policy at the cost of a smaller ratio of Premium admissions.
3. The Basic-only policy will obviously result in no QoS degradations, but also no Premium admissions.

3. System Description

We have considered the system abstractions at higher levels. In order to implement the policies, appropriate lower level mechanisms such as resource reservation, packet classification, packet scheduling, processor scheduling, and buffer management (depending upon the resource under consideration) need to be present. There has been a significant prior research on these issues

[Aurr98] and that part is considered out of scope of this work. In other words, this work assumes the appropriate mechanisms are present in the system to implement policies and carry out QoS enforcement.

The system consists of a single abstract resource. The admission policies are applied to control the resource.

4. Evaluation Measures

This section first describes the fundamental measures and then a composed measure of admission effectiveness.

4.1 Fundamental Measures

Following are two fundamental measures used to evaluate the admission policies.

- *Premium Admission Probability (P):* This measure is defined to be the proportion of the number of sessions admitted at Premium service level with respect to the total number of session requests.
- *Absolute Degradation Probability (D):* This measure is defined to be the proportion of the number of sessions that are degraded with respect to the total number of session requests. The reason it is called absolute will be clear with the definition of the next fundamental evaluation measure.

An additional measure derived from the above two measures is defined as follows.

Relative Degradation Probability(R): This measure is defined to be the proportion of the number of sessions degraded with respect to the number of sessions admitted at Premium service level. This is called *relative* because it gives the relative or conditional probability of degradation given that there was a Premium admission. To differentiate the previous measure from this one, the previous measure is called *absolute*.

At first, the necessity of defining D may not be clear. But if we consider the tradeoff between Premium admissions and QoS degradations, we find that for the Conservative admission policy, a way of not doing degradations is not to admit at the Premium level. To take that into consideration, we devise the measure of D. In fact, the composed measure described next considers D and not R. Considering R would be an injustice to Conservative admission policy that occasionally admits at Basic level. Additionally, R is undefined for Basic-only admission policy.

What is the necessity of R? R explains the degradation phenomenon graphically better than D. In fact, it helps to understand the behavior of D with respect to the resource load. This will be clearer from Figure 5 in Section 6.4

4.2 A Flexible Composed Measure: Admission Effectiveness

The admission effectiveness (E) measure is defined as follows:

E = w(1-D) + (1-w)P

Where:

- E is [0,1]
- w**:** reward [0,1] for not degrading QoS
- (1-w): reward [0,1] for admission at Premium level
- w = 1 => E = 1 - D (Emphasis on not degrading QoS)
- w = 0 => E = P (Emphasis on admitting at Premium level)
- Value of w depends on the application and the users.

Admission Effectiveness indicates the tradeoff of not getting degraded vs. getting a higher QoS level.

Following is an argument that 0 <= E <= 1:

Consider three boundary cases:

All sessions admitted at Premium level and none get degraded

=> $P = 1, D = 0$ => $E = 1$

All sessions admitted at Premium level and all get degraded

=> $P = 1, D = 1$ => $E = 1 - w$ => $E = 0$ if $w = 1$ or $E = 1$ if $w = 0$

All sessions admitted at Basic level

=> $P = 0, D = 0$ => $E = w$ => $E = 1$ if $w = 1$ or $E = 0$ if $w = 0$

Admission Effectiveness considers all the possible relative weightings of P and (1-D). For example, if we want to give four times as much importance to having Premium admissions than not to have degradations, then following equation will give us the value of w to use.

$4w = 1 - w$ => $w = 0.2$

For the opposite case of not degrading having four times the importance,

$w = 4(1- w)$ => $5w = 4$ => $w = 0.8$

Human in the loop [Partridge94]:

Humans definitely like to have a higher quality but they also remember even infrequent failures/degradations in the service. So just providing higher quality is not enough. Efforts should be made to reduce any failures/degradations as far as possible. The weighting factor ‘w’ facilitates this tradeoff. Depending on the target users, a value of ‘w’ can be chosen and accordingly an admission policy can be selected which gives the highest admission effectiveness at that value of w.

5. Theoretical validation by a queuing model

Following is a brief description of a queuing model called the ‘Truncated Poisson’ (M/M/c/c) model found in the literature [Allen92] that is close to the system under consideration. Note that, the term ‘job’ in this section is equivalent to a ‘session’ in our system.

This theoretical model has a memory-less inter-arrival time distribution and service time distribution with multiple servers (c) and a finite capacity of c jobs. It means that it does not enqueue the jobs. If all the servers are busy, then any incoming job is lost. So this model involves consideration of blocking probability, which is given by Erlang's B formula. The difference between the Truncated Poisson model and our work is that our work does not have blocking. In other words, in our work, arrivals do not occur when all sessions are active. The model of our work can be viewed as depicted in Figure 4. The non-conforming arrivals are filtered out at the input dispatcher.

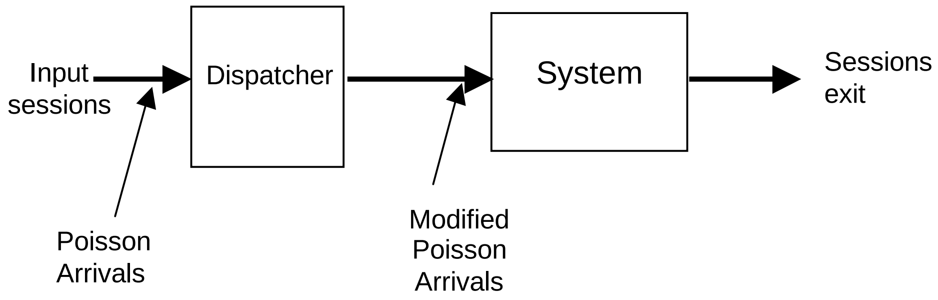

Figure 4: System model similar to Truncated Poisson model

In order to validate the simulations, first the Truncated Poisson model was simulated and the blocking probability results were found to be comparable with the theoretical value. The model used for this work differed in only three lines of code, suggesting that it probably also yields valid results.

6. Simulation Results

This section describes the simulation process in detail along with the simulation results.

6.1 Simulation Methodology

A discrete event toolkit Sim++ [Fishwick95] is used for carrying out the simulation experiments. A typical simulation involves a chain of discrete events sorted according to their occurrence time. As the simulation clock reaches the occurrence time, the particular event routine is executed.

The event scheduling is carried out in the simulator in the following manner. The simulation starts with an initial session arrival that schedules the departure of this session as well as a next session arrival. The next arrival does the same and this process continues as long as the specified number of arrivals has not occurred.

Following are the software components of the simulated system:

1. Allocation module: Allocates resource to the incoming session request depending on the admission policy.
2. Degradation module: Degrades an existing session from Premium level to Basic level. The choice of the session to degrade depends on the degradation selection criterion.
3. Resource availability module: Reports the current resource allocation value. As this work assumes constant bit rate sources, the actual instantaneous resource usage is equal to the resource allocated/assigned. In case of variable bit rate sources, this will not be the case and some measurement-based technique must be employed to get the actual resource usage. In fact, measurement-based admission control algorithms use the same procedure [Jamin96] while admitting or rejecting an incoming request.

The resource is divided into multiple channels or facilities. No actual videophone session establishment is simulated. It is only assumed that the session requests arrive with the given distribution. No actual data transfer after the establishment is simulated.

Simulations were carried out with five different random number seeds, each for one million session requests. The fundamental evaluation measures were computed for each simulation and their mean was determined. It was observed that with this long run, the sample variance was negligible and hence confidence interval analysis was not deemed necessary.

6.2 Example Topology

This work is limited to a single generic resource. As mentioned in Section 2.1, a link between two hubs (H1 and H2) is considered as the resource under consideration.

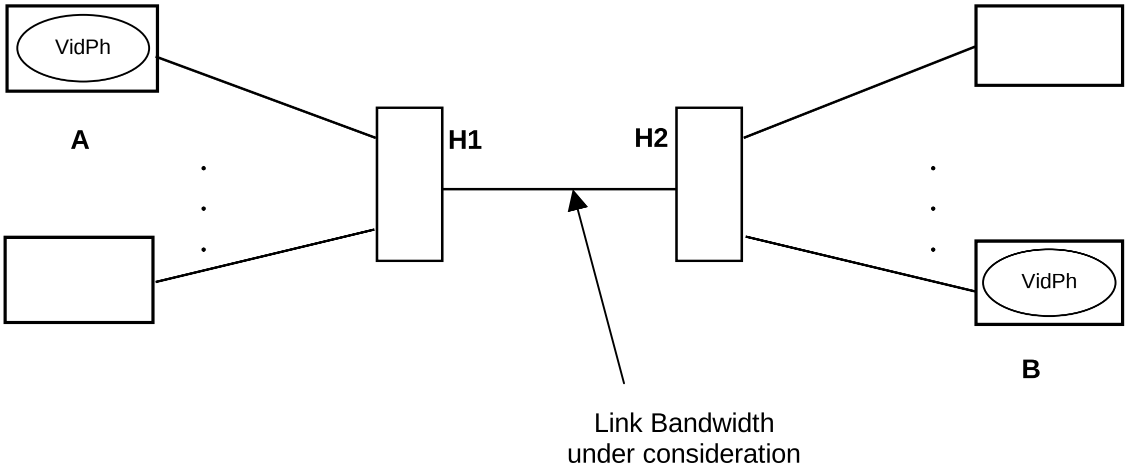

Figure 4. Example Simulation Topology

6.3 Simulation Parameters

Consider the example topology, with the units of the values for the simulation parameters being that of the link bandwidth resource. Actually, more abstract quantities such as ‘100 units of bandwidth’ could be used, but to make the results more concrete, the unit of megabits per second has been chosen.

The simulation parameters are divided into two categories:

1. Fixed: These are kept at a fixed value.
    - Total bandwidth: 100 Mbps
    - Maximum number of concurrent sessions: 50

- Mean session duration: 1000 seconds
- Bandwidth requirements for all session requests:
    - Basic service level: 2 Mbps
    - Premium service level: 4 Mbps

2. Variable: This parameter is varied in order to vary the offered load on the system.
    - Mean arrival rate: From 0.01 session requests per second to 1 session request per second.

At high offered load, very few arrivals are dispatched to the system because at the time of most of the arrivals, all sessions are active. For example, for mean arrival rate of 1 session per second, only 4% of the Poisson arrivals are dispatched to the system. But in these cases, simulation duration is scaled accordingly so that 1 million actual system arrivals occur.

The bandwidth required for a videophone session is the product of frame rate, frame resolution, and color depth in number of bits per pixel. The following are the calculations to show the feasibility of the bandwidth values chosen in these simulations, corresponding to the Basic and Premium service level. For Basic service level:

8 bits/pixel * 250X200 pixel frame * 5 frames per second = 2 Mbps.

For Premium service level:

16 bits/pixel * 250X200 pixel frame * 5 frames per second = 4 Mbps (keeping the frame rate and frame resolution constant).

6.4 Results

Figure 5 shows the behavior of the fundamental measures. Even though the plot only shows the behavior of the Greedy policy for clarity, the corresponding curves for the Conservative policy are also similar. The X-axis represents the offered load in terms of session request rate (in sessions per second). As the session duration has a fixed distribution with a fixed mean, any

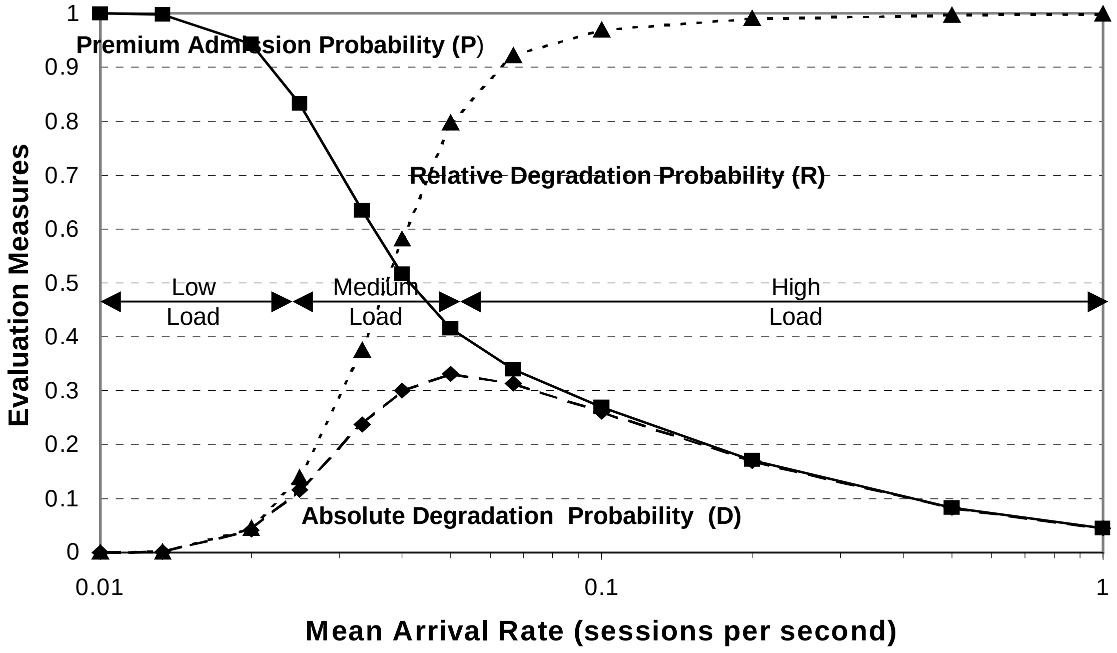

Figure 5: Fundamental evaluation measures (Greedy Policy)

increase in arrival rate increases the load on the system. The load is approximately divided into three categories:

1. Low load: less than 25 sessions are active in the system on average.
2. Medium load: more than 25 (but not all) sessions are active on average
3. High load: Almost all the sessions are active (note that all the mean arrival rates equal to and greater than 0.05 sessions per second fall in this category).

The Y-axis shows the fundamental measures that all lie between 0 and 1, being probabilities.

P decreases monotonically as the load increases. That is expected because the admission policy can admit less and less sessions at Premium service level. R increases monotonically as the load increases because more and more sessions get degraded. At very high load, almost all of the Premium sessions get degraded and hence R reaches almost 1. D can be considered as the product curve of the above-mentioned curves. Hence it increases initially with the increasing load but later becomes limited by the P curve. The point where it changes the slope, that is, the peak is an artifact of its definition.

After the arrival rate of 0.05 sessions per second, the system is generally fully loaded. We have considered the arrival rates up to 1 session per second to see what happens if requests, even though limited in number, arrive very fast. As seen from the plots, P continues to worsen. But as D is bounded by P, it almost coincides with P. So what is happening is almost all of the sessions that get admitted at Premium service level, get degraded at some point in their lifetime whereas the proportion of sessions admitted at Premium level itself is very low.

An obvious question arises: if we can derive D as a product curve or derive R as a division curve, why do we have both of them? Actually, D is the measure that is used in the computation of the composed measure of Admission Effectiveness. But R is plotted as it is more intuitive to understand and also it helps in better understanding of D. The reason D is chosen and not R is R just considers degradations with respect to Premium admissions. It does not do justice with an admission policy that occasionally admits sessions at Basic level to keep degradation low.

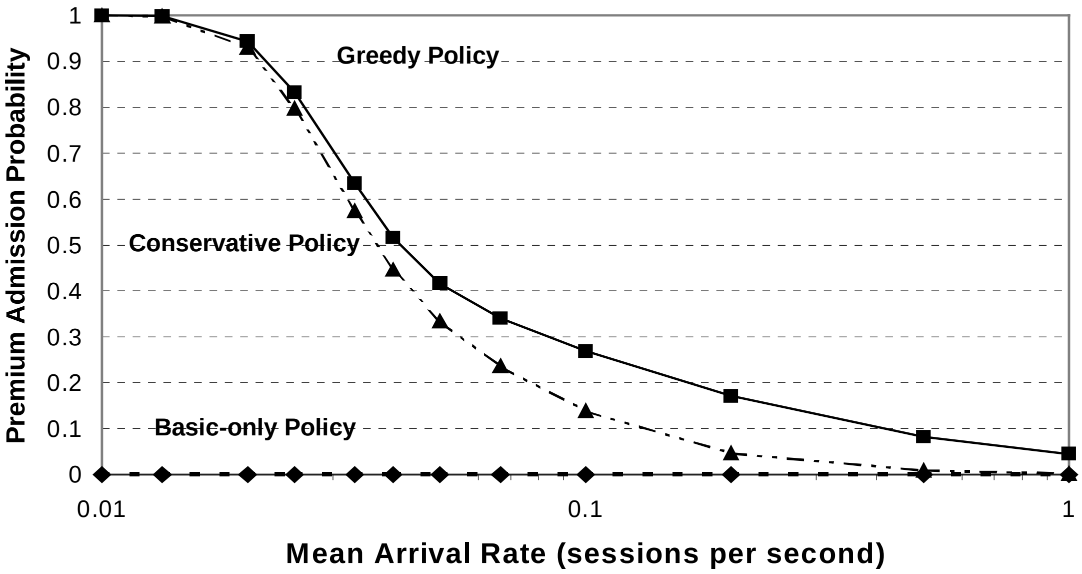

Figure 6: Premium Admission Probability

Figure 6 shows the comparison of the three admission policies with respect to P. The Greedy policy has better P than the Conservative because the policy just gives away the Premium level resources if they are available at the admission time. The Basic-only policy has P of 0 for all the load values.

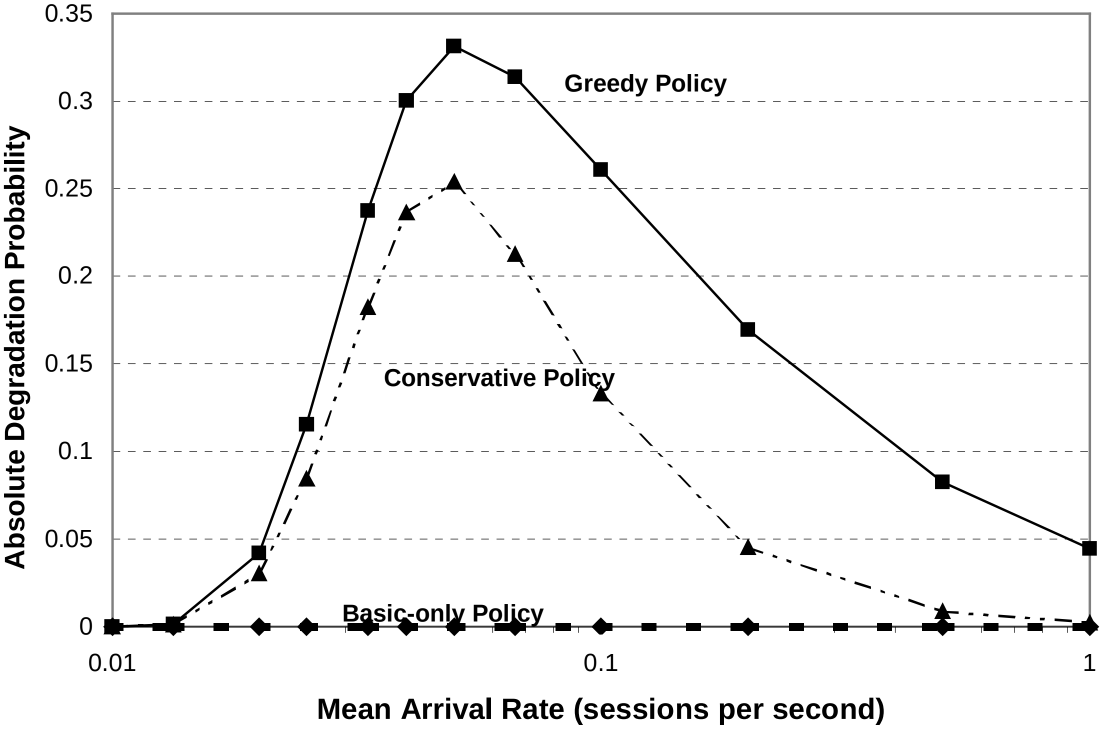

Figure 7: Absolute Degradation Probability

Figure 7 shows the comparison of the three admission policies with respect to D. The Conservative policy performs better than the Greedy policy. It induces fewer degradations because of keeping resources in reserve for the future incoming requests, so that when the future requests do arrive, they do not cause any session to degrade.

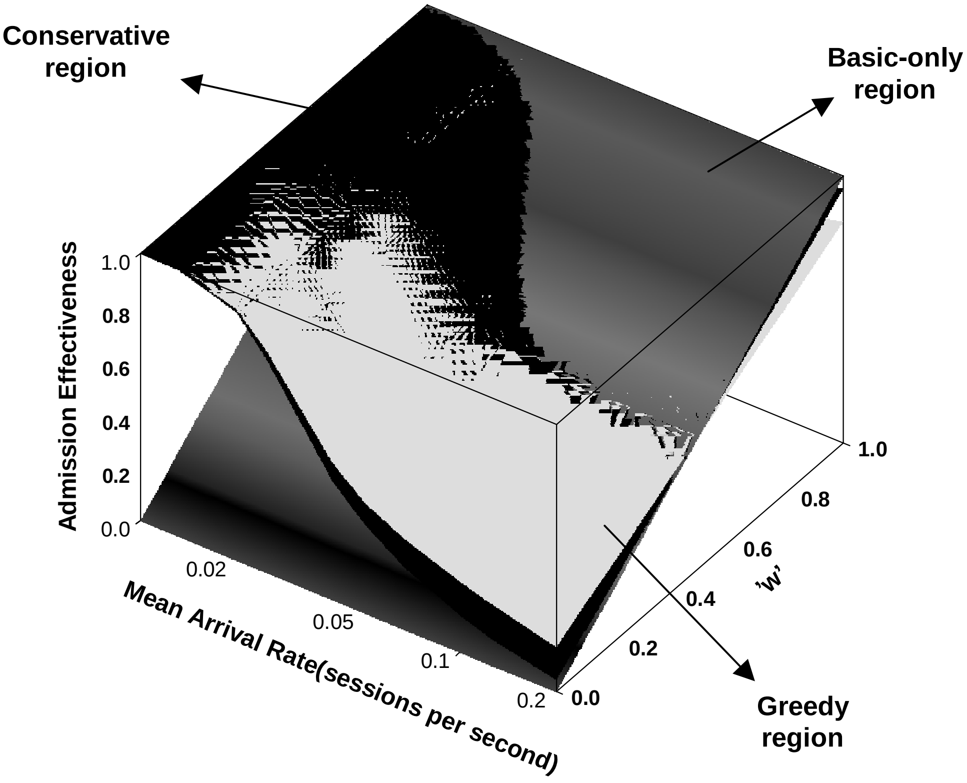

Figure 8: Admission Effectiveness regions

Figure 8 shows the 3-D plot of Admission Effectiveness versus offered load versus 'w'. Before describing this graph, let us define two ranges of values of w. Let us call the range 0 <= w <= 0.5 as the 'P range' as this range gives higher reward to the Premium admissions than to not having degradations. Let us call the range 0.5 <= w <= 1, as the 'D range' as higher reward is given to not having degradations. (Note: w=0.5 is present in both ranges as the reward is equal for both, having Premium admissions and not having degradations)

At low load, the Greedy and Conservative policies perform almost the same for the D range. Because of the variability of the session request pattern and the session duration pattern, there are some times when more than 24 (the maximum number of sessions that can exist all at the Premium level with the Conservative policy) sessions are active in the system. In those cases, the Conservative policy admits fewer sessions at Premium service level and thus has less admission effectiveness than the Greedy policy. But that difference is very small.

For medium and high load, and for the P range of w (i.e. w <= 0.5), the Greedy policy is the most effective of all. That is clear from the Figure 6 as the Premium Admission Probability is the highest for the Greedy policy.

For low load and for D range of w (i.e. 0.5 <= w <= 1), the Conservative policy is the most effective. As the value of w approaches 1, the Basic-only policy becomes the most effective. Towards the higher end of D range and medium load, the Basic-only performs the best but in the case of medium load and lower end of D range, the Conservative policy performs the best. For high load and D range, Basic-only performs the best. As more and more importance is given to not having degradations over having Premium admissions, the Basic-only policy performs better and better as it produces no degradations at all.

Thus we get different regimes of operation and depending on the value of w and the offered load, a suitable admission policy can be chosen. In the case of changing conditions, the policies can be switched if the system is going from one regime to another.

If we consider an over-sized system (where some sessions are at the Premium level when all sessions are active), then the Basic-only region gets reduced as the Conservative region pushes it to the right in the D range. That happens because, for a given load, D decreases and P increases. Also, the Greedy region starts becoming more effective at the lower end of D range for low and medium load. The extreme case is when the system is sized for worst case, that is, sized such that all sessions can be active at Premium service level. Then the entire region is occupied by the Greedy policy as there is no harm in allocating the Premium service level to the incoming sessions.

The results are in accordance to the hypotheses made in Section 2.5. Now not only do we know qualitatively which policy works better when, but also the approximate boundaries of the regions where a policy is the best choice.

7. Other Applications

- *Video on demand*: The video server can be sized to provide the maximum number of possible sessions in a Basic mode while some video sessions enjoy the Premium level when all sessions are not active.
- *ISP link to the backbone:* See Figure 9 for the link connecting a modem router to the Internet Service Provider (ISP) backbone. We know how many maximum number of modem sessions (N) we need to support. If we assume, the Basic level = 28 Kbps; the Premium level = 56 Kbps; then the link bandwidth = N * 28 Kbps can support all active sessions at 28 Kbps. If all modems are not active, the link can support some sessions at the Premium level.

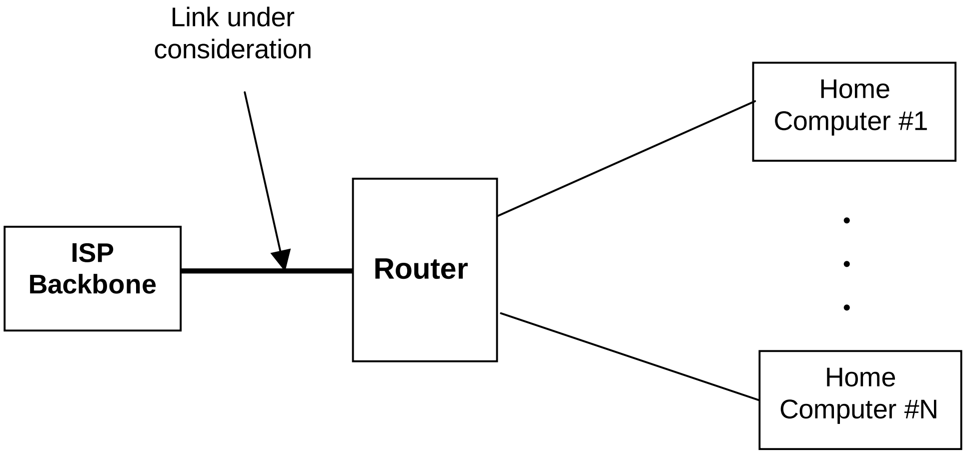

Figure 9: Modem router’s link to the ISP backbone

8. Related Work

There has been a tremendous amount of work in the area of QoS in general and admission control and resource allocation in particular. The idea of graceful degradation is not new. But a combination of the ideas (for a closed system) of QoS degradation and trading off degradation frequency against probability of achieving higher level of QoS and an admission policy of keeping reserves is not found in the literature.

[Cheng98] discusses QoS degradation of low priority clients in case of overload. A new client is accepted if the negotiated QoS requirement can be accommodated with the remaining capacity or enough room can be made by lowering the QoS of existing low priority clients. It considers the problem of reward optimization to come up with a proportion of resource capacity for low priority clients and high priority clients. The high priority clients get guaranteed QoS whereas the low priority clients get best effort (may be constantly fluctuating) QoS. A QoS degradation has no explicit penalty, only the reward to the system is reduced. The admission of any client is always at the higher QoS level. The admission policy is greedy.

[Lee99] involves setting aside a portion of the resources as reserves and manages it to maximize the total system utility. This policy is similar in some respects to our Conservative policy. This work does not consider QoS degradation of running sessions but it can admit sessions at the lower QoS level.

[Jamin96] compares different admission control algorithms. Our Greedy policy is similar to their Simple Sum algorithm. In fact, these algorithms can be considered as a different ways of implementing an admission policy. Thus these algorithms are at a lower level than the admission policies considered in this paper.

[Chen98] proposes and analyzes admission control algorithms based on reward optimization for on-demand multimedia servers. The algorithms are developed based on ‘QoS control’ and ‘reservation control’. The system is given a reward if a client is successfully served and a penalty is imparted if a client is rejected or is not satisfied with the QoS delivered. But there are no renegotiations (that is, run-time QoS changes as carried out in this work) and the admission policy is greedy. Their analysis is also at the session level and they abstract the server capacity as if having N capacity slots.

9. Conclusions

There is a tradeoff involved in accepting sessions at the Premium service level and not degrading an existing session from the Premium service level for a closed system. The Greedy admission policy gives a preference to Premium admissions whereas the Conservative admission policy gives a preference to not degrading a session's service. For an exact sized system, at high load, these policies degenerate into the Basic-only policy.

Table 1 lists the guidelines for selecting an admission policy.

Admission Policy	**Useful when**	**Note**
Basic-only	High penalty for degradation and high load	Usually wastes capacity, no degradations
Conservative	High penalty for degradation and low load	Moderate efficiency, moderate risk of degradation
Greedy	High reward for Premium admissions and all loads	High efficiency, but high risk of degradation

Table 1: Guidelines for selecting an admission policy

10. Future work

Future extensions can be explored along four dimensions:

Generalizations: More service levels and sessions with variable service demand could be studied. More penalty-reward functions could be used.

More realistic workload: Actual video sessions are never a constant bit rate. Compression introduces variability in the bit rate. If the bit rate is specified by an average then there might be some instants when because of a burst, some data is lost. If the bit rate is specified by its peak, then most of the time, some part of the bandwidth allocated to the session will be unutilized. A token bucket traffic shaper may need to be used to allow some bursts in the traffic.

Restorations: Changing the QoS of a degraded session back to the previous level introduces complexity, but some simplified analysis needs to be done in this regard.

Sophisticated Predictions: Instead of using the input distribution for predictions, sophisticated prediction algorithms can be used to set aside resources for future session requests. That may improve the performance of the Conservative policy.

Acknowledgements

The author would like to thank his advisor Dr. Philip Koopman for suggesting the basic idea behind this work and providing an able guidance throughout the course of this work. The author also acknowledges the able guidance provided by Dr. Daniel P. Siewiorek, author's M. S. committee member. The author would also like to acknowledge the valuable help offered by Carol Hoover, Dr. Jeffrey Hansen, Dr. John Lehoczky, and Dr. Ragunathan Rajkumar for this work.

This research was supported by the Quorum Program of the Defense Advanced Research Projects Agency (DARPA) under agreement number N66001-97-C-8527.

References

[Allen92] Allen A., "Probability, Statistics, and Queuing Theory with Computer Science Applications", 2nd Ed., Morgan Kauffman Publishers, 1992.

[Aurr98] Aurrecoechea C., Hauw L., Campbell C., "Survey of QoS Architectures", ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6 No. 3, May 1998, pp. 138-151.

[Chen98] Chen I., Hsi T., "Performance Analysis of Admission Control Algorithms Based on Reward Optimization for Real-Time Multimedia Servers", Performance Evaluation, Vol. 33, No. 2, 1998, pp. 89 - 112.

[Cheng98] Cheng S., Chen C., Chen I., “A study of self adjusting QoS control schemes”, Proceedings of IEEE Winter Simulation Conference, Washington D.C., Dec 1998, pp.1623-1628.

[Fishwick95] Fishwick P., “Sim++ User Manual”, University of Florida, Gainesville, 1995.

[Jamin96] Jamin S., “A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks”, University of South California Tech Report, USC-CS-96-639, 1996.

[Lee99] Lee W., Srivastava J., Sabata B., “Quality of Service Negotiation and Admission Control with a Reserves-Based Strategy for Multimedia Applications”, Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, June 1999 pp. 147-152.

[Partridge94] Partridge C., “Gigabit Networking”, Addison-Wesley Publishing, 1994, pp. 178-179.