

Anne Nevin

The Uncertainty of Decisions in Measurement Based Admission Control

Thesis for the degree of Philosophiae Doctor

Trondheim, November 2010

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Telematics



NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

© Anne Nevin

ISBN 978-82-471-2423-9 (printed ver.)

ISBN 978-82-471-2424-6 (electronic ver.)

ISSN 1503-8181

Doctoral theses at NTNU, 2010:218

Printed by NTNU-trykk

Abstract

Most real-time voice and video applications are delay/loss sensitive but *relaxed* in the sense that they can tolerate some packet loss/delay. Using this information, network utilization can be greatly improved by exploiting statistical multiplexing.

To this end, Measurement Based Admission Control (MBAC) has for a long time been recognized as a promising solution. MBAC algorithms, do not require an *a priori* source characterization which in many cases may be difficult or impossible to attain. Instead, MBAC uses measurements to capture the behavior of existing flows and uses this information together with some coarse knowledge of a new flow when making an admission decision for this requesting flow.

The number one requirement of MBAC to be successful is that it can *robustly* provide Quality of Service (QoS) to the accepted flows. Being robust means that MBAC must be able to withstand a sudden increase in the number of users trying to access the network, handle applications with various capacity requirements and handle an aggregate rate that may change in a highly unpredictable manner. These robustness issues become challenging since MBAC relies on erroneous measurements.

Measurements are unavoidably inaccurate. This imperfection creates uncertainties which affect the MBAC decision process. The degree of uncertainty depends on flow characteristics, the length of the observation window and the flow dynamics. Flows will be accepted when they should have been rejected, *false acceptance*, and rejected when they should have been accepted, *false rejections*. For the service provider, false rejections translate into a decrease in utilization and for the end user, false acceptance means that the QoS of the flow can no longer be guaranteed. Basing admissions on measurements clearly requires the understanding of the measurement error and how this impacts the performance of MBAC.

This thesis considers the uncertainty of the MBAC admission decision process and describes a methodology for analyzing measurement errors and the resulting performance of MBAC. When studying the performance of MBAC, the key is to focus on the time-scale over which measurements are collected and the admission decision is made. This is in contrast to the infinite time-scale used when evaluating the performance of MBAC with respect to utilization and loss/delay probabilities.

We find how the uncertainty in the measurements vary with the length of the observation window. Non-homogeneous flows cause increased complexity for the MBAC decision algorithm and also for the estimation process. The concept of *similar flows* is introduced, which is a restriction to simplify the analytical

expressions in a non-homogeneous flow environment.

The *probability of false acceptance* can be reduced by adding a slack in bandwidth. When determining the size of this slack, the service provider is confronted with the trade-off between maximizing *useful traffic* and reducing *useless traffic*. We show how the system can be provisioned to meet predetermined performance criteria. This work is fundamentally different from any previous work concerning MBAC and opens up for new thinking and methods for analyzing MBAC performance.

Preface

This thesis is submitted in partial fulfillment of the requirement for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU). During the PhD work, I have been hosted and funded by the Centre of Quantifiable Quality of Service in Communication Systems, Centre of Excellence (Q2S). Q2S is funded by the Norwegian Research Council, NTNU and Uninett. The PhD study was formally conducted at the Department of Telematics, NTNU and Professor Peder J. Emstad has been the supervisor of this work. In addition, Professors Yuming Jiang and Øivind Kure have been co-supervisors.

During my years at Q2S, there are several people I would like to thank. First and foremost is Professor Peder J. Emstad for his outstanding skills as a supervisor and mentor. Without his feedback, advice, patience and wisdom, this work would not be possible. I have also been very fortunate to have Yuming Jiang as an additional supervisor who has provided me with much insight, been extremely motivating and always optimistic. His enthusiasm gave me the drive to keep going in times when I was ready to give up. I would also like to thank Guoqiang Hu for being a co-author and his valuable support and many fruitful discussions.

I thank all former and current people working at Q2S and in addition several people at the Department of Telematics for creating a very enjoyable work environment. Special thanks go to Astrid Undheim and Laurent Paquereau for all the help with L^AT_EX related problems, to Anniken Skotvoll and Mette Veronica Olsen for their great handling of administrative matters, and the technical staff, Hans Almåsbygg for providing a reliable working environment.

I would like to express my gratitude to my parents Arne Tyssø and Veslemøy Tyssø for reading through the manuscript, providing me with valuable comments and unconditional support.

Finally, I am thankful for my wonderful friends and family, especially my husband Eric and my children Selma and Sven that make everyday so enjoyable, inspiring and meaningful.

Symbols

<i>symbol</i>	<i>description</i>
A	offered flow load, the Erlang load
A_{useful}	carried useful traffic
$A_{useless}$	carried useless traffic
a_n	acceptance probability in state $N = n$
B	burstiness
CV	coefficient of variation
c	the system capacity
$K(t)$	flow rate process
l	number of levels
N	state variable indicating the current number of flows in the system
$N(t)$	flow process of the aggregation of homogenous flows
n_{max}	the maximum number of flows a system can handle
$R(t)$	the aggregate rate process
$E(R)$	mean aggregate rate
\hat{R}	estimate of the mean aggregate rate
t	index to represent time
p	activity parameter of the ON-OFF process
r	peak rate of the arriving flow
T_L	flow lifetime distribution
uc	maximum average rate the system can handle
w	measurement window size
P_{Facc}	probability of false acceptance of a flow
P_{Frej}	probability of false rejection of a flow
P_B	blocking probability
$\rho(\tau)$	auto-covariance of the flow rate process with lag τ
$\Psi(\tau)$	auto-correlation of the flow rate process with lag τ
$1/\alpha$	mean time the ON-OFF process is off
$1/\mu$	mean flow lifetime
$1/\beta$	mean time the ON-OFF process is on
δ	measurement error
λ	arrival rate of flows
ξ	mean rate of the flow rate process
σ^2	variance of the flow rate process
$\zeta^2(w)$	variance of the time average
$\hat{\sigma}(w)$	estimated variance
$\theta^2(w)$	variance of the estimated variance

Table of Contents

1 Thesis Introduction	1
1.1 Motivation	1
1.2 Thesis Outline	2
1.3 Contributions	4
1.4 Papers by the Thesis Author	5
1.5 Publications where the Author of the Thesis has Contributed	6
2 Background	7
2.1 Quality of Service	7
2.2 Multiple Time-scale Traffic control	10
2.3 Admission Control	12
2.4 Admission Control Algorithms for Statistical QoS	13
2.5 Measurement Based Admission Control	17
2.6 Robustness Issues	22
2.7 Evaluating the performance of the MBAC Admission Decision	28
2.8 Previous work on Measurement Error	29
3 Measurement Error and MBAC Performance, Concepts and Definitions	33
3.1 The MBAC Behavior	34
3.2 Traffic Modeling	35
3.3 Traffic Classes and System State	37
3.4 Measurements	38
3.5 Measurements to Assess the Mean Aggregate Rate	40
3.6 Measurement Uncertainty	40
3.7 Flow Level Traffic Concepts and Performance Measures	41
3.8 Rate model: The ON-OFF source model	43
3.9 Simulation	45
4 Quantifying the Uncertainty in Measurements	47
4.1 System Model and Assumptions	47
4.2 Measurement Error and Provisioning	48
4.3 Case Study using MMRP sources	50
4.4 Comparison with Simulation	52
4.5 Conclusion	56

5	MBAC: Impact of the Measurement Error on Key Performance Issues	57
5.1	Flow Level and Performance Measures	57
5.2	Provisioning	60
5.3	Case study using MMRP source models	62
5.4	Conclusion	67
6	MBAC and Performance at the Flow Level: A Simulation Study	69
6.1	Description of MBAC and MBAC Strategies	71
6.2	The Simulation Setup	72
6.3	Flow Arrival Rate and Admission Decision	73
6.4	Conclusion	75
7	MBAC: The Measurement Error with Non-Homogenous Flows	77
7.1	System Assumptions and the Concept of Similar flows	77
7.2	Ideal Admission Controller and the Stochastic Knapsack	78
7.3	Measurement Error and Similar Flows	80
7.4	Similar Flows Generated by MMRP Sources	83
7.5	Conclusion	87
8	Measurement Error when the Variance is Unknown	89
8.1	Estimating the variance	89
8.2	Case Study with ON-OFF Sources	90
8.3	Conclusion	93
9	Concluding Remarks	95
	Bibliography	97

Chapter 1

Thesis Introduction

In Measurement Based Admission Control (MBAC), the decision of accepting or rejecting a new flow is based on measurements of the current traffic. The problem with measurements, is that they are unavoidably inaccurate. This imperfection creates uncertainties which affect the MBAC decision process. Flows will be accepted when they should have been rejected and rejected when they should have been accepted.

This thesis addresses the question: *How will the uncertainty in the admission decision affect MBAC performance?* To answer this question, the thesis describes a methodology for analyzing measurement errors and the resulting performance of MBAC.

1.1 Motivation

The Internet with the current best effort service is facing tremendous pressure from new applications that demand Quality of Service (QoS). Real-time multimedia applications have stringent *end-to-end* delay requirements and cannot respond to varying network conditions in the same way as more traditional data traffic. A small burst in demand at peak hours will increase queue build up in routers, resulting in an unacceptable total end-to-end delay of packets. The application will be of non-satisfactory quality, the customers become unhappy and eventually the service providers lose their revenues. With their non-adaptable behavior and in addition having the potential of consuming a major part of the network capacity, multimedia applications are also a threat to the stability of the current network.

In times of network overload, *admission control* is needed, where users are denied access. This will protect QoS of already accepted applications and also the network stability.

The main function of admission control is only to admit a new flow if the QoS requirement can be assured for the requesting flow and all flows already admitted. In addition, the network capacity should be used as efficiently as possible.

Real-time multimedia applications transmit packets at a highly variable rate and in most cases, they can handle some small packet delay/loss. By letting the

1.2. Thesis Outline

instantaneous aggregate rate exceed the link capacity with a small probability, network utilization can be greatly improved.

To take advantage of this statistical multiplexing of flows, the admission controller must somehow receive detailed information about the traffic characteristics. It is hard if not impossible for the user or an application to know the required traffic characteristics before establishing a connection.

The solution is to use measurement based admission control (MBAC), which instead uses measurements of the aggregate rate to capture the behavior of existing flows. When a new flow requests admission, the admission decision is based on the measurements and some coarse knowledge of the requesting flow.

Though increase in utilization has been the main motivator of MBAC, the foremost requirement of MBAC to be successful is to *robustly* provide QoS to the accepted flows. Being robust means that MBAC must be able to withstand a sudden increase in the number of applications trying to access the network, handle applications with various capacity requirements and handle an aggregate rate that may change in a highly unpredictable manner. These robustness issues are challenging since MBAC relies on measurements.

The problem with measurements is that *measurement errors* are unavoidable and the true value just an abstract concept. The errors create uncertainties that abate with the length of the observation window. This uncertainty affects the admission decision in terms of *false rejection* (rejecting a flow when it should have been accepted) and *false acceptance* (accepting a flow when it should have been rejected).

For the service provider, a false rejection translates into a decrease in utilization, and for the end user, a false acceptance means that the QoS can no longer be guaranteed. Basing admission on measurements clearly requires the understanding of the measurement error and how this impacts the performance of MBAC.

1.2 Thesis Outline

This thesis is organized as follows:

Chapter 2: Background. This chapter starts with an introduction to QoS and QoS provisioning before introducing MBAC and a review of some proposed MBAC algorithms. MBAC and its robustness issues are then addressed. There is a clear behavioral difference between MBAC and its counterpart, the *ideal* controller. This chapter seeks to give a clear view of this difference and highlights problems with the current procedures of determining MBAC performance.

Chapter 3: Measurement Error and Performance Analysis. In this chapter, we systematically introduce the system, the MBAC algorithm and assumptions made in this thesis. We describe the observation process and how the measurement error is characterized. For performance analysis, new performance measures that can capture the performance of the admission

decision are introduced. These performance measures become important concepts for understanding the MBAC behavior.

Chapter 4: Quantifying the Uncertainty in Measurements. Measurement errors become significant because the measurement window size is limited. This chapter focuses on the probability of false acceptance due to the uncertainty of the measured average rate when the flows are homogenous. In the analysis, the flow level dynamics such as arrival rates and flow lifetimes are not considered and it is assumed that no flows are admitted or depart from the network during the measurement window. To make up for the measurement error, the reserved bandwidth for the flows must be reduced by some slack. The size of this slack depends on the flow rate characteristics and the measurement window size. In this chapter, we derive the fundamental formula for determining the probability of false acceptance and a thorough analysis is given. The chapter is an extended version of Paper [C].

Chapter 5: MBAC: Impact of the Measurement Error on Key Performance Issues. There is a tradeoff between rejecting too many flows thus wasting resources, and accepting too many flows resulting in QoS violations. In this chapter we study how the measurement errors and flow dynamics impact the performance of MBAC in terms of the performances measures defined in Section 3.7. An example shows how the system can be provisioned with predefined performance criteria. This chapter is an extended version of Paper [D].

Chapter 6: Robustness Issues: A Simulation Study. In this chapter, we elaborate on the flow level performance measures studied in Chapter 5. The impact of multiple arrivals on the MBAC performance is investigated and robustness issues related to changes in the offered flow load are addressed. In the literature, the so called *Peak-rate* strategy and the *Back-off* strategy are often used to make the admission controller more robust to high flow loads. With the flow-level performance measures we gain insight into how these strategies affect the admission decision and how they can enhance MBAC performance. This chapter is based on ideas from Paper [A] and Paper [B] together with the knowledge gained in Paper [D].

Chapter 7: MBAC: The Measurement Error with Non-Homogenous Flows. Assuming that flows are homogeneous (i.e. belong to the same class) is very restrictive, even if flows are of same type e.g only video applications. Non-homogeneous flows cause increased complexity for the MBAC algorithm and also the measurement process. This chapter introduces the concept of *similar flows*, which is a restriction to simplify the analytical expressions in a non-homogeneous flow environment. Similar flows share a common correlation structure and the error analysis becomes straightforward. In contrast, without this restriction, the correlation structure of each flow must be used which again results in a more complex analysis. An example is given to demonstrate the concept. This chapter is directly based on Paper [E].

1.3. Contributions

Chapter 8 The Measurement Error when the Variance is Unknown

When the auto-covariance is known, the uncertainty of the measurement can be stated up-front. In this chapter, we shall see how the uncertainty of the measurement error can be found when the variance of the sources is unknown. This chapter also motivates the use of similar flows as the analysis can be greatly simplified by assuming that the auto-correlation of the flow rate processes is known.

Chapter 9: Concluding Remarks. This chapter contains a summary of the main results and conclusions of the thesis.

1.3 Contributions

The focus of the thesis is on the estimation process and the inherent measurement errors and how these errors impact the admission decision. This thesis is neither about finding a better MBAC algorithm nor answering the question of proper QoS provisioning. We have set up an analytical framework in a very simplified network environment. This is done to make the analysis as tractable as possible. The main contributions are:

- The derivation of analytical expressions to determine the characteristics of the measurement error.
- An analytical framework to evaluate the measurement error and the impact the error will have on MBAC performance.
- The definition of flow level performance measures that specifically target the MBAC decision process. These performance measures open up new ways of analyzing MBAC performance. Not only do these measures make it possible to study the impact of measurement errors, but they can also be used to study other MBAC features specific to a certain MBAC implementation.
- The concept of *similar flows* simplifies the error analysis with non-homogeneous flows. Similar flows share a common correlation structure and the error analysis becomes straightforward. In contrast, without this restriction, the correlation structure of each flow must be used which again results in a more complex analysis.

1.4 Papers by the Thesis Author

This thesis is based on papers written under supervision and in cooperation with Professor Peder J. Emstad. In addition Professors Yuming Jiang and Øivind Kure have been co-supervisors. Solid background on MBAC was gained by contributing to MBAC related research where Professor Yuming Jiang was the main author. During this work the MBAC robustness issue was discovered which resulted in the early work [A] and [B]. The main results of this thesis are based on paper [C], [D] and [E].

Papers Published by the Author

- [A] Anne Nevin, Yuming Jiang, and Peder J. Emstad. “MBAC Robustness to Traffic Variations: A Simulation Study.” In *Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN)*, Toulouse, France, February 2007.
- [B] Anne Nevin, Yuming Jiang, and Peder J. Emstad. “Robustness Study of MBAC Algorithms.” In *Proceedings of the IEEE Symposium on Computers and Communications in (ISCC)*, Shanghai, Marakesh, Morocco, July 2008.
- [C] Anne Nevin, Peder J. Emstad, Yuming Jiang, and Guoqiang Hu. “Quantifying the Uncertainty in Measurements for MBAC.” In *Proceedings of the 15th International Workshop –The Internet of the Future (EUNICE 2009)*, Barcelona, Spain, September 2009.
- [D] Anne Nevin, Peder J. Emstad and Yuming Jiang. “MBAC: Impact of the Measurement Error on Key Performance Issues.” In *Proceedings of Networked Services and Applications-Engineering, Control and Management, EUNICE 2010*, Trondheim, Norway, June 2010.

Paper Submitted for Publication

- [E] Anne Nevin and Peder J. Emstad. “MBAC: The Measurement Error with Non-Homogeneous Flows.” *Submitted for publication, 2010*.

[C]:Guoqiang Hu participated in the discussions and contributed to ideas in this paper.

1.5. Publications where the Author of the Thesis has Contributed

1.5 Publications where the Author of the Thesis has Contributed

The author has contributed to papers related to MBAC and much of this work has provided the in-depth knowledge for writing this thesis.

- [H] Yuming. Jiang, Peder J. Emstad, Victor Nicola, Anne Nevin “Measurement-based admission control: A revisit.” In *Proceedings of the 17th Nordic Teletraffic Seminar (NTS-17)*, Oslo, Norway, September 2004.

- [I] Yuming. Jiang, Peder J. Emstad, Anne Nevin, Victor Nicola, Markus Fidler “Measurement-based admission control for a flow-aware network.” In *Proceedings of the the 1st EuroNGI Conference on Next Generation Internet Networks - Traffic Engineering(NGI '05)*. Rome, Italy, April, 2005.

- [J] Yuming. Jiang, Anne Nevin and Peder J. Emstad “Implicit admission control for a differentiated services network.” In *Proceedings of the the 2nd EuroNGI Conference on Next Generation Internet Design and Engineering (NGI '06)*. Valencia, April, 2006.

[H,I,J]: The thesis author participated in discussions, performed simulations and contributed to ideas in these papers

Chapter 2

Background

This chapter provides relevant background and sets the stage for the work in this thesis.

The chapter starts with the basics by introducing the term Quality of Service (QoS) as it is used in this thesis. Section 2.2 explains traffic control which is needed to provide the required QoS and motivates the use of admission control described in Section 2.3. Both audio and video traffic can handle some delay/loss violations and higher network utilization can be reached by providing statistical QoS. In this respect *measurement based admission control* (MBAC) is a promising solution and an introduction and a review of relevant MBAC research is given in Section 2.5. Section 2.6 gives a thorough introduction to the MBAC robustness issues and contrasts the behavioral differences between MBAC and its counterpart, the *ideal* controller. Problems with the current approach of evaluating MBAC performance are then addressed. The final section reviews previous work on measurement error.

2.1 Quality of Service

Quality is a very general concept and Quality of Service (QoS) has been given several interpretations. How it is defined depends on the organization behind the definition and whether QoS is seen from a technical perspective, a user perception or a business perspective [1]. The International Telecommunication Union (ITU) defines QoS as: "*the collective effect of service performance which determines the degree of satisfaction of the service*". The European Telecommunications Standard Institute (ETSI) defines QoS in a very similar way [2].

Multimedia applications add stringent requirements to how the information is transferred over the Internet. In this thesis the applications in mind are human to human interactive applications such as voice over IP (VoIP), gaming, distance learning and video-conferencing. These *real-time* applications require that the interaction between the people involved appears as if they were in the same room. Too much delay in the transfer of information causes difficulties in the reconstruction of the original signal, and annoying talk overlap.

2.1. Quality of Service

The quality experienced by the end user is thus tightly linked to the *network performance* and the transfer of packets through the network from node to node, via network links, see Fig. 2.1. The QoS related to the packet delivery service is defined in Internet Engineering Task Force (IETF) in [3] as "the nature of the packet delivery service provided, as described by parameters such as achieved bandwidth, packet delay, and packet loss rates". The delay, packet loss, packet loss pattern, throughput and delay variations are *network QoS metrics* [4], that characterize network performance, relevant to real-time applications. Network QoS metrics do not necessarily map directly to an application level QoS metrics or metrics determining how quality of service is experienced by the end user. Quality of experience has become a hot topic (i.e. the ITU-T Recommendation G.1000 [5]).

It is the sensitivity of real time application to delay and varying packet delay or *jitter* that challenge network performance. Consider a video conversation between two parties, a user at terminal A and a user at terminal B in Fig. 2.1. At the sender side, information generated by the application is encoded and packetized and packets are sent through the network to the receiver. When packets are sent through the network, their order will normally not be altered. However, the packets may vary in the delay mainly because of varying waiting times in network nodes (it may also have happened that packets are routed differently).

When packets from various applications arrive to a router in a network node the packets are routed to the correct output queue based on their destination. Packets are then transmitted one at a time on the same network link. This sharing, termed *multiplexing*, can be modeled by a *system* consisting of a *buffer* of finite size where packets are waiting for transmission and a *server* representing the link capacity. Fig. 2.2 shows n applications sharing a network link. The time it takes to serve a packet is the transmission time or serialization delay, where $\text{transmission time} = (\text{packet size})/(\text{link capacity})$. The time a packet spends in this system is governed by the transmission time and the time waiting in queue.

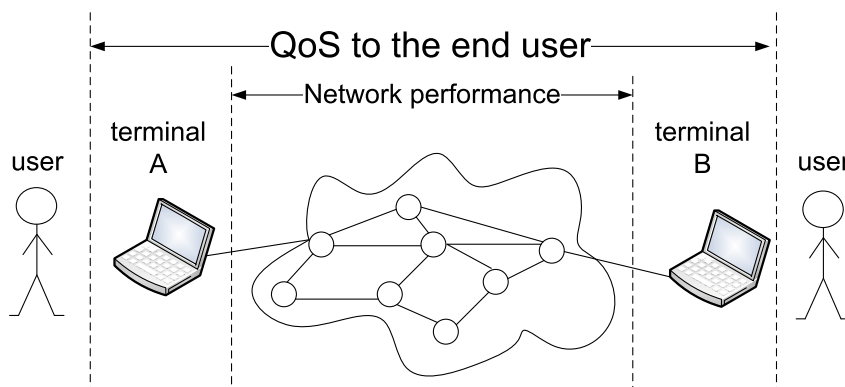


Figure 2.1: The level of QoS provided to end users depends on network performance and the terminal equipment

The total delay through the network consists of four components [6]:

- (1) *Propagation delay*: depends on physical media. For example will optical fiber result in a delay of 5 ms per 1000 kilometers.
- (2) *Switching delay*: is the time it takes from a packet is received on an incoming router interface to the packet is put in the queue of the output scheduler of the router. In high performance routers this time can be considered negligible [6].
- (3) *Transmission time delay*: or serialization, is the time it takes to transmit a packet or clock a packet onto a link.
- (4) *Queueing delay*: Is the time a packet waits in queue before it is clocked onto the outbound link.

Jitter is due to packets experiencing varying queuing delays. When a link is not shared by any other applications, the only delay will be the transmission delay. However when a burst of packets arrive simultaneously the delay will increase just slightly because only one packet can be served at a time. The delay will increase even more dramatically at times when the aggregate rate exceeds the link capacity. Packets are queued and the delay is further increased. In the case when the buffer becomes full, packets must be dropped implying infinite delay. The *jitter* is not detectable as long as the delay variation is below 40ms but as the jitter increases, the sound and picture quality drops drastically [4]. Jitter can be removed by a *playout buffer* at the receiving side, where incoming packets are buffered and then read out at a nominal rate [4]. The playout buffer adds to the total delay and it must be sized to keep the overall delay as low as possible. A proper network design must apportion the total delay requirement of an application to the various components contributing to the total delay [6], both delays caused by the network and the additional delay introduced in the end terminals. For a voice conversation, the one-way delay should be kept below 150 ms in order to be considered as acceptable [7].

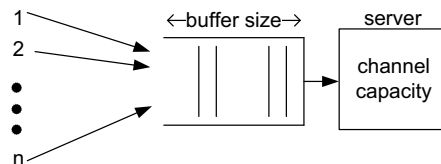


Figure 2.2: Multiplexing of internet packets at the output of a router

2.2. Multiple Time-scale Traffic control

The end-to-end transfer of packets can be characterized by a loss probability when the delay requirement is given, since packets that do not arrive on time to the playout buffer, are considered lost. Packets loss are due to:

- (1) Loss of packets due to erroneous packets
- (2) Loss of packets due to packets dropped in routers during congestion
- (3) Loss of packets due to late arrival

There is a tradeoff between the jitter buffer size and packet loss. Tolerance to packet loss largely depends on the compensation technique that can be used, the packet size, the loss pattern and price paid for the service [4]. Packet repair techniques can usually conceal losses that are scattered but repair becomes difficult when packets are lost in bursts.

2.2 Multiple Time-scale Traffic control

For real-time multimedia quality to be acceptable, it is imperative that network delay is somewhat predictable. Inadequate capacity causes both packet delay and packet loss thus an apparent solution for providing QoS is to *over-provisioning* and ensure that there is enough network capacity for congestion never to occur. There is no differentiations between users, and the architecture remains simple. Very inexpensive optical fibers make this method of *"throwing capacity at the problem"* not only the simplest but also the cheapest solution. The problem with this method, is that the decision of what is adequate capacity in the future is not trivial and unforeseen incidents can and will happen. Small bursts in demand at peak hours may not affect application only concerned with throughput but will lessen the quality of real-time multimedia applications. A discussion regarding the viability of over-provisioning is out of scope but just like in the traditional telephone system over-provisioning is needed to some extent. Nevertheless efficient use of network capacity becomes an important issue when bandwidth is scarce such as in mobile and wireless networks.

QoS mechanism are mechanisms that can be added to control the use of network capacity and protect applications that need protection against delays and loss. Within IETF there are two standardized QoS architectures, Integrated Service Architecture (IntServ) [8] and the Differentiated Services (DiffServ) [9]. Neither of these are currently deployed as a whole. However, together they define several *traffic control* building blocks which are necessary for achieving QoS [10]: *Admission Control, Shaping and policing, Signaling and resource management, Queuing and scheduling, Congestion control and queue management, and QoS routing*. In addition important building blocks (not specified within the IETF QoS architectures) are *QoS policy management* and *QoS pricing* [10]:

Internet is by nature packet-switched and handles packets on a *hop-by-hop* bases. Real-time applications are delay sensitive and demand a *circuit-switched* equivalent. For traffic control it becomes natural to separate control into two separate time-scales: *packet time-scale* which allocates resources to individual

packets; and *flow level time-scale* which allocates resources at a flow level. (In addition it is common to consider the *management time-scale* which allocates resources to aggregates of flows) [11], [12], [13].

2.2.1 Packet Level Control

Packet control is the most 'natural' level of control in the packet-switched Internet. Research related to packet level traffic control thus includes a great body of research. The primary idea is to keep the network as simple as possible and let the applications have intelligence to adapt to changes in the network. This work includes congestion control to prevent overload (by the transport control protocol (TCP)), per-packet payment mechanisms, scheduling mechanisms and marking disciplines [11]. In DiffServ a packet with a high QoS requirement will be given a higher priority at the router than a packet belonging to a class with a lower QoS requirement. This method can be very effective for preserving QoS of high priority traffic in times of congestion but can only provide relative QoS. Most of the time, when the network is working under normal traffic loads, the QoS provided to high priority traffic will be the same as low priority traffic.

Packets are dropped at routers to indicate congestion and adaptable applications will then reduce their sending rates. With only packet level control, the network must rely on all applications being adaptable and well-behaving. Non-adaptable applications that do not respond when the network becomes congested, threaten the stability of the network which will also affect the most adaptable sources. With severe congestion, even the highest priority class will suffer unacceptable service. A network link may be fully utilized, however service provided can be considered wasted if the applications are not able to perform their tasks.

2.2.2 Flow Level Control

The flow level time-scale is the equivalent of call level time-scale in the circuit based network, Asynchronous Transfer Mode (ATM) [14]. The task of flow control is basically to emulate the traffic control carried out in traditional circuit switched networks. There is no absolute definition of what exactly is an Internet flow but for practical purposes, the packets belonging to a given flow have the same identifier and occur with a maximum separation of a few seconds. One can thus think of a flow as corresponding to the transfer of data from an application. A more formal flow definition is given by [3]: *"a set of packets traversing a network element all of which are covered by the same request for control of quality of service. At a given network element a flow may consist of the packets from a single application session, or it may be an aggregation comprising the combined data traffic from a number of application sessions"*

To illustrate the need of flow level control, consider this trivial example:

- A number of subscribers using real-time video share a network link. When a subscriber uses the Internet the video application generates packets representing one flow. In this particular example, it is enough bandwidth to

2.3. Admission Control

handle exactly n_{max} flows while still ensuring QoS for all flows concurrently on the link. This n_{max} is very large, but not greater than the number of subscribers, so there is a chance (though should be small) that there are n_{max} flows already using the link. In this case, allowing one more video application into the network will result in an unacceptable QoS for all flows. There is absolutely no gain in allowing the flow into the network. To preserve QoS guarantees for the ongoing flows, this flow should be blocked. This is the role of *admission control*.

2.3 Admission Control

Flow level *admission control* is needed to preserve QoS to on-going applications and ensure network efficiency in times of overload.

When a user wants to transmit information, resources are first requested (i.e. the needed capacity). If the resources are available, the admission controller grants admission and the application can transmit data. Otherwise it is denied service. A new flow should only be accepted if the admission controller can say yes to the following basic *admission criteria*:

- (A.1) Are there sufficient resources to meet the QoS requirement of the arriving flow?
- (A.2) If the flow is accepted will the QoS of the already accepted flows still be met?

If the answer to either A.1 or A.2 is no, the flow is blocked.

When *flow level admission control* is introduced an additional QoS metric is added; the *blocking probability*. This is a flow *flow level performance measure* used for network dimensioning purposes. For dimensioning in the classical telephone network, a blocking probability below 1 % is considered acceptable [15]. It is reasonable to assume a similar or less strict requirement in the Internet. The end users should feel that they have unrestricted access to the network resources.

2.3.1 Implicit Admission Control and Flow-aware Networking

Most existing admission control approaches require the use of a signalling protocol to convey traffic information and QoS requirement of a flow to routers along its path, such as the resource reservation protocol (RSVP) in IntServ. The bandwidth broker approach [16], [17], [18] to be used with DiffServ, also requires a form of signalling protocol. This signalling requirement has imposed significant constraints on the network and consequently limited its implementation [19] [20].

Flow aware networking, (FAN) was introduced by Roberts et al [12], [13]. This is a more lightweight mechanism for providing QoS at a flow level and solves some of the technical and economical shortcomings of IntServ and DiffServ architectures.

User-defined flows are identified on the fly such that traffic control can be executed at flow level. Admission control is local to a particular network link,

where local traffic and service information can be easily obtained. In flow aware networking, the term *implicit admission* control is used since no explicit signalling is required. [21]. A flow is said to have ended or left when no packets with the same header field values are observed for a pre-defined time-out period. Note that the term implicit admission control is also used as a mechanism for TCP flows in a non-related work by Mortier et. al [22], [11].

In [23], we have proposed the use of an admission control framework called *implicit admission control for a Diffserv network* (iAC), which is easy to implement and compatible with DiffServ. Traffic information and the service requirement of a flow are carried by each packet of the flow using the DiffServ field in the IP header. We refer to [12], [13] and [23] for more information on flow-aware networking.

2.4 Admission Control Algorithms for Statistical QoS

Admission control is straight forward if capacity is allocated based on worst case behavior of flows; all flows continuously sending packets at peak rates. In this case, the admission controller accepts a flow with peak rate r_i , as long as the sum of the peak rates of the aggregate of flows is less than the available capacity c . With n flows currently in the system, the admission control test becomes simply:

$$r_i + \sum_{j=1}^n r_j \leq c \quad (2.1)$$

In reality applications send traffic at a very variable rate. The combined arrival rate from these flows, the *aggregate rate*, $R(t)$, may be considerably lower than the sum of the peak rates. Fig. 2.3 illustrates the resulting aggregate rate $R(t)$ as it varies with time, when n applications are multiplexed onto a link. The straight line marked as 'peak', is the capacity that would be required if all sources were sending at peak rate. If the admission controller takes advantage of this *multiplexing gain*, more flows can be accepted into the network.

Most real-time voice and video applications are *relaxed* in the sense that they can tolerate some packet loss. Using this fact, network utilization can be greatly improved by exploiting statistical multiplexing and permitting the instantaneous aggregate rate to exceed the link capacity with a small probability. The service provided is a *statistical or stochastic service guarantee* which may be expressed as [24]:

$$P(\text{packet loss worse than required}) \leq \epsilon \quad (2.2)$$

In contrast to the more *relaxed* voice and video applications, real-time music is in general more sensitive to both loss and delay of information. A critical example is interactive music playing for which one-way delay should be smaller than 40ms [25] in order to get acceptable quality. In this case the network must provide *deterministic service guarantees* (i.e. $\epsilon = 1$).

2.4. Admission Control Algorithms for Statistical QoS

For *stochastic service guarantees*, admission control becomes not only the task of fulfilling the admission requirements A.1 and A.2. In addition the admission controller must strive to improve network utilization as much as possible:

- *The objective of admission control is to provide the required QoS for each accepted flow, while at the same time allowing efficient use of the network.*

Consider the simple example from Section 2.2.2 where the link capacity is c , and the flows are *homogeneous* and *independent* and transmits packets with mean rate, ξ and peak rate r . The stochastic service requirement of these flows allows the instantaneous aggregate rate to exceed the capacity with a probability ϵ . This *loss probability requirement* can be written:

$$P(R(t) > c) \leq \epsilon \quad (2.3)$$

The task is to find the maximum number n_{max} of flows that can be multiplexed while meeting the *loss probability requirement* (2.3). Allocating resources based on the peak rates, meets the loss probability requirement but the number of flows that can be accepted is smaller than n_{max} . In contrast, the best multiplexing gain will be reached if each source is allocated a bandwidth of ξ , but then the loss probability requirement is violated. The loss probability requirement can be met if each source is allocated some bandwidth somewhere between the peak rate and mean rate. This amount of capacity required by a flow, is termed the *equivalent bandwidth*, EB of a flow, where $\xi < EB < r$, is the minimum capacity which ensures the loss probability requirement is met, see Fig. 2.4.

In this simple example where flows are *homogeneous* all flows will have the same effective bandwidth and flows are admitted as long as the number of flows in the system is below n_{max} . In a more realistic setting a flow i will have *equivalent bandwidth*, EB_i , where $\xi_i < EB_i < r_i$. The equivalent bandwidth varies with the stochastic properties of a flow, and the loss probability requirement. When the

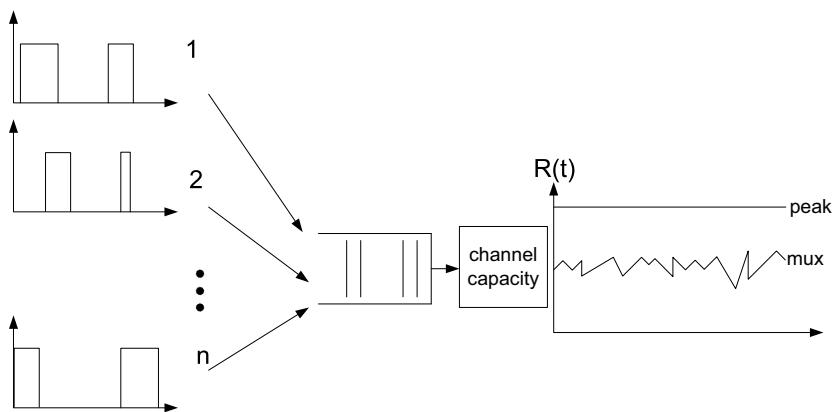


Figure 2.3: Illustration of multiplexing gain for n sources

effective bandwidth of the flow is known, the admission control test becomes the simple *additive effective bandwidth* test. A new flow i with effective bandwidth EB_i will be accepted if [26], [27]:

$$EB_i + \sum_{j=1}^n EB_j \leq c \quad (2.4)$$

A considerable amount of work in the ATM literature covers methods of determining the effective bandwidth, under the assumption of both bufferless and buffered multiplexing and an overview of this work can be found in [28] and [27]. Bufferless multiplexing, makes for more straight forward analysis but at the penalty of decreased link utilization. Adding a buffer to absorb prolonged bursts in the arrival process, increases utilization but requires a more complex analysis [26], [27].

The *additive effective bandwidth* approach (2.4), has some shortcomings. For one, it cannot exploit economics of scale as the equivalent bandwidth of a flow is independent of properties of other traffic flows, the number of multiplexed flows and the link capacity. In addition, the *additive effective bandwidth* approach does not consider long range dependent traffic.

Refinements to the effective bandwidth approach using large deviation theory address the limitations of the *additive effective bandwidth* approach [26]. The *Maximum variance approach* (MVA), considers the aggregate traffic and is based on the assumption that the instantaneous aggregate rate is normally distributed. This approach has been given considerable attention as the estimated loss probability proves to be quite accurate [26].

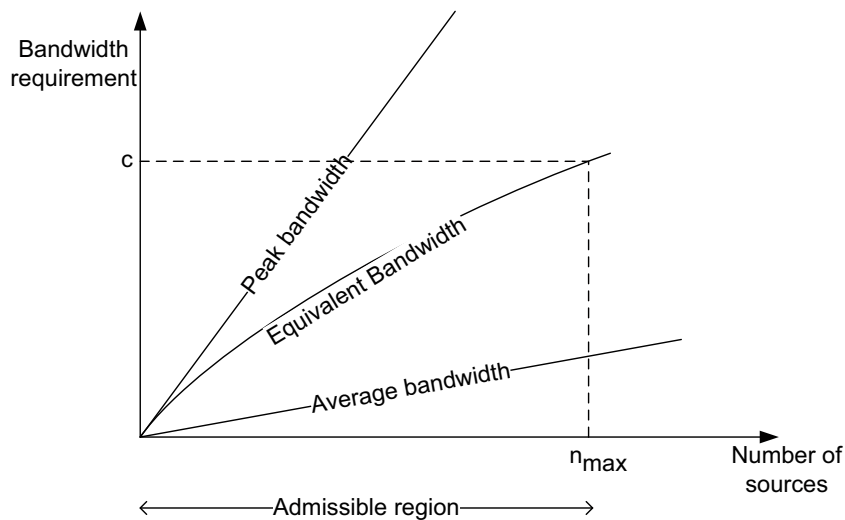


Figure 2.4: The admissible region

2.4. Admission Control Algorithms for Statistical QoS

2.4.1 Traffic Characterization

Traffic characterizations aims at describing a sources rate process or the aggregate rate process. *Stochastic traffic models* are models that approximate the statistical behavior of network traffic as accurately as possible. Some examples of stochastic traffic models are Markov-modulated Poisson processes, regression models, long-range dependent models (LRD), and models based on the normal distribution [29]. Using these models, the traffic demand can be characterized by a limited set of parameters. The assumption that the aggregate rate, $R(t)$, follows a normal distribution is generally very realistic. With this assumption, the loss probability can be very accurately calculated, if the variance and the mean of the aggregate traffic can be found.

Bounding traffic models are traffic constraint functions that are used to describe *bounds* on the rate transmitted from a flow or aggregate of flows. For each bounding traffic model, the exact traffic pattern for a flow is unknown. The only requirement is that the amount of traffic is bounded in a specific way. The token bucket model [30], provides deterministic bounds on the traffic. With such bounds on the traffic, deterministic service guarantees in a network can be predicted. Statistically bounded traffic models can more tightly characterize traffic and thus greatly increase network utilization. A good overview and introduction to stochastic bounding functions can be found in [31]. The flow *traffic descriptors* (such as the traffic specification (TSpec) [32] used in IntServ), is a set of traffic parameters describing a bounding traffic model of a flow. For example, a token bucket is described by a token rate, bucket depth and peak rate. The more tightly a flow rate can be characterized by these traffic descriptors (e.g. adding multiple token buckets in sequence), the more accurately is it possible to predict the resulting delay and loss probability.

2.4.2 Parameter Based Admission Control

Parameter-based admission control algorithms are algorithms that use traffic descriptors of each flows to predict the resulting loss probability.

In reality, tight characterization of flows is not possible as applications are not able to know *a priori* how the traffic rate from a given application will vary with time [33] [34], [35]. Traffic such as video traffic, may vary considerably during the connection time. Traffic may be distorted when mixed with other traffic such that the traffic descriptor no longer resembles the traffic. Even if the source characterization was exactly known, policing the flow is also a major issue. Policing of flows becomes more difficult as the statistical multiplexing gain potential increases [36].

The solution is that the parameter based admission controller must rely on worst-case parameters such as peak rate of flows. As such, parameter-based admission control is considered useful only for providing *deterministic service guarantees*.

2.5 Measurement Based Admission Control

Measurement-based admission control (MBAC) has for a long time been recognized as a promising solution for providing statistical QoS. ATM has generated a large body of MBAC research e.g. [37], [33], [38] and [27], and MBAC still continues to be on the research agenda for the Internet, e.g. [39], [40], [21].

The idea behind MBAC is that instead of relying on accurate source characterizations, MBAC uses measurements of the aggregate rate to characterize the flows and make admission decisions. Only a coarse traffic descriptor such as the peak rate of the requesting flow is required. (The ATM literature also includes several MBAC algorithms which measure the characteristics of each flow instead of the aggregate. However, these algorithms are considered non-scalable and will not be further discussed.)

The advantages of using MBAC over parameter based admission control are [40], [41]:

- The traffic descriptor of the flow can be trivially simple such as the peak rate.
- A conservative traffic descriptor will not result in over-allocation of resources throughout the flow duration.
- It is easier to estimate the aggregate behavior than the behavior of existing flows.

Grossglauser et al sets up a list of three requirements that must be fulfilled in order for an MBAC approach to be successful in practice:

- (1) **Robustness:** MBAC must be robust to measurement error, flow heterogeneity, self-similarity, and heavy offered loads e.g. due to "flash crowds".
- (2) **Resource Utilization:** After fulfilling the QoS requirement of all flows, the secondary goal of MBAC is to maximized utilization.

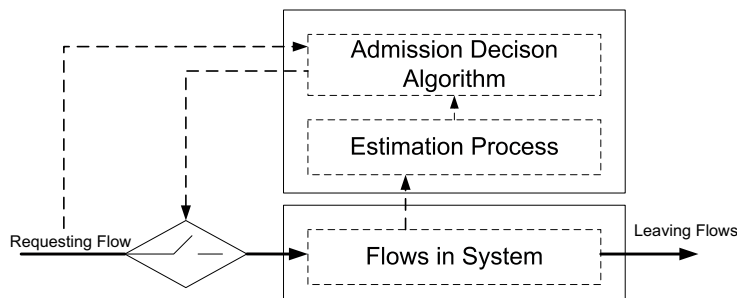


Figure 2.5: MBAC structure model

2.5. Measurement Based Admission Control

- (3) **Implementation:** The MBAC implementation must be cost effective. The computational complexity must be scalable in the flow arrival rate and in the link capacity. Adding MBAC to the network should be as non-intrusive as possible.

For simplicity, the assumption in this thesis is that the available capacity is fixed. Figure 2.5 depicts the structure of MBAC. It shows that an MBAC algorithm for a network system consists of (1) an *admission decision algorithm* and (2) an *estimation process*. The estimation process uses measurements to estimate parameters that describe the aggregate rate process of accepted flows. These estimated parameters are used as input to the admission decision algorithm. In addition, the admission decision algorithm, also relies on some input from the requesting flow, which typically includes its quality of service requirement and its traffic description (peak rate).

To a specific MBAC implementation, the system in Figure 2.5 could represent a single node, a domain, or an end-to-end MBAC. Depending on how MBAC is implemented it can be classified as [19]:

- *Node-by-node MBAC:* In this case, MBAC is implemented in each node [41].
- *Ingress MBAC:* MBAC is implemented at the ingress of the network [42] [43].
- *Egress MBAC:* MBAC is implemented at the egress of the network [44].
- *Centralized MBAC:* MBAC is implemented at a central controller such as bandwidth broker in DiffServ [9].
- *Endpoint MBAC:* If the system is the whole end-to-end network, the three elements for MBAC may be implemented at end-systems/applications [45] [42].

Though the work in this thesis is not restricted to any service architecture, the assumption of a flow aware network such as defined in [23] motivates the use of *node-by-node* admission control. The admission control implementation can be modular and requires minimal intrusion on the current infrastructure. We thus only focus on single node MBAC algorithms.

In the remainder the following assumptions are made:

- *Each flow is a stationary process, independent of other flows.*
- *All flows have the same loss probability requirement.*

2.5.1 Admission Decision Algorithms

For the admission decision shown in Figure 2.5, many algorithms have been proposed and investigated using various theoretical grounds. Based on the assumptions and analyzes that they are built upon, these algorithms can be broadly put into the following categories: *Effective Bandwidth Approximation*, *Loss Probability by Gaussian Approximation*, and *Measured Sum*.

Effective Bandwidth Approximation

Many MBAC algorithms have been developed based on the concept of effective bandwidth as discussed in Section 2.4. In these algorithms, the purpose is to calculate the effective bandwidth of each flow and/or the existing aggregate flow. The calculations are based on either a Gaussian distribution [46], [38], Hoeffding bounds [47], measured bandwidth requirement in Measure CAC [48], or other distributions.

The MBAC algorithm represented in [47] uses the measurement of the average aggregate rate \hat{R} of existing flows together with the peak rates r_i of all accepted flows to determine the equivalent bandwidth of the aggregate \hat{C}_H :

$$\hat{C}_H = \hat{R} + \sqrt{\frac{\ln(\frac{1}{u_H}) \sum_{i=1}^n r_i^2}{2}} \quad (2.5)$$

where, u_H is a tuning parameter. A new flow is admitted if the sum of the peak rate of the new flow r and the equivalent capacity \hat{C}_H is less than the capacity of the link c :

$$\hat{C}_H + r_i \leq c \quad (2.6)$$

A class of admission decision algorithms that indirectly use the effective bandwidth in the admission decision has also been investigated. Particularly, several admission decision algorithms, which are motivated by special choices of Chernoff bound, are proposed in [49]. These choices of Chernoff bound correspond to different tangents to the effective bandwidth function, which include tangent at peak, tangent at arbitrary location, tangent of slope one, and tangent at origin.

Loss Probability by Gaussian Approximation

This class of MBACs uses the assumption that the aggregate rate $R(t)$ follows a Gaussian distribution. The loss probability is then determined by calculations. If the loss probability requirement is higher than what is calculated, the requesting flow is admitted; otherwise, it is rejected. The work in [34] and [40] assumes a bufferless multiplexer and the probability of overflow can be calculated and compared to a target overflow probability.

Theory based on maximum variance approach (MVA) as used in [50] and [51], requires the measurement of the variance, average rate and dominant time-scale to determine proper bounds on the loss probability. In practice this proves difficult as in order to find the dominant time-scale and the corresponding variance, the variance must be known. A solution to this problem is considered in [50] where a combination of measurements and analysis is used to find a stopping criterion for the dominant time-scale.

A related approach for MBAC is aggregate traffic envelopes [52] [53]. In [53], aggregate rate envelopes are used in the design of the proposed MBAC approach. Its idea is to characterize the traffic of an aggregate via maximal rate envelopes. The estimated average \hat{R}_{env} and the estimated variance $\hat{v}(t)$ of the measured

2.5. Measurement Based Admission Control

rate envelopes over the past m measurement windows are computed and used by the admission decision algorithm. A flow i with peak rate r_i is admitted, if the following test is successful:

$$\hat{R}_{env} + r + \alpha \hat{v}(t) \leq c \quad (2.7)$$

where α is a tuning parameter ensuring a loss probability requirement. The setting of this parameter is discussed in [53].

In addition, there are also MBAC schemes that directly measure the loss probability using virtual queues [54], [55].

Measured Sum

The *Measured sum* proposed in [41], is possibly the simplest admission decision algorithm. Its idea is to only measure the average rate \hat{R} of existing traffic in the system. A requesting flow i is admitted if the following test succeeds:

$$\hat{R} + \nu < uc \quad (2.8)$$

where ν denotes the requested rate by the flow, c the total link bandwidth as defined earlier, and u here is a targeted utilization.

The measured sum is simple yet representative. Breslau et al [56] investigated the MBAC algorithms the *Measured Sum* [41]), *Hoeffding Bounds* [47]), *Tangent at Peak* [49]), *Measure call admission control* [48]) and *Traffic Envelope* [53] and concluded that they all share a common structure, share a common behavior and they all can be mapped to the simple *measured sum* algorithm (2.8). The algorithms will then basically differ in the way u is determined but when it is found, all algorithms can be considered being the same.

The utilization parameter depends on the traffic characteristics. If the flow rate characteristics are known, u can be calculated based on e.g. equivalent bandwidth theory. In the more realistic case where the flow rate characteristics are unknown, u can be estimated based on measurements of the aggregate rate process or observing the individual flow characteristics over some time.

Clearly in order to estimate u (as many MBACs implicitly do) an estimate of only the average load is not sufficient. More information regarding the aggregate rate is needed. When the sources are non-homogeneous which will be the case in a realistic network setting, determining the admission region, or uc becomes difficult as uc will change depending on traffic composition. The issues of statistical multiplexing and non-homogeneous flows can become quite complex and analytical analysis have been conducted in e.g. [57] and [58]. One can argue that the proper setting of u is a question of proper dimensioning. In [59], u is determined based on measurements of the variance (implicit by a virtual queue) and the average of the aggregate rate process. In this thesis a proper setting of u is out of scope.

2.5.2 The MBAC Estimation Process

The task of the MBAC estimation process is to estimate parameters based on observations of the aggregate rate process. Several estimation methods are proposed

in the MBAC literature and what to be measured depends on the parameters needed by the admission control algorithm.

Here, we only consider the *measured sum* algorithm. This algorithm needs the measurement \hat{R} , which is a measure representing an estimate of the current load of the system. To show how the measurement process may differ, we will present 4 methods of estimating this current load. The *average rate*, *time-window average*, *maximal rate envelope* and *exponential moving average*.

- \hat{R}_w is the *average rate*: If the traffic arriving in a time interval $w = t - s$ is $A(s, t)$, then the *arithmetic average rate* is simply:

$$\hat{R}_w = \frac{A(s, t)}{t - s}$$

If the average is based on t being the current time, it becomes a simple moving average.

- \hat{R}_{tw} is the *time window average*: The measurement process originally designed for the measured sum algorithm is termed the *time window* [41]. In this scheme, time is slotted into slots of size τ . After every time-slot the average rate over the slot is computed. After k slots, \hat{R}_{tw} is computed as the maximum average rate seen in a block $k\tau$. At the end of a time block, a new estimate replaces the old. When a new flow is admitted, the estimate is increased by the parameters requested by the new flow and the time window is restarted. If a newly computed average is above the currently estimated value, this estimated value is immediately raised to the new value.
- \hat{R}_{env} is the *average maximal rate envelope*: This measurement process is used by the traffic envelope approach in [53]. In this case, time is also slotted into slots of size τ and the arriving rate in terms of number of bits arriving within a slot is recorded. The current maximal rate envelope, R_k^1 , is a set of rates representing the maximal rate envelope, based on k such recordings. The definition and details of how to determine the maximal rate envelope is given in [53]. The average \hat{R}_{env} over the past m windows of length w_k is then computed and used by the admission decision algorithm:

$$\hat{R}_{env} = \sum_{n=1}^m R_k^n / m$$

- \hat{R}_{ema} is the *exponential moving average*: There is a vast literature covering different methods of predicting the aggregate rate using recursive filters. A simple filter, is the exponential moving average (i.e. [47]). This filter works by adding most *weight* to the most recent measurement \hat{R}_{new} (the average over an interval) and the average rate is determined according to

$$\hat{R}_{ema} \leftarrow (1 - weight) \cdot \hat{R}_{ema} + weight \cdot \hat{R}_{new} \quad (2.9)$$

2.6. Robustness Issues

The parameter *weight*, has a range from 0 to 1, and can be viewed as a history parameter. A value of one means no history included and as weight approaches 0, the estimate reflects very old states of the system.

A slightly more complex filter is the Kalman filter which includes measurement errors when predicting the state of the system. Most weight is then given to the measurement of least uncertainty, [38]. We also mention the Fractional AutoRegressive Integrated Moving Average (FARIMA) model [60] as a more advanced method of predicting system state. Prediction of system state can also be done experimentally using neural networks as in e.g. [61].

2.5.3 Tuning Parameters

The *tuning* parameters are the adjustable parameters used by MBAC.

As we have seen in the various admission control algorithms mentioned above, they all have a tuning parameters built into them. By varying a tuning parameter, an MBAC can be made more or less pessimistic. The measurement process in MBAC implies additional tuning parameters. For example, the measurement process, (2.9) includes two tuning parameters, the length of a measurement window w and the *weight*. Also the *time window* measurement process includes two tuning parameters. These are the length of the block and the length of the slots, where a discussion of the proper setting is given in [41]. The measurement process of the MBAC algorithm in [53] includes three tunable variables, the slot size, the measurement window and the number of measurement windows used to find the average and variance. The measurement process can be tuned to utilize the bandwidth in an efficient way under a particular traffic scenario. However, there is no answer as to how they should be re-tuned when the traffic scenario changes. There are attempts such as [62] and [63] to use varying window sizes that adapt with the traffic situation, which have shown to improve performance in some situations. The adaptive algorithm also has external tuning parameters that will depend on the traffic characteristics [40].

2.6 Robustness Issues

The most important requirement of MBAC is to ensure the requested QoS to the accepted flows in all situations. The robustness issues become a challenging task for MBAC since it relies on erroneous measurements. Measurement errors cause uncertainties in the admission decision. The consequence of this uncertainty will depend on flow rate characteristics, how often new flows arrive and how long the flows stay in the system.

2.6.1 The Certainty Equivalent Controller and the Ideal Admission Controller

In the case where the estimation process in Fig. 2.5 is completely removed and the MBAC is instead fed with the 'true' parameters of the aggregate rate of the

accepted flows, the result is an 'certainty equivalent' controller [40], [40].

- *For a given MBAC algorithm, the 'certainty equivalent' admission controller, is an admission controller which uses the same admission decision algorithm as its MBAC counterpart but does not depend on measurements.*

The certainty equivalent controller is merely an analytical concept as the true parameters are difficult to obtain in a realistic network environment, see Section 2.4. How well an admission controller performs is determined by *how correctly the admission controller can make an admission decision*. A requesting flow should be correctly accepted if the admission criteria (A.1) and (A.2) is fulfilled. Otherwise the flow should be rejected.

The certainty equivalent controller bases its decision on the behavior of stochastic processes and must rely on approximations. The decision of admitting or rejecting a flow will thus not always be correct.

The *ideal* controller can be seen as a *psychic controller*. It is a controller that can 'see' everything regarding flow characteristics. This controller has full control on the number of flows currently in progress and does not rely on approximations for determining how many flows can be accepted.

- *The ideal controller, is an admission controller that always makes a correct admission decision.*

In the case where flows are *homogeneous*, the *ideal* controller can be much simplified. The ideal controller is then the *Quota* algorithm [56]. This ideal controller accepts a flow as long as there are less than n_{max} flows in the system, where n_{max} is the maximum number of flows the system can handle. We define:

- **The saturation case** is the case where there is always a new flow waiting for admission and as soon as one accepted flow leaves, a new flow can be considered for admission immediately.

Based on the definition of the *certainty equivalent* controller, the *ideal* controller and the *saturation case*, the following can be stated:

- *For the 'ideal' controller, the maximum utilization the system can handle or the system limit, is reached in the saturation case.*
- *For a given 'certainty equivalent' controller, the maximum utilization that can be achieved may be less than the system limit. However, it should never exceed this limit.*

The *ideal* controller is merely an ideal concept and cannot be implemented in real systems. In a simulation environment when it is possible to have full control on all parameters the *ideal* becomes a realistic concept. The maximum utilization the system can handle is most easily found by increasing the number of flows until the loss probability requirement can no longer be met. The performance of

2.6. Robustness Issues

the certainty equivalent controller can be evaluated with respect to how well it can mimic the ideal controller. In [26] the performance is evaluated for different *certainty equivalent* controllers in the *saturation case* and the findings show that a number of admission controllers from the literature performs close to that of the ideal controller.

2.6.2 The MBAC Performance and the Loss-Load Curve

We start the discussion regarding MBAC performance by stating two questions that must be answered before an admission decision can be made:

- **Question1:** What is the maximum utilization that can be achieved while meeting the loss probability requirement?
- **Question2:** Is the current state of the system above or below the maximum utilization?

No matter how accurately an MBAC can answer **Question1** it will be of little use if **Question2** is not answered with a certain degree of confidence.

Given a certainty equivalent admission controller, if **Question1** can be accurately answered, then we can conclude that the models used for predicting the loss probability are accurate, thus answering **Question2** can be done with good confidence. For this reason, a performance measure such as the *loss-load* curve gives a good indication of the overall performance.

In this discussion it is first important to have a precise understanding of what utilization is:

- *Utilization is the average usage of the link or the proportion of time the link is in use. It is determined based on observing the long term average rate, which is then normalized to the link capacity.*

In [56], an extensive investigation has been conducted through simulation to compare several representative MBAC algorithms mainly focusing on providing answers to **Question1**. Two criteria are used for the investigation and comparison:

(C.1) The *loss-load* curve

(C.2) Ability to meet a QoS target

The *loss-load* curve is a curve which represents the loss probability that will result for a given load where the load is the long term average rate. The second criterion is used to evaluate how well an algorithm can meet its target service guarantee based on its associated tuning parameter.

A particular point on the *loss-load* curve is found by using the following procedure: For a given source model and loss probability, several simulation runs are performed where the MBAC is tuned differently in each run until the highest link utilization is reached.

In [56], it is found that each algorithm has nearly the same *loss-load* curve and shows similar deviation from the *ideal* controller. Figure 2.6, shows the *loss-load* curve of MBAC compared to the *ideal* controller based on the results in [56].

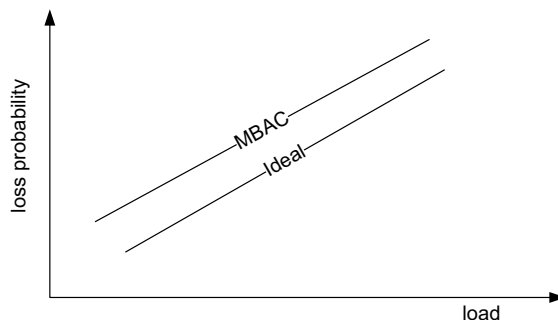


Figure 2.6: The loss-load curve to deduct the performance of MBAC

The somewhat, at first, surprising result holds across different traffic models including homogeneous exponential ON-OFF sources, Pareto ON-OFF sources, Star Wars traces and a mix of different ON-OFF sources, and heterogeneous sources. In addition, the *loss-load* curve in Fig. 2.6 shows that MBAC will regardless of the setting of the tuning parameters be unable to perform as well as an *ideal* controller [56].

Based on the results found, the following can be stated:

- *The loss-load curve should not be used as a performance metric for comparing different MBAC algorithms, since the same loss-load curve can generally be expected for each of them.*

For the second criterion, the QoS metric investigated in [56] is the loss probability. It is found in [56] that all algorithms are unable to achieve performance close to the targeted one in a consistent manner. In other words, none of the algorithms provides tuning parameters that are useful as performance targets. At best, these parameters can be seen as largely uncalibrated tuning knobs for increasing or decreasing network utilization. With this regard, the following conclusion is drawn in [56]:

- *None of the investigated MBAC algorithms in [56] can reliably match actual performance to targeted service guarantees. The ability of future algorithms to improve in this regard is an open question.*

Focusing only on **Question1** is not sufficient when discussing the performance of MBAC. The problem lies in what time-scale the performance is based on:

- *The infinite time-scale* which is the time-scale used to measure long term performance over an infinite time horizon.

2.6. Robustness Issues

- *The measurement time-scale* which is finite and used when predicting the current system state.

When performance related to **Question1** is addressed, the infinite time-scale is considered. However in order to deduct the performance of the MBAC admission decision the time-scale of interest becomes the measurement time-scale. Before addressing the performance issues, we will first look more into the details of the admission decision.

2.6.3 The MBAC Admission Decision

An *ideal* controller will give perfect answer to both **Question1** and **Question2**. If now the maximum utilization the system can handle is given, then the task of determining **Question1** is removed and we can solely focus on **Question2**. The MBAC differs significantly from the *ideal* admission controller and we identify three causes that impact each other and together results in degrading effect on MBAC performance: 1) *measurement error*, 2) *timeliness*, and 3) *offered flow load*.

Measurement Errors

When measurements are repeated, different set of observations will yield different estimates of the average aggregate rate. The measurement error is defined as the deviation of the measured value \hat{R} from the true value $E[R]$:

$$\delta = \hat{R} - E[R] \quad (2.10)$$

The statistics of the measurement error will be further discussed in Section 3.6.

Admission Timeliness

The *ideal* controller always has the current state of the system available. For MBAC it inevitably takes some time for a measurement to reflect the current state of the system. Given that the number of flows is kept constant, the state of the system can be estimated more and more accurately by observing the process over longer time. However, longer observation time means that the MBAC becomes less responsive to flow arrivals and departures. If most of the flows have left the system by the time the measurement is updated, the measurement is of little use.

The *admission timeliness* is a concept that arises in MBAC due to the measurement process and refers to the time it takes the admission controller to make an admission decision and the responsiveness to flow arrivals and departures. The *timeliness* depends on the measurement process and how the MBAC chooses to treat flow arrivals, the MBAC *strategy*.

- (1) *An MBAC strategy is an MBAC decision strategy, which determines how an MBAC chooses to handle a new arriving flow.*

To give a simple illustration of timeliness and MBAC strategy consider an MBAC which bases its decision on measurements that are updated at regular time intervals as is illustrated in Fig. 2.7. The ideal controller can accept flows f_1 and f_2 , as soon as they arrive. For MBAC, when flow f_1 arrives, the controller must thus decide if the flow should be accepted based on the most recent measurement or wait until a measurement is updated which results in an admission decision delay of T_w . For flow f_2 , there are additional options of rejecting/accepting/waiting, where the choice of *strategy* will have consequences effecting the resulting loss-load curve and MBAC performance. (Chapter 6 gives a demonstration of how different admission control strategies affect MBAC performance.)

Whether f_1 is accepted immediately based on the most recent window or must wait for a new update, the issue of timeliness still exist for MBAC in the setting of the measurement window length. More advanced measurement methods, where the most recent measurement may always be made available, will still have timeliness issues related to parameters governing the history of the measurements.

The issue of timeliness becomes more profound as the arrival rate increases. If on the other hand, flows arrive very seldom and stay in the system for very long time, the timeliness issue becomes less of a concern. In this case, the number of flows will remain approximately constant for a time sufficient for measurements to become very accurate. The resulting loss-load curve will very closely coincide with that of the *ideal* controller.

Offered flow load

For the 'psychic' *ideal* controller, the long term average aggregate rate will reach the system limit, uc in the saturation case. An MBAC may be 'tuned' to produce a long term average rate for a certain offered flow load, the Erlang load (see Section 3.7). As the offered flow load increases the probability of false acceptance increases and more and more flows will be accepted in error. As the system approaches saturation, an MBAC will eventually operate in overload where the average aggregate rate, is consistently above the maximum uc and will approach the capacity of the link, c . This is sketched in Fig. 2.8.

An MBAC that is tuned to offer high utilization under normal loads, may not be able to withstand high offered loads. On the other hand, an MBAC which is tuned to handle extreme loads may behave too pessimistically under normal

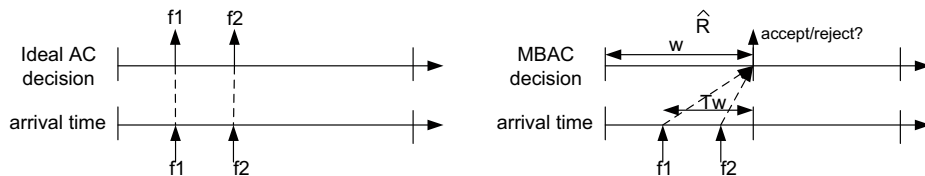


Figure 2.7: The ideal controller accepts arriving flows as they arrive whereas this particular MBAC must wait for the measurement to be updated

2.7. Evaluating the performance of the MBAC Admission Decision

loads. To what extent a particular MBAC can handle varying offered loads, heavily depends on what MBAC strategy is employed. This is discussed further in Chapter 6.

2.7 Evaluating the performance of the MBAC Admission Decision

The MBAC *loss-load* curve inevitably deviates from the *ideal*. However the *loss-load* curve cannot explain the reasons for this deviation. A network operator wishing to employ MBAC, may be more interested in knowing how to set the tuning parameters in order to reach a certain point on the loss-load curve. This question was raised as a serious concern in the work by Breslau et al [56]. To answer this question, understanding how the measurement error, timeliness and offered flow load impact the admission decision must be answered separately.

The *loss-load* curve is constructed based on observations over an *infinite time-scale*. An MBAC which alternates between being in a state of very heavy overload following a period of underload may produce the same loss-load performance as an MBAC which rarely accepts and rejects flows in error. One can ask the question: "Does an MBAC that often admits flows in error but operates at a higher overall utilization have a better performance than an MBAC which hardly ever accepts flows in error but at a cost of lower utilization?"

Performance measures based on long term averages cannot explain the performance of MBAC with respect to false acceptances and false rejections. This requires the understanding of what happens when the admission decision is being made and *the measurement time-scale*, must be considered.

Flow level performance measures are needed to identify what happens at flow level. The flow level performance measure *blocking probability*, is used for dimensioning in the classical telephone system and blocking probability has also

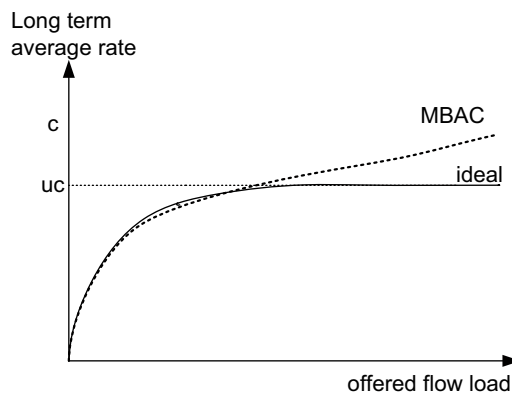


Figure 2.8: Long term average aggregate rate as the offered flow load increases, MBAC vs. ideal controller

been given some attention in admission control research related to packet switched networks. In the case where all flows are homogeneous, blocking probability is an indirect measure of utilization. When flows are non-homogeneous, the blocking probability can be used to study how a specific admission controller discriminates between flows that have different bandwidth requirements [58]. As a side, the authors in [64] notes that link utilization itself is not a good indicator of efficient usage of network resources as it ignores control overhead. They instead suggest that performance be measured by the flow level performance measures such as admittance probability, blocking probability and average call duration.

In this thesis we focus on **Question2**: *Is the current state above or below the maximum aggregate rate the system can handle.* Answering this question with a certain degree of confidence, requires knowledge of the accuracy of a single measurement which is taken over a window. The time-scale of interest is then the length of the measurement window. This measurement is used when making the admission decision. New flow level performance measures are defined that are specific to MBAC performance evaluation. As will be seen later in this thesis, these performance measures can be used to directly answer the question of how to reach a certain point on the *loss-load* curve and analytically explain why and by how much an MBAC will deviate from the performance of an *ideal* controller.

2.8 Previous work on Measurement Error

The fact that measurements are uncertain and should not be treated as the 'true' value, was already in early MBAC research flagged as a serious concern [65]. The focus has for a most part been on the accuracy related to: "*How accurately can the maximum utilization be estimated with its corresponding loss probability or How accurately can the MBAC predict a certain loss probability?*" The estimation error is then minimized by tuning the measurement windows.

Work that considers the measurement error, can be grouped into work which takes the traffic correlation structure into account and work which does not, the *memory-less approach*. With the memory less approach it is either assumed that the MBAC itself is memoryless and bases the admission on the instant aggregate rate, such as the Bayesian approach in [33] or it is assumed that the traffic itself is memory-less such as in work done by Dziong et al [38] and Y.S. Sun et al [66].

Today's Internet traffic possesses inherent correlation structures thus assuming memory less traffic behavior may only give limited insight into the measurement error. Correlation between samples is shown to additionally degrade performance [67]. In the literature we mention [27], [67], [34] and [40] as work which considers the estimation error and also takes correlation characteristics into account.

The COST-242 final report [27], represents an *Adaptive* MBAC based on instantaneous measurement of the total mean rate of the link. When the variance and mean rate of the rate process is assumed known, the measurement error is analytically stated upfront and incorporated into the MBAC algorithm. When the distribution of the number of flows is known, the method works quit well at predicting the effective bandwidth. For accurate results a long measurement

2.8. Previous work on Measurement Error

window is required. Though the analytical analysis can be carried out when flows are non-homogenous, the efficiency of the method suggests a homogenous flow environment [27].

The randomness of measurement error and its accuracy has been studied in the asymptotic regime under heavy load and where the rate from a single flow is very small compared to the link capacity. By means of large deviation theory, the work in [67] analyzes the impact of measurement error on the packet loss rate. This analysis is used as a bases for studying how correlation between samples impact the resulting packet loss rate. The work in [53] uses *conditional prediction* by considering the correlation between successive measured traffic envelopes to make more accurate estimates of the loss probability. A study of a robust MBAC which emphasizes the impacts of estimation errors, measurement memory, call level dynamics and separation of time-scales is given in [34] and [40].

Linear filters such as the Kalman filter includes measurement errors when predicting the state of the system. Most weight is then given to the measurement of least uncertainty, [38].

Using a prediction model such as the Kalman filter is useful at predicting system state, based on several measurements. In this work we focus on the accuracy of a single measurement which is taken over a window where the number of flows does not change, e.i. the aggregate rate process is assumed stationary with a known distribution.

2.8.1 Accounting for Measurement Error when making the MBAC decision

How the measurement error actually affects the MBAC decision has been given little attention [39], however, there are efforts to incorporate this error into the MBAC decision process. We distinguish between two methods: 1) adding a *Back-off strategy*, and 2) adding a *safeguard*.

The back-off strategy

The *Back-off* strategy introduced in [65] and later adopted by [33] and [66], works by turning down subsequent arrivals after one flow has already been denied admission. When a flow has been rejected admission, no flows will be admitted until a flow has left the system. In reality it may not be possible to keep track of flow departures, [20], [40]. Alternatively, a deterministic wait can be added before another flow is accepted [37]. In the latch algorithm proposed in [66], the *Back-off* strategy is also induced when packet loss violation is registered.

The motivation for using the *Back-off* strategy is to reduce the probability that the MBAC will accept a flow due to under-estimation of the aggregate rate. It is analytically shown that the *Back-off* strategy is robust when the offered flow load is high [33].

Adding a Safeguard

The most common method of dealing with measurement error is by adding a safeguard to the decision algorithm to make up for the measurement error. A safeguard or spare bandwidth [38], is a slack in bandwidth which can be added to any MBAC algorithm. Basically it works by making some of the reserved bandwidth unavailable. For example, if the reserved average rate to the flows is uc , and the safeguard is $slack$, then the reserved bandwidth is reduced by this $slack$. In cases where the QoS requirement is used as an input to the algorithm, a safeguard is implemented by using a stricter QoS requirement which again translates to a slack in bandwidth, [40].

If the measurement error is not treated analytically, the safeguard is 'hidden' in the tuning parameter u , where u is tuned to an 'optimal' setting based on simulations [56], [47], [68].

2.8.2 Call Level Dynamics and MBAC Performance

There has been limited work on seeking the understanding of the effect the flow dynamics have on measurement error. The distribution of accepted flows will depend on the offered flow load and the measurement error. Gibbens et al [33] use a decision-theoretic approach for call admission control to explicitly incorporate call-level dynamics into the model. A call is accepted if the instantaneous rate measurements are above a threshold. However, they do not specify how these measurements should be taken.

A critical time-scale dependent on flow lifetimes was defined in [34], [40]. However, in this work, the arrival pattern is excluded from the model and instead the focus is on an artificial saturation region with flows always available for admission and the blocking probability driven to infinity. In theory one could say that an infinite arrival rate of new flows gives the worst case in the sense that there always is a flow available for admission. We believe that it is of equal importance to understand how the arrival process affects performance.

Chapter 3

Measurement Error and MBAC Performance, Concepts and Definitions

MBAC uses measurements to capture the behavior of existing flows and uses this information together with some coarse knowledge of a new flow, when making an admission decision for the requesting flow.

A new flow should only be accepted if the admission controller can say yes to the following basic *admission criteria* (repeated from Chapter 2):

- (A.1) Are there sufficient resources to meet the QoS requirement of the arriving flow?
- (A.2) If the flow is accepted, will the QoS of the already accepted flows still be met?

Measurements are unavoidably inaccurate. This imperfection creates uncertainties which affect the MBAC decision process. The degree of uncertainty depends on flow characteristics, the length of the observation window and the flow dynamics. A flow that is accepted when it should have been rejected is denoted a *false acceptance*. A flow that is rejected when it should have been accepted is denoted a *false rejection*. Clearly, by answering yes to the above questions when the answer should have been no will put all the flows at risk of QoS violations. This wastes network resources and provides little utility to the end users. The measurement error may be highly significant, and can have a huge impact on the overflow probability [34] and there is great doubt of the practical application of an admission control scheme that does not consider this error [37]. An in-depth understanding of the measurements themselves and how they are affected by the underlying traffic is vital for the design of a robust MBAC.

In reality, the true value of the measurement is always unknown and in order to describe the measurement error and the consequence it will have on MBAC performance, this must be investigated theoretically.

3.1. The MBAC Behavior

In this chapter, the underlying assumptions regarding the analytical framework are given and the flow level performance measures are defined. When performing the analytical analysis, the underlying parameters are disclosed and the measurement error can be characterized.

3.1 The MBAC Behavior

Consider a network link of capacity c and Internet flows with real-time requirements competing for this resource.

The flows have a QoS requirement which can only be guaranteed as long as the average aggregate rate is at or below the *limit* uc , where u , $0 < u < 1$, is a tuning parameter. An optimal value for u depends on the flow characteristics. In this work, u is assumed a given constant and a discussion around its optimal settings is out of scope.

An MBAC is put in place to control access to this link and prevent the average aggregate rate from exceeding its upper limit, uc . The behavior of the MBAC is as follows: The MBAC measures the average aggregate rate \hat{R} based on observations of the aggregate rate $R(t)$ over a measurement window of size w . This measurement replaces the measurement taken in the previous window. A new arriving flow carries with it a bandwidth requirement ξ which is fed to the MBAC. When a new flow arrives, it will be accepted if:

$$\hat{R} + \xi \leq uc \quad (3.1)$$

Otherwise the flow is lost. Additional flows arriving within the measurement window are also denied admission.

This MBAC algorithm is simple and also representative because different MBAC schemes can be mapped to this algorithm [56]. Notice however, that the MBAC algorithm uses the mean rate of the arriving flow as opposed to the peak rate more commonly used in the literature. Assuming peak rate, r , of the arriving flow, adds a pessimism to the MBAC which can be translated to subtracting uc with a slack bandwidth of size $r - \xi$. Fig. 3.1 shows an example where two flows,

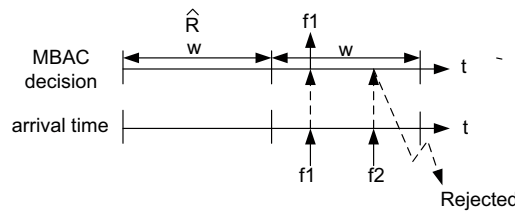


Figure 3.1: Relationship between measurement window updates and flow arrival times. Here flow $f1$ is accepted if $\hat{R} + \xi \leq uc$ and flow $f2$ will be lost

$f1$ and $f2$ arrive within the same interval. According to the MBAC behavior, flow $f2$ will be rejected. Flow $f1$ will be accepted or rejected according to the

Chapter 3. Measurement Error and MBAC Performance, Concepts and Definitions

MBAC algorithm based on the most recent measurement update. In order to do the analytical analysis, assumptions regarding the underlying traffic are needed. In the following sections, we will give the analytical assumptions and background used in the remainder of this thesis.

3.2 Traffic Modeling

For the analytical analysis it is useful to model the traffic at different time-scales and we distinguish between the rate level, where the measurements are done and the flow-level where the admission decision is made, see Fig. 3.2. At the flow level, the number of flows $N(t)$, the *flow level dynamics*, varies with time due to flow arrival and departures. The aggregate rate $R(t)$, the *rate level dynamics*, varies with time due to both the varying number of flows and also due to the variable rate process of the individual flows.

When aggregating a large number of flows, the aggregate rate can be assumed normally distributed. It is not possible to give a hard number on the exact number that is required as this depends on the statistics of the individual flows. According to [69] the normal assumption is safe, even when the number of flows is only a 'few tens'.

The flow level dynamics are governed by the *Distribution of flow lifetimes* and the flow *Arrival Processes* and in addition, the MBAC admission algorithm. The MBAC admission decision makes its decision based on measurements done at the rate level which again affects the flow level dynamics. Note that this is

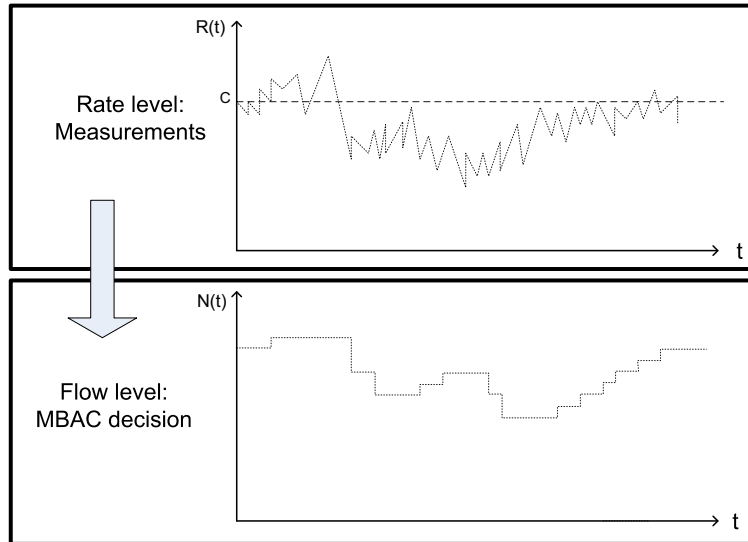


Figure 3.2: The aggregate rate process $R(t)$ and number of flows $N(t)$ vs time

3.2. Traffic Modeling

fundamentally different from the traditional telephone system, where the rate level is not a decision parameter.

When describing an individual flow we distinguish between *flow level characteristics* and *rate level characteristics*.

3.2.1 Flow Level Characteristics

The distribution of flow lifetime and arrival process characterizes a flow at flow level.

Distribution of Flow Lifetime

A flow, when accepted by the MBAC, stays in the system for a lifetime T_L . In this thesis we will assume that flow lifetimes are negatively exponentially distributed with mean, $1/\mu$. However, the analytical framework we define holds for arbitrary distributions of flow lifetimes as long as the mean lifetime is known.

Arrival Process

Flows arrive following a Poisson process with parameter λ . The Poisson assumption is reasonable in the case for Internet flows, at least for the flows belonging to the category of streaming flows [35], [70]. In fact [71] points out that the assumption of Poisson arrival process of flows is one of the few assumptions that hold in a very wide range of environments. The Poisson assumption will thus be made throughout this thesis

3.2.2 Rate Level Characteristics

All flows are taken to be independent and at the rate level, it is assumed that the flow rate process $K(t)$ is a stationary rate process and can be described by its:

- **mean**, ξ
- **peak rate**, r
- **variance**, σ^2
- **burstiness**, B : Given the peak rate and mean rate of the rate process, we define in accordance with the ITU-T definition, the flow burstiness as the ratio of the peak to mean rate [72]:

$$B = \frac{r}{\xi} \tag{3.2}$$

A source is said to be *bursty* when $B \geq 2$.

Chapter 3. Measurement Error and MBAC Performance, Concepts and Definitions

- **auto-covariance function, $\rho(\tau)$** : The auto-covariance function is given by:

$$\begin{aligned} \rho(\tau) &= \text{cov}(K(t), K(t + \tau)) \\ &= E\{(K(t) - E[K(t)])(K(t + \tau) - E[K(t + \tau)])\} \\ &= E[K(t)K(t + \tau)] - \xi^2 \end{aligned} \quad (3.3)$$

- **auto-correlation function, $\Psi(\tau)$** : The auto-correlation function is given by:

$$\Psi(t) = \frac{\text{cov}(K(t), K(t + \tau))}{\sigma^2} \quad (3.4)$$

3.3 Traffic Classes and System State

To distinguish between different types of flows, the concept of traffic classes is often used in the literature, see for example [65] and [73]. Specifically, let there be k classes of flows, where the members of a given class are those with the same values of their traffic parameters, see Fig. 3.3. In our definition flows belong to

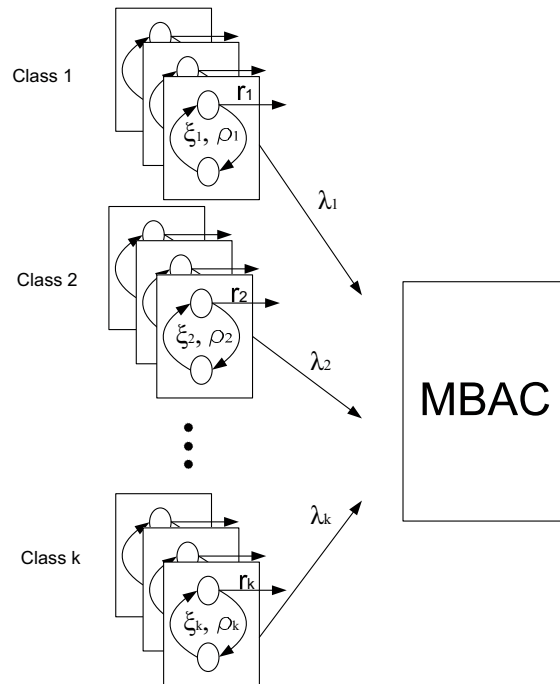


Figure 3.3: A system with k classes of flows

3.4. Measurements

the same particular class i , if they have the same *flow level characteristics* and *rate level characteristics*. In a system where all flows belong to the same class, the flows are said to be *homogeneous*.

The system state is described by the current number of flows. If the system consists of homogenous flows, the system state is the number of flows $N = n$. In a system with several classes of flows, the state vector $\mathbf{N} = \mathbf{n} = (n_1, n_2, \dots, n_k)$ describes the current state of the system, the number of accepted flows from each class.

3.4 Measurements

We define in accordance with the field of metrology, *measurement* as the process of experimentally determining the value of a *measurable quantity* [74].

The MBAC observes the aggregate rate process $R(t)$ which is a stochastic process. An *observation* of the stochastic rate process is a random variable and a set of observations constitute a *sample*. In statistics, an *estimator* is a function, which takes as input a set of observations and produces an estimate of the parameter of interest.

For the MBAC under consideration, the *unknown* parameter of interest is the mean aggregate rate $E[R]$. It is assumed that the mean exists. The instantaneous aggregate rate and the average aggregate rate over an interval are two unbiased estimators of the mean rate. Both are measurable quantities, the latter has less variance. The average aggregate rate can be found based on *discrete* observations or *continuous* observation.

3.4.1 Observation Method 1: Equidistant Sampling

Consider the individual flow rate process $K(t)$ in Fig. 3.4, which is covariance stationary with covariance $\rho(\tau)$ and mean ξ . The process is observed every time

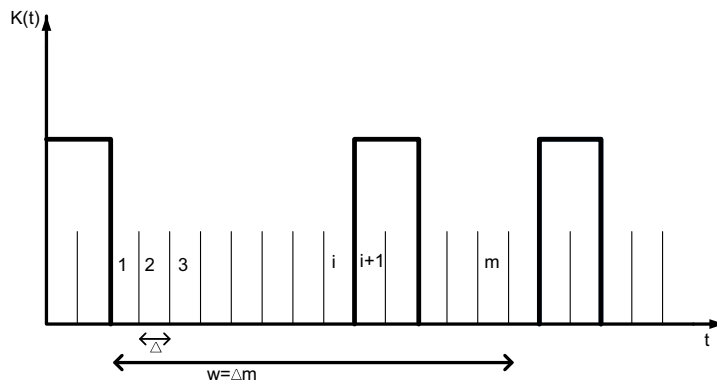


Figure 3.4: Rate $K(t)$ vs time

Chapter 3. Measurement Error and MBAC Performance, Concepts and Definitions

slot Δ , where X_i is the observation at the end of time slot i . A measurement window w , consists of m observations of the process, $w = m\Delta$, see Fig. 3.4.

With equidistant sampling, an instant sample of the rate $K(t)$ is taken at every $t = \Delta i$. X_i is the sampled rate at the end of time slot i given by $X_i = K(t_i)$. The measured sample $X = X_1, X_2, \dots, X_m$ will be identically distributed but correlated observations, where the X_i s have a sample mean, \hat{K} given by

$$\hat{K} = \frac{1}{m} \sum_{i=1}^m X_i \quad (3.5)$$

The sample mean is an *unbiased* estimator since $E(\hat{K}) = \xi$.

A general expression for the variance of \hat{K} , $Var(\hat{K})$, is given by [75]:

$$\begin{aligned} Var(\hat{K}) &= E[(\hat{K} - \xi)^2] \\ &= \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m E[(X_i - \xi)(X_j - \xi)] \end{aligned} \quad (3.6)$$

and with a covariance stationary process:

$$Var(\hat{K}) = \frac{1}{m^2} \sum_{h=1-m}^{m-1} (m - |h|) \rho(h) \quad (3.7)$$

3.4.2 Observation Method 2: Continuous Observation

The best estimate of the mean rate is found by continuous observation over the window. Analytically this is done by letting the sampling rate go towards infinity.

Let now $\Delta \rightarrow 0$ and $m \rightarrow \infty$ keeping the product $m\Delta$ constant such that $t_i = i\Delta \Rightarrow t$ then:

$$\hat{K} = \frac{1}{m} \sum_{i=1}^m X_i \Rightarrow \lim_{\Delta \rightarrow 0 | w = m\Delta} \frac{1}{m} \sum_{i=1}^m X_i = \frac{1}{w} \int_0^w K(t) dt \quad (3.8)$$

Using limit considerations known from the literature, the variance of the time average, $\zeta^2(w)$ can be found:

$$\begin{aligned} \zeta^2(w) &= \lim_{\Delta \rightarrow 0 | w = m\Delta} Var(\hat{K}) \\ &= \lim_{\Delta \rightarrow 0 | w = m\Delta} \left(\frac{\Delta}{w}\right)^2 \sum_{i=1-m}^{m-1} (m - |i|) \rho(t_i) \\ &= \frac{1}{w^2} \int_{-w}^w (w - |t|) \rho(t) dt \\ &= \frac{2}{w^2} \int_0^w (w - t) \rho(t) dt \end{aligned} \quad (3.9)$$

3.5. Measurements to Assess the Mean Aggregate Rate

Note that $\zeta^2(w)$ only depends on the window size and the auto-covariance function $\rho(t)$. We will in general write ζ^2 as a function of w , however it is also a function of $\rho(t)$ which again includes several other parameters.

In the remainder the mean rate is always estimated by means of continuous observation.

3.5 Measurements to Assess the Mean Aggregate Rate

Let the vector $\mathbf{N} = \mathbf{n} = (n_1, n_2, \dots, n_k)$ describe the current state of the system, the number of accepted flows from each class. Conditioned on the system being in a particular state \mathbf{n} and the mean of the aggregate process is:

$$\xi_{\mathbf{n}} = \sum_{i=1}^k n_i \xi_i \quad (3.10)$$

A measurement of the average aggregate rate is then found by conditioning on the system being in a particular state $\mathbf{n} = (n_1, n_2, \dots, n_k)$. An estimate of the aggregate mean is then:

$$\hat{R} = \sum_{i=1}^k \sum_{j=1}^{n_i} \frac{1}{w} \int_0^w K_j(t) \quad (3.11)$$

The covariance of the aggregate rate is $\sum_{i=1}^k n_i \text{cov}(K_i(t), K_i(t+\tau))$ and inserting into (3.9), the expression for the variance of the time average of the aggregate is:

$$\zeta_{\mathbf{n}}^2(w) = \frac{2}{w^2} \int_0^w (w-t) \sum_{i=1}^k n_i \text{cov}(K_i(t), K_i(t+\tau)) dt \quad (3.12)$$

For the homogenous case $\xi_{\mathbf{n}} = \xi_n = n\xi$ and (3.13) and (3.12) is simplified and in state $N = n$:

$$\hat{R} = \sum_{j=1}^n \frac{1}{w} \int_0^w K_j(t) \quad (3.13)$$

$$\zeta_{\mathbf{n}}^2(w) = n\zeta_n^2(w) = n \frac{2}{w^2} \int_0^w (w-t) \sum_{i=1}^k \rho(t) dt \quad (3.14)$$

3.6 Measurement Uncertainty

Measurement error is the deviation of the measured value \hat{R} from the true value $E[R]$. In this work, the observations of the aggregate rate process are *perfect*. We only consider the random error due to the underlying stochastic process. The *uncertainty* of the measurement is a measure of the measurement error. It is characterized by a *confidence interval* which is an interval in which the true

Chapter 3. Measurement Error and MBAC Performance, Concepts and Definitions

value lies with a certain probability. To describe the perfection or quality of a measurement, measurement *accuracy* reflects how close the measurement is to the true value [76]. Accuracy can also be described quantitatively taking into consideration the *relative measurement error*, $\delta/E[R]$. In this thesis accuracy is characterized indirectly by the measurement error or the uncertainty. For example, reducing the confidence interval gives rise to more accurate measurements.

Consider a measurement done in state $\mathbf{n} = (n_1, n_2, \dots, n_k)$. When the number of flows is in the order of a few tens [69] (e.g. 30 flows), the aggregate rate is approximately normally distributed, thus also $\hat{R} \sim \mathcal{N}(\xi_{\mathbf{n}}, \zeta_{\mathbf{n}}^2(w))$. As long as there are no single flow that dominates the entire link, the assumption of the aggregate being normally distributed holds also for the non-homogeneous case [40]. This assumption will be made here. The uncertainty of this measurement can then be described by the $1 - \varepsilon$ confidence interval:

$$\hat{R} - z_{\frac{\varepsilon}{2}} \zeta_{\mathbf{n}}(w) \leq \xi_{\mathbf{n}} < \hat{R} + z_{\frac{\varepsilon}{2}} \zeta_{\mathbf{n}}(w)$$

where $z_{\frac{\varepsilon}{2}}$ is the $(1 - \varepsilon/2)$ quantile of the normal distribution.

It is intuitive to think that in order to achieve a certain confidence level, all that is needed is to increase the window size. However, in order for the above estimate to hold, the requirement is that no flows leave during the window, i.e. the aggregate rate process is stationary with a known distribution. Otherwise the actual estimate becomes incorrect. The flow lifetime therefore sets an upper limit for the window size.

The coefficient of variation (CV) is a normalized measure of dispersion. This measures characterizes the relative measurement error. For a non-zero mean ξ and variance σ^2 , CV is given by:

$$CV = \frac{\sigma}{\xi} \tag{3.15}$$

3.7 Flow Level Traffic Concepts and Performance Measures

Consider just one class of flows. New flows arrive with arrival rate λ and if accepted stay in the system for a lifetime with mean $1/\mu$. At the flow level, the *offered traffic* or equivalently the *offered flow load* is the Erlang load [15] denoted by A . This is the expected number of simultaneous flows if there is no lost traffic and it is given by:

$$A = \frac{\lambda}{\mu} \tag{3.16}$$

Measurement errors cause flows to be accepted such that the average aggregate rate exceeds uc . In this case, the traffic that is carried by the system can be considered *useless traffic*. Only traffic that is carried when the average aggregate rate is at or below the target bandwidth uc is *useful traffic*. The traffic concepts used in this thesis are shown in Fig. 3.5. False rejections increase the blocking probability and decrease the useful traffic. False acceptances increase the useless

3.7. Flow Level Traffic Concepts and Performance Measures

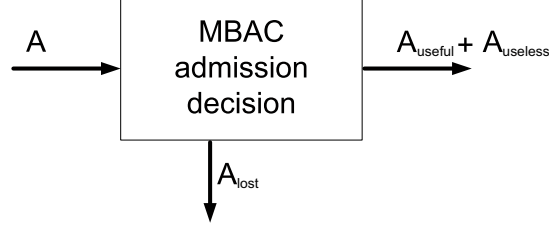


Figure 3.5: Relation between offered traffic, lost traffic, carried useful traffic and carried useless traffic

traffic. For analyzing the performance of MBAC, we define the following flow level performance measures:

- **Probability of False acceptance**, P_{FAcc} , is the probability of accepting a flow when it should have been rejected.
- **Probability of False rejection**, P_{FRej} , is the probability that an arriving flow is rejected when it should have been accepted.
- **Probability of Correct rejection**, P_{CRej} , is the probability that an arriving flow is correctly rejected.
- **Probability of rejection due to multiple arrivals**, P_{Mrej} , is the probability that a flow is rejected due to previous arrival(s) within the window of size w :

$$P_{Mrej} = \int_0^w (1 - e^{-\lambda t}) \frac{dt}{w} = 1 + \frac{e^{-\lambda w} - 1}{\lambda w} \quad (3.17)$$

- **Blocking probability**, P_B , is the probability that an arriving flow is rejected.

$$P_B = P_{FRej} + P_{CRej} + P_{Mrej}$$

- **Carried useful traffic**, A_{useful} , is the traffic that is carried when the system is at or below its target bandwidth.
- **Carried useless traffic**, $A_{useless}$, is the traffic that is carried when the system is above the target bandwidth.
- **Lost Traffic**, A_{lost} , is the traffic that is not accepted by the MBAC.
- **Probability of useless traffic**, $P_{useless}$, is the fraction of time the system is carrying useless traffic.

In an ideal system without measurement error, $P_{FAcc} = 0$ and $P_{FRej} = 0$, and all traffic that is carried is useful traffic.

3.8 Rate model: The ON-OFF source model

The ON-OFF process $I(t)$ is a process that alternates between the values 0 and 1. The source is then described as either being ON ($I(t) = 1$) or in an OFF ($I(t) = 0$) state. The ON and OFF periods (T_{ON} and T_{OFF}) are independent and identically distributed positive random variables, see Fig 3.6. A source of

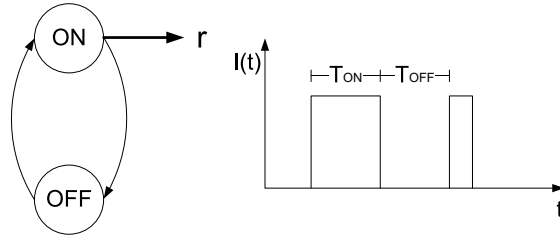


Figure 3.6: The ON-OFF source model alternates between the value 1 and 0

variable bit-rate can be modeled by an ON-OFF rate process, $K(t) = rI(t)$, where r is the peak rate of the source. This ON-OFF model is very realistic in the sense that in a packet switched network, a link is either busy or idle where the rate in the busy state is given by the link capacity. ON-OFF source models are simple and flexible and can represent a wide range of sources such as voice, video and long-range dependent traffic [77], and are much used in loss performance analysis [65], [33] [68]. Among traffic sources with the same mean and peak rate, ON-OFF sources are very useful for performance studies as they are shown to cause the worst case behavior with respect to packet loss probability [65] [68].

The probability that an ON-OFF source is ON is given by its *activity parameter* p [65]:

$$p = 1/B = \frac{\text{mean rate}}{\text{peak rate}} = \frac{m}{r} \quad (3.18)$$

When sampling the stationary ON-OFF source continuously over the measurement window, the time average is given by:

$$\hat{K} = \frac{1}{w} \int_0^w K(t) dt = \frac{r}{w} \int_0^w I(t) dt \quad (3.19)$$

The variance of the time average is given by:

$$\zeta^2(w) = \frac{2}{w^2} \int_0^w (w-t)\rho(t) dt = \frac{r^2}{w^2} \text{var} \left(\int_0^w I(t) dt \right) \quad (3.20)$$

The MMRP ON-OFF process

When developing a source model, the two-state Markov modulated bit-rate process (MMRP) is attractive because of its analytically tractability. This source model can be used to model both speech sources and video sources [78]. For this model,

3.8. Rate model: The ON-OFF source model

the state durations T_{ON} and T_{OFF} follow a negative exponential distribution with mean $1/\alpha$ and $1/\beta$ respectively. The *time constant* for this rate process is $1/\alpha + 1/\beta$.

Key measures for an ON-OFF source are:

- **activity parameter, p :**

$$p = \frac{\alpha}{\alpha + \beta} \quad (3.21)$$

- **mean, ξ :**

$$\xi = r \frac{\alpha}{\alpha + \beta} \quad (3.22)$$

- **variance, σ^2 :**

$$\sigma^2 = r^2 p(1 - p) \quad (3.23)$$

- **auto-covariance function, $\rho(\tau)$:**

$$\rho(\tau) = \text{cov}(K(t), K(t + \tau)) = \sigma^2 e^{-\tau(\alpha + \beta)} \quad (3.24)$$

- **auto-correlation function, $\Psi(w)$:**

$$\Psi(w) = \frac{\text{cov}(R(t), R(t + w))}{\sigma^2} = e^{-w(\alpha + \beta)} \quad (3.25)$$

- **variance of the time average, $\zeta^2(w)$:** The expression for the variance of the time average is:

$$\begin{aligned} \zeta^2(w) &= \frac{2}{w^2} \int_0^w (w - t) \rho(t) dt \\ &= \frac{2r^2 \alpha \beta}{w^2 (\alpha + \beta)^3} \left(w - \frac{1}{\alpha + \beta} (1 - e^{-w(\alpha + \beta)}) \right) \end{aligned} \quad (3.26)$$

The auto-correlation increases remarkably as the time constants increase, that is, $\alpha + \beta$ decreases. This is shown in Fig. 3.7(a).

$\zeta^2(w)$ approaches 0 as the window size increases. The size of $\zeta^2(w)$ depends on the peak rate r , the sum $\alpha + \beta$ and also the ratio between α and β . The value of $\zeta^2(w)$ increases fast as the peak rate r of the sources increases.

Processes with long time constants ($\alpha + \beta$ small), have a large value of $\zeta^2(w)$, resulting in less accuracy.

Chapter 3. Measurement Error and MBAC Performance, Concepts and Definitions

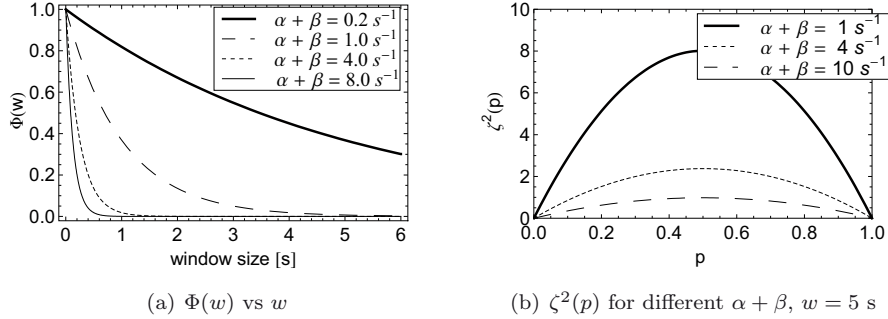


Figure 3.7: (a) The Autocorrelation $\Phi(w)$ as the window size, w , increase. (b) The variance of the time average $\zeta^2(p)$ for different $\alpha + \beta$ as p varies, when $w = 5$ s, $r = 10$ Mbps

Keeping $\alpha + \beta$ constant and $r = 10$ Mbps, $\zeta^2(w)$ reaches its maximum value when the activity parameter $p = 0.5$ (or for burstiness $B = 2$). This is shown in Fig. 3.7(b) for a window size of $w = 5$ s.

The coefficient of variation ζ_{CV} characterizes the relative measurement error, and with the same settings as above, ζ_{CV} is shown in Fig. 3.8.

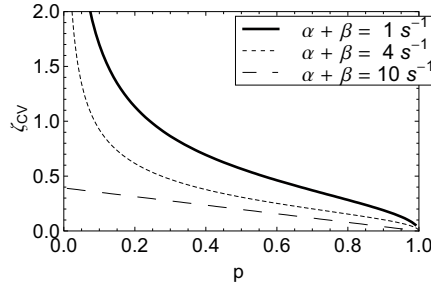


Figure 3.8: The coefficient of variation ζ_{CV} , for different $\alpha + \beta$ as p varies, when $w = 5$ s, $r = 10$ Mbps

3.9 Simulation

We use trace based simulations to check the analytical formulas. All Simulations are implemented in DEMOS (Discrete Event Modeling on Simula) [79]. Each point in the simulation is generated based on several simulation runs with different seeds. The data gathered during a transient period of each run was discarded. The simulation times were set long to ensure high accuracy in the data collected. The 95 % confidence interval is included in the plots. However, in most cases the interval is negligible.

Chapter 4

Quantifying the Uncertainty in Measurements

The quality of measurements are improved when they are taken over a longer measurement window. However, flows leaving within the window result in flawed estimates, thus the flow lifetimes set an upper limit for the window size. Given this window size, how confident can we be that this is not a false acceptance? To make up for the measurement error, the reserved bandwidth for the flows must be reduced by some slack to act as a safeguard. But how large should this slack in bandwidth be?

The objective of this chapter is to provide answers to the above questions by means of analytical analysis of a system with homogeneous flows. In the analysis, the flow level dynamics such as arrival rates and flow lifetimes are not considered and it is assumed that no flows are admitted or depart from the network during the measurement window.

The chapter is organized as follows: Section 4.1 provides the system model with assumptions. The estimation error is treated in detail in Section 4.2. Section 4.3 presents an analytical evaluation, a Simulation Study is given in Section 4.4 before the conclusion is given in Section 4.5.

4.1 System Model and Assumptions

The system under study is described in Section 3.1 and assumptions regarding the flows are given in Section 3.2.

A mix of flow classes will cause increased complexity for the MBAC algorithm and the measurement process. To simplify, only the homogenous case where flows belong to the same class will be considered in this chapter. With the knowledge of the mean aggregate rate of the individual flows ξ , the current average aggregate rate can be specified by the current number of flows. The maximum number of flows the system can handle, the system size, is thus $n_{max} = uc/\xi$.

4.2. Measurement Error and Provisioning

4.2 Measurement Error and Provisioning

In the following a detailed analysis of the estimation error will be presented to give an in-depth understanding of the accuracy of the measurements. A new flow is accepted based on measurements over a complete measurement window w and we shall assume that flows do not leave during this window. Each flow has a rate process $K_i(t)$ with mean ξ .

With $N = n$ flows in the system,

$$\hat{R} = \sum_{i=1}^n \frac{1}{w} \int_0^w X_i(t)$$

is an estimator of the aggregate mean $n\xi$. According to the MBAC algorithm, as long as $\hat{R} \leq \xi n_{max} - \xi$, a new flow will be admitted. When $N = n_{max}$, additional flows should be rejected but due to measurement error, underestimation of the aggregate rate will cause a flow to be admitted erroneously and constitutes a false acceptance. The probability of false acceptance for a flow, P_{FAcc} , depends on the state probabilities and thus requires the inclusion of flow dynamics which will be considered in Chapter 5. Here we will only consider a static system remaining in state n_{max} , excluding the impact of flow dynamics. The requirement is to keep the probability of a false acceptance in state n_{max} below a performance target value ε .

Conditioning on being in the state n_{max} the *conditional performance target* can be written:

$$P(FAcc | n_{max}) = P(\hat{R} + \xi \leq \xi n_{max} | N = n_{max}) \leq \varepsilon \quad (4.1)$$

The probability of underestimating the aggregate mean rate increases as the measurement window size decreases. Because the measurement window size in general is very limited, it may be impossible to meet the required performance target given in (4.1). To cope with this problem, we make some of the bandwidth unavailable by introducing a slack in bandwidth. This slack in bandwidth works as a safeguard and has a size $l\xi$, where l is the number of *levels* and ξ is the size of one level. When the slack bandwidth is added, a new flow is only accepted if:

$$\hat{R} + \xi \leq \xi n_{max} - l\xi, \quad l = 0, 1, \dots, n_{max} \quad (4.2)$$

The performance requirement is rewritten:

$$P(\hat{R} + \xi \leq \xi n_{max} - l\xi | N = n_{max}) \leq \varepsilon, \quad l = 0, 1, \dots, n_{max} \quad (4.3)$$

The task is now to determine the size of the safeguard in terms of number of levels, l . This requires the distribution of \hat{R} , see Section 3.5. With the assumption that $\hat{R} \sim N(n\xi, n\zeta^2(w))$ and conditioned on being in state $N = n_{max}$ we have that:

$$\left(\frac{\hat{R} - \xi n_{max}}{\sqrt{n_{max}}\zeta(w)} \leq z_\varepsilon \right) = 1 - \varepsilon \quad (4.4)$$

Chapter 4. Quantifying the Uncertainty in Measurements

where z_ε is the ε - quantile, $F_z(z_\varepsilon) = 1 - \varepsilon$. Rearranging the terms gives:

$$P(\hat{R} \leq \xi n_{max} + \sqrt{n_{max}} \zeta(w) z_\varepsilon) = 1 - \varepsilon \quad (4.5)$$

and due to symmetry in the normal distribution:

$$P(\hat{R} \leq \xi n_{max} - \sqrt{n_{max}} \zeta(w) z_\varepsilon) = \varepsilon \quad (4.6)$$

Comparing (4.6) and (4.3), the performance target (4.1) will be met if l and $\zeta(w)$ satisfy:

$$\xi(l + 1) = \sqrt{n_{max}} \zeta(w) z_\varepsilon \quad (4.7)$$

Since l , is an integer, the requirement can be expressed:

$$l + 1 = \left\lceil \frac{\sqrt{n_{max}} \zeta(w) z_\varepsilon}{\xi} \right\rceil \quad (4.8)$$

For a given quantile and known $\zeta(w)$, this equation determines the required number of levels, l , in the refined admission control algorithm (4.2).

With the introduction of levels, there will be a region between n_{max} and $n_{max} - l$ where a flow may be admitted in error according to the condition given in (4.2) but will not necessarily be a false acceptance. We define the region between n_{max} and $n_{max} - l$ the critical region and we define a *Hazardous Acceptance* to be the act of admitting a flow when the number of accepted flows is above $n_{max} - l$, see Fig. 4.1. Note again that (4.3) gives the probability of false acceptance

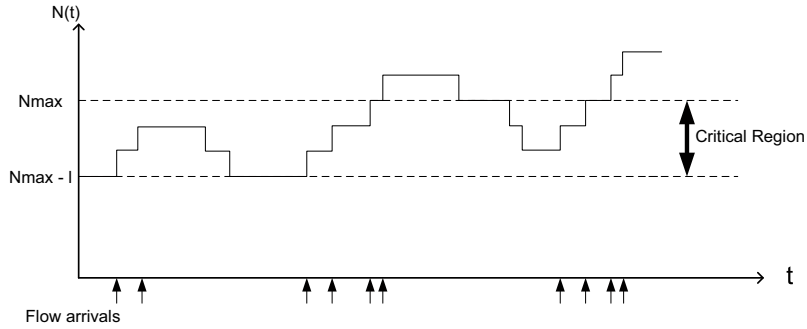


Figure 4.1: Illustration of the critical region

conditioned on the system being in state $N = n_{max}$.

4.2.1 Rate Correlation Between Measurement Windows

The above error analysis does not take into consideration the correlation that may exist between consecutive measurement windows. Acceptance or not is

4.3. Case Study using MMRP sources

solely governed by the constellation of the sources through their rates. If a false acceptance occurs another false acceptance is more likely to occur if there is correlation in the rate process at those two instances. This correlation depends on the time constants of the sources, the flow arrival process and the window size. The auto-correlation between two consecutive measurement windows of size w is given by:

$$\Phi(w) = \frac{\text{cov}(R(t), R(t+w))}{\text{var}(R(t))} \quad (4.9)$$

Correlated measurement windows and the effect this will have on the probability of false acceptance will be demonstrated with an example in the next section.

4.3 Case Study using MMRP sources

In this section we shall study the probability of false acceptance and evaluate the formula (4.8) when the flows are modeled as two-state MMRP sources. This source type is described in Section 3.8.

4.3.1 Provisioning to Control the Probability of False Acceptance

The focus is on a system that is in state $N = n_{max}$. Accepting another flow in this state and the system can no longer guarantee QoS to the flows. The task is now to keep $P(FAcc | N = n_{max})$ below the conditional performance target $\varepsilon = 0.025$. A slack in bandwidth can make up for the measurement error and the relationship between $P(FAcc | N = n_{max})$, slack bandwidth $l\xi$, variance of the time average $\zeta^2(w)$ and system aggregation size n_{max} is given by equation (4.8).

Setting $w = 5$ s, $r = 10$ Mbps and $n_{max} = 50$, Fig. 4.2(a) shows how the number of levels increases as the activity parameter p approaches 0. This is as would be expected; bursty sources (see Section 3.8) are more difficult to handle. Fig. 4.2(a) may be somewhat misleading. One level will have a size corresponding to the mean value of a flow. When the burstiness increases, the mean value decreases. However, since n_{max} is kept constant, also the value of uc decreases, so the relative values will be comparable. In Section 4.3.2, this will be further discussed.

As n_{max} increases, the impact of one single flow on the aggregate will be reduced. This causes again an increase in $P(FAcc | N = n_{max})$ as the difference between n_{max} and $n_{max} + 1$ becomes more and more negligible. To state this another way, the size of the critical region must be increased when the system size n_{max} goes up to keep the level of confidence at a certain value. As an illustration, with $w = 5$ s, $r = 10$ Mbps and $\alpha = \beta = 2$ s⁻¹, Fig. 4.2(b) shows how the required number of levels increases as the system size increases.

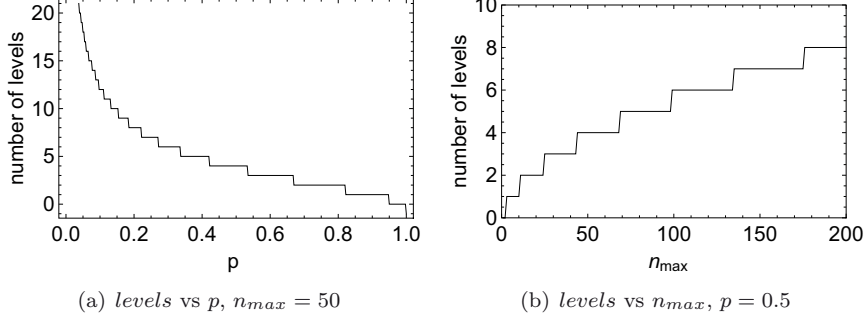


Figure 4.2: Needed safeguard in terms of number of levels, l , to keep $P(FAcc | N = n_{max}) < 0.025$, when $w = 5$ s and $\alpha + \beta = 4$ s⁻¹, as: (a) The activity parameter p of the sources is varied. (b) The system size n_{max} increases

4.3.2 Assuming Worst Case Behavior of Arriving Flows and No Safeguard

In the following we would like to investigate the performance of the MBAC with respect to $P(FAcc | n_{max})$ when it is assumed that the new arriving flow will behave in the worst case scenario, that is, the sending rate is constant at peak rate r . A new flow with peak rate r , is then accepted according to the MBAC algorithm if:

$$\hat{R} + r \leq uc \quad (4.10)$$

Assuming r for the incoming flow as opposed to just the mean rate ξ adds a pessimism to the algorithm. Since ξ is the size of 1 level, then r is equivalent to r/ξ levels. Viewing it this way, we now let l be a non-integer value. For example with $\alpha = \beta$, then r corresponds to two levels and will be equivalent to one level of reduction. As the burstiness increases, the number of levels "built into" r also increases, thus there will be a "naturally" added pessimism as a source becomes more bursty. Will this pessimism be sufficient to protect a system from bursty sources? Let $uc = 100$ Mbps and let $r = 10$ Mbps. Then the number of sources that can be multiplexed, the system size n_{max} , will depend on the flows' activity parameter p . In the following $\alpha + \beta = 4$ s⁻¹, $w = 5$ s and then p is varied. Since the number of flows must be an integer, then depending on the burstiness, $1/p$, the actual uc achieved may be less than 100. In this example we disregard the fact that n can take on only integer values. Fig. 4.3(a) shows how the probability of false acceptance increases as p approaches 0 (the burstiness increases). Bursty sources will thus require a much larger safeguard to protect against false acceptance.

Processes with long time constants ($\alpha + \beta$ small) increase the value of $\zeta^2(w)$ resulting in less accuracy. In the following, let $\alpha = \beta$. With the above settings, this implies that $\xi = 5$ Mbps and $n_{max} = 20$. As can be seen in Fig. 4.3(b), the probability of false acceptance increases rapidly as the time constant increases.

4.4. Comparison with Simulation

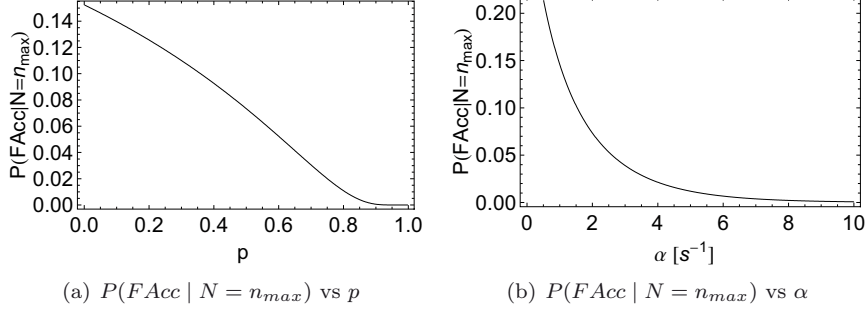


Figure 4.3: Probability of false acceptance, $P(FAcc | N = n_{max})$, when $w = 5$ s: (a) As the activity parameter p is varied when $\alpha + \beta = 4$ s⁻¹. (b) For increasing α , with $\beta = \alpha$

4.4 Comparison with Simulation

To check the defined analytical expression (4.8) and see the effect of correlation between consecutive windows we use simulation. In the following simulations, we use a safeguard of size ξl , where $l = 1$.

4.4.1 Required Number of Flows

For the measurement error analysis, the assumption is that the time average is normally distributed. If the aggregate rate is normally distributed, then also the time average will be normally distributed. In the following we are interested in finding the minimum number of flows that can be multiplexed while still complying to the formula (4.8). Let the MMRP sources have parameters $\alpha = 1$ s⁻¹, $\beta = 4$ s⁻¹ and $r = 25$ Mbps. We vary the window size and let new flows always be available for admission, however we do not let the system enter the states above n_{max} .

Fig. 4.4(a)-4.4(f) show the simulated result together with the theoretically predicted values for different aggregation values, n_{max} . The assumption that \hat{R} is normally distributed seems to result in a good approximation for the probability of false acceptance even when the aggregation level is only 3 flows. As the measurement window is reduced below $w = 2$ s, the simulated values deviates from the theoretically predicted value. This is due to correlation between measurement windows and is discussed next.

4.4.2 Correlation Between Window Measurements

Positive correlation increases the probability of false acceptance beyond what is theoretically predicted by (4.8). To demonstrate this fact, we run simulations where $r = 2$ Mbps, $n_{max} = 20$ and with different values of $\alpha + \beta$. Fig. 4.5(a)-4.5(d) shows how the probability of false acceptance, $P(FAcc | N = n_{max})$, varies with

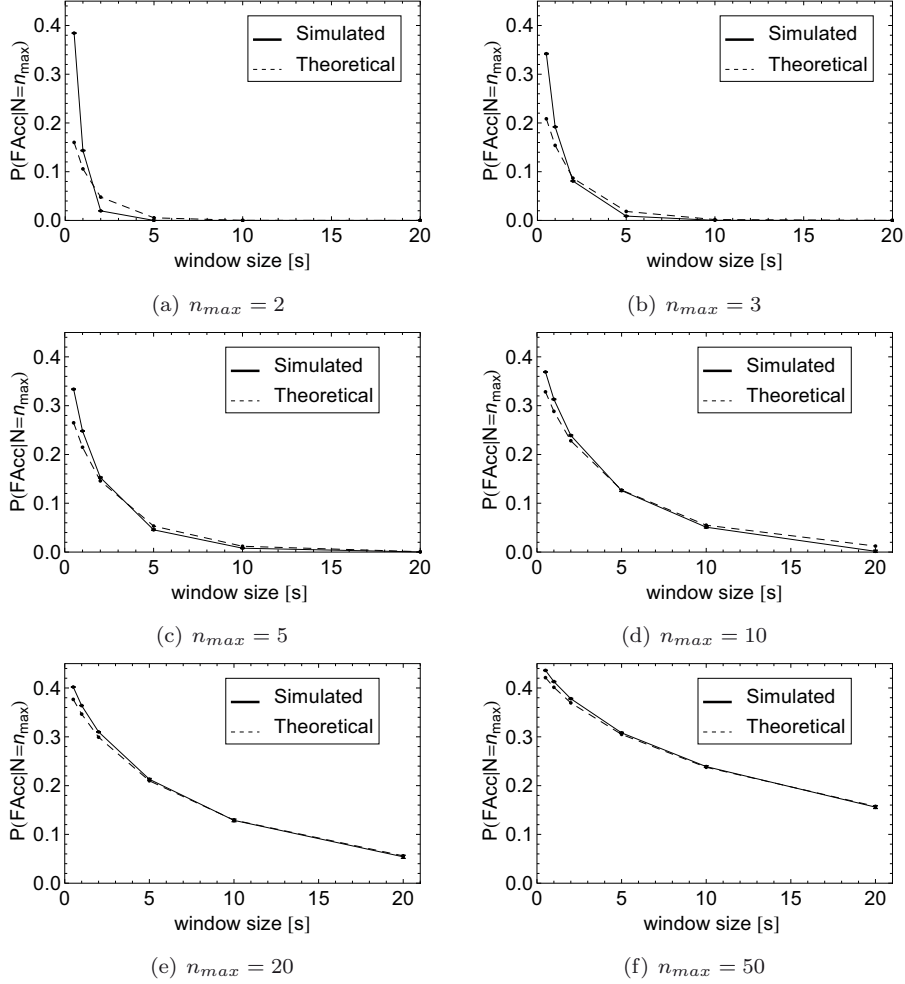


Figure 4.4: The performance measures $P(FAcc | N = n_{max})$ for different system sizes n_{max} , as the measurement window size, w increases

window size. It is evident that as the time constants increase thereby increasing the correlation between consecutive windows, also the probability of false acceptance increases. According to these simulations, in order to be able to neglect the correlation effect, the measurement window size should be at least $(1/\alpha + 1/\beta)$ s.

4.4.3 Sensitivity of the Distribution to the Parameters in the ON-OFF process

In the following we use simulation to study ON-OFF sources with Pareto distributed ON-OFF times and how this affects the probability of false acceptance.

4.4. Comparison with Simulation

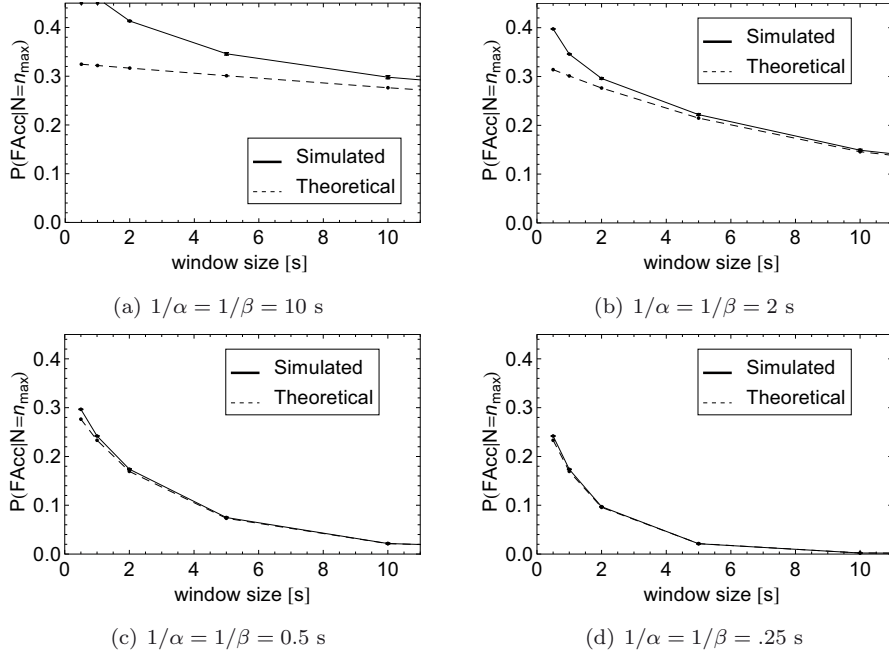


Figure 4.5: Comparing the theoretically predicted value of $P(FAcc | N = n_{max})$ vs window size, w to the simulated value for different settings of α and β

The Pareto distribution has a heavy tail described by two parameters; location and shape. If the shape parameter is less than 1, the distribution has infinite mean and if the shape parameter is less than 2, the distribution has infinite variance. The Pareto distribution is much used when representing Internet sources and for performance evaluation of MBAC algorithms, e.g. [80]. An aggregation of Pareto ON-OFF sources is known to generate long range dependent series [80]. We will study two different scenarios with Pareto sources, one where the shape parameter is 1.2 and one with the shape parameter 2.1. The MMRP sources with negative exponential distributed ON-OFF times with $1/\alpha = 1/\beta = 0.5$ s, will serve as a benchmark. The source models with the respective parameters are shown in Table 4.1. All sources have the same peak rate and burstiness.

Table 4.1: Source types with parameters

Source name	shape	mean ON	mean OFF	peak rate
NegExp	-	0.5 s	0.5 s	2 Mbps
Pareto1	2.1	0.5 s	0.5 s	2 Mbps
Pareto2	1.2	0.5 s	0.5 s	2 Mbps

Chapter 4. Quantifying the Uncertainty in Measurements

The variance of the time average $\zeta^2(w)$ for the different source types is found by simulation. Running 1000 replications, Fig. 4.6 shows the resulting $\zeta^2(w)$ of the three different sources for increasing window size.

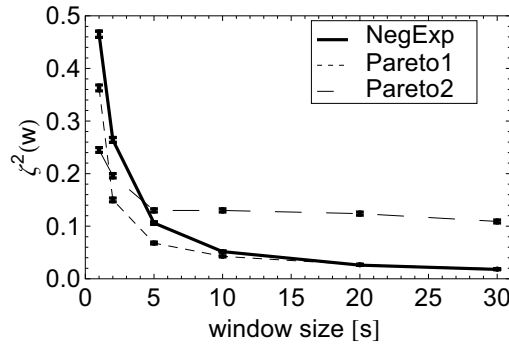


Figure 4.6: The variance of the time average $\zeta^2(w)$ [$Mbps^2$] vs window size w

We see that with the Pareto2, the most heavy tailed, the variance $\zeta^2(w)$ decays very slowly. One would expect that such sources will have a negative impact on the probability of false acceptance. Consider now three different systems, where $n_{max} = 50$. All systems are homogenous with Pareto1 sources, Pareto2 sources and NegExp sources respectively.

For the three different systems, the probability of false acceptance as the window size is varied is shown in Fig. 4.7.

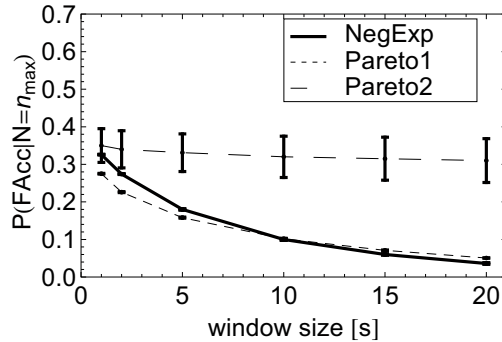


Figure 4.7: Probability of false acceptance for different distributions of the ON and OFF period, $n_{max} = 50$

Due to the very unpredictable behavior of Pareto2, the error bars are large. However, the trend is clear. With very heavy tailed distribution such as Pareto2, increasing the window size will hardly improve the accuracy of the measurements. With the system consisting of Pareto1 sources, the deviation from that of NegExp

4.5. Conclusion

is not significant. In fact for small window sizes, Pareto1 sources appear more predictable resulting in a smaller probability of false acceptance.

4.5 Conclusion

When the number of flows in the system reaches the maximum number the system can handle, n_{max} , no flows should be accepted. In this chapter we have set up analytical expressions for the probability of false acceptance conditioned on that the system is in the state $N = n_{max}$. The task is to keep this probability below a pre-defined value, $P(FAcc | N = n_{max}) < \varepsilon$. If the probability is too high, a slack bandwidth must be added to work as a safeguard. The size of this slack depends on the flow rate characteristics and the measurement window size. Using simulations with two-state Markov modulated rate processes, we have seen that the $P(FAcc | N = n_{max})$ can be accurately predicted even when the aggregation of flows is as low as 3 flows. Positive correlation between consecutive windows increases the probability of false acceptance.

In this chapter only the static system remaining in state n_{max} is considered. In order to determine a proper value for $P(FAcc | N = n_{max})$, the effect of flow dynamics must be included. The distribution of accepted flows will depend on the flow load A , together with the size of the measurement error. This will be discussed in Chapter 5.

Chapter 5

MBAC: Impact of the Measurement Error on Key Performance Issues

In the previous chapter we set up the analytical expressions for evaluating the probability of false acceptances conditioned on the system remaining in the state $N = n_{max}$. The task was to keep this probability below a pre-defined value, $P(FAcc | n_{max}) < \varepsilon$. This probability can be controlled by adding a safeguard and the formula for determining the size of this safeguard was derived.

In order to determine a proper value for ε , the effect of flow dynamics must be included. The distribution of accepted flows will depend on the Erlang load, A , together with the size of the measurement error. There is a trade-off between rejecting too many flows thus wasting resources, and accepting too many flows resulting in QoS violations. In this chapter we study how the measurement errors and flow dynamics impact the performance of MBAC in terms of the performances measures defined in Section 3.7. A simple example shows how the system can be provisioned with a predefined performance criteria.

The remainder of this chapter is organized as follows: First, Section 5.1 introduces the analytical framework we will use to study the measurement error and sets up the expressions of the performance measures. Provisioning is discussed in Section 5.2 and follows up with a case study in Section 5.3, before the conclusion is given in Section 5.4.

5.1 Flow Level and Performance Measures

Based on measurements taken at the rate level, the decision is made at flow level. According to the MBAC decision algorithm, a new flow will be accepted if: $\hat{R} + \xi \leq uc$, where $uc = n_{max}\xi$.

We define the *conditional accepting probability* a_n as:

$$a_n = P(\hat{R} + \xi \leq n_{max}\xi | N = n) \quad (5.1)$$

5.1. Flow Level and Performance Measures

With the assumption that $\hat{R} \sim \mathcal{N}(n\xi, n\zeta^2(w))$, a new arriving flow will be accepted with a probability:

$$\begin{aligned} a_n &= P(Y \leq uc - \xi \mid N = n) \\ &= \frac{1}{\zeta(w)\sqrt{2\pi n}} \int_{-\infty}^{uc - (l+1)\xi} e^{-\frac{(x-n\xi)^2}{2n\zeta^2(w)}} dx \end{aligned} \quad (5.2)$$

We will assume that the arrival rate is such that the probability of more than one flow arrival per window is very small. The lost traffic due to multiple arrivals within the window is thus very small and can be neglected.

The number of flows currently accepted by the MBAC follows a continuous time Markov chain, see Fig. 5.1 and the probability that there are n flows in the system is:

$$P(N = n) = P(n) = \frac{\frac{A^n}{n!} \prod_{x=0}^{n-1} a_x}{\sum_{j=0}^{\infty} \frac{A^j}{j!} \prod_{x=0}^{j-1} a_x} \quad (5.3)$$

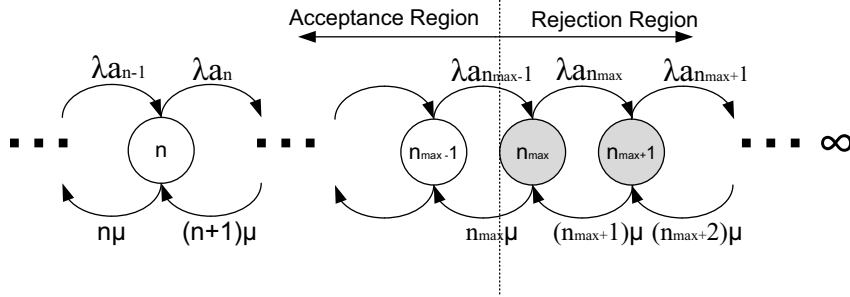


Figure 5.1: State diagram of the number of sources accepted by the MBAC

Implied by (5.3) and as also discussed in [33], the distribution $P(n)$ is indeed insensitive to the distribution of flow lifetime and only depends on the expected flow lifetime.

The system state space can be divided into two regions; the *acceptance region*, $N < n_{\max}$ and the *rejection region*, $N \geq n_{\max}$ see Fig. 5.1.

Based on this defined framework, the flow level performance measures defined in Section 3.7 becomes:

- **Probability of False acceptance**, P_{FAcc} , is the probability that an arriving flow is accepted when it should have been rejected.

$$P_{FAcc} = \sum_{n=n_{\max}}^{\infty} P(n)q_n \quad (5.4)$$

- **Probability of False rejection**, P_{FRej} , is the probability that an arriving flow is rejected when it should have been accepted.

Chapter 5. MBAC: Impact of the Measurement Error on Key Performance Issues

$$P_{FRej} = \sum_{n=1}^{n_{max}-1} (1 - q_n)P(n) \quad (5.5)$$

- **Blocking probability**, P_B , is the probability that an arriving flow is lost.

$$P_B = P_{FRej} + P(N \geq n_{max} \cap rejection) + P_{Mrej} = \sum_{n=1}^{\infty} (1 - a_n)P(n) + P_{Mrej} \quad (5.6)$$

P_{Mrej} is the probability that a flow is lost due to previous arrivals within the window.

When predicting performance in this study, we simply ignore P_{Mrej} thus:

$$P_B = \sum_{n=1}^{\infty} (1 - a_n)P(n) \quad (5.7)$$

- **Carried useful traffic**, A_{useful} , is the expected number of flows in the acceptance region.

$$A_{useful} = \sum_{n=0}^{n_{max}} nP(n) \quad (5.8)$$

- **Carried useless traffic**, $A_{useless}$, is the expected number of flows in the rejection region.

$$A_{useless} = \sum_{n=n_{max}+1}^{\infty} nP(n) \quad (5.9)$$

- **Lost Traffic**, A_{lost} , is the traffic that is blocked from the network.

$$A_{lost} = P_B A \quad (5.10)$$

If there are no measurement errors, the admission controller becomes *ideal*, $\hat{R} = E(R)$ and the distribution of flows is then as for the Erlang Loss system. This system only carries useful traffic and arriving flows will experience a blocking probability given by the Erlang B formula.

5.2. Provisioning

5.1.1 Flow Load and Window Size Limitations

In the analytical formulas used for the performance evaluation, when increasing A , it is indifferent if this is done by increasing the mean flow lifetime $1/\mu$ or increasing the arrival rate λ . In reality this is not true. Many arrivals within a measurement window will increase the blocking probability since the MBAC only admits at most one flow after a measurement update. When predicting the performance using the analytical formulas, we accept just one arrival within a window. It is assumed that the blocking probability due to multiple arrivals is negligible.

For the accepting probability (5.2), the assumption is that the number of flows is constant during a measurement window. If flows leave during the window, then the theoretically predicted performance will be optimistic compared to actual performance in terms of system utilization. Given a constant A , a longer measurement window (thereby reducing the measurement error) can be used for long lifetimes (infrequent arrivals) as compared to short lifetimes (frequent arrivals).

The performance measures can be directly stated upfront if the following assumptions hold:

- **Assumption 1:** The lost traffic due to previous arrivals within the window can be ignored, (see Section 3.7).
- **Assumption 2:** The probability of a flow leaving within a measurement window is small for the actual parameter values μ and n . The probability of a flow leaving within a window depends on the distribution of flow lifetime T_L and the number of currently accepted flows n . Using Little's formula [15], the rate of departure can be directly found as $n/E(T_L) = n\mu$.
- **Assumption 3:** The correlation at arrival points can be neglected. As explained in Section 4.4, positive correlation increases the probability of false acceptance.

5.1.2 False rejections in state $N = n_{max} - 1$

An interesting phenomena happens in state $N = n_{max} - 1$ where flows, for all window sizes, are rejected 50% of the time. This is a natural consequence due to the fact that the number of flows is discrete. Conditioning on being in state $n_{max} - 1$, gives an absolute measurement error of $\delta = \hat{R} - (n_{max} - 1)\xi$. If δ is positive a new flow is falsely rejected. If δ is negative, the flow is correctly accepted. Since δ is symmetric, false acceptance and false rejection are equally likely.

5.2 Provisioning

The QoS provided to the flows can only be guaranteed as long as the number of flows is at or below n_{max} , thus admitting more than n_{max} flows should be avoided.

Chapter 5. MBAC: Impact of the Measurement Error on Key Performance Issues

If the probability of false acceptance is too high, a slack in bandwidth to be used as a safeguard, can be added to make up for the measurement errors.

As in Section 4.2, let the safeguard have increments of size $l\xi$, $0 < l < n_{max}$ and the refined admission control algorithm becomes:

$$\hat{R} + \xi \leq \xi n_{max} - l\xi \quad (5.11)$$

The critical situation arises as soon as the system reaches state n_{max} , where accepting a flow will result in the first false acceptance. With the condition that the system is in state n_{max} we define the *conditional performance requirement*:

$$P(FAcc \mid N = n_{max}) = P(\hat{R} + \xi \leq \xi n_{max} \mid N = n_{max}) \leq \varepsilon \quad (5.12)$$

where ε is termed the *conditional performance target*.

Repeated from Section 4.2, for a given quantile and predefined window size, $P(FAcc \mid N = n_{max})$ can be kept below the target if the number of levels is given by:

$$l + 1 = \left\lceil \frac{\sqrt{n_{max}} \zeta(w) z_\varepsilon}{\xi} \right\rceil \quad (5.13)$$

where z_ε is the ε -quantile of the standard normal distribution. The resulting l can be used for provisioning the system.

Since the MBAC solely estimates the number of flows through measurements, l is independent of the system state. When there are n flows in the system, a new arriving flow will be accepted by the MBAC, with a probability $a_n = P(\hat{R} + \xi \leq (n_{max} - l) \mid N = n)$.

The size of slack, $l\xi$ controls the probability of entering the rejection region by shifting the probability distribution, $P(n)$, to the left. Obviously, if the slack is too large, the MBAC becomes too pessimistic and resources are wasted unnecessarily. The actual performance can be evaluated by means of the performance measures defined in Section 5.1. To the customer, the performance measures of interest are the blocking probability and the probability of false acceptance. The service provider seeks to balance the carried useless traffic and carried useful traffic.

For a given flow load the network can be provisioned to meet a desired performance target. But what flow load should be used? The network should be dimensioned to ensure a small blocking probability under normal loads, say $P_B < 0.01$. At such low loads the probability of entering the rejection region is very small and excellent QoS can be provided to all flows. The problem arises in times of excessive demand. With an *ideal* controller, when the load increases above what is predicted the blocking probability increases to unacceptable high values. However, the QoS to the already admitted flows will not be harmed. With MBAC on the other hand, also the probability of false acceptance and useless traffic increase with increasing loads. Reviewing work in the MBAC literature, it is common practise to test the performance under a heavy flow load, resulting in 50% blocking probability (e.g. [56]) or an infinite load (e.g. [40]).

We do not attempt to answer, exactly what load to use for provisioning purposes. The load must be relatively high, since the main task of MBAC is to

5.3. Case study using MMRP source models

preserve QoS to its users when the load exceeds normal values [20]. Obviously at such loads, the normal blocking probability (e.g. $P_B = 0.01$) cannot be met. Another important issue which we do not consider, is the effect of repeated calls. When a call is blocked it is very likely that it will try again, and as the blocking probability increases the repeated calls increase thus increasing the arrival rate.

5.3 Case study using MMRP source models

Let the flows be modeled by a two-state Markov modulated rate process (MMRP) which is a simple, yet realistic source model used to model both speech sources and video sources. This source model is described in Section 3.8. In the following, $\alpha = \beta = 2 \text{ s}^{-1}$, $r = 2 \text{ Mbps}$, and ξ is then 1 Mbps . The flows have a QoS requirement that can only be guaranteed as long as $uc \leq 50 \text{ Mbps}$ or equivalently $n \leq n_{max} = 50$.

5.3.1 Distribution of number of flows

Before analyzing the MBAC performance with respect to the performance measures defined in Section 5.1, it is useful to have a visualization of how the probability distribution of flows $P(n)$ in (5.3), is affected by the flow load A and window size w . An *ideal* controller is used as a benchmark. For the *ideal* the distribution of number of flows, is the Erlang-B distribution. For MBAC, as the window size

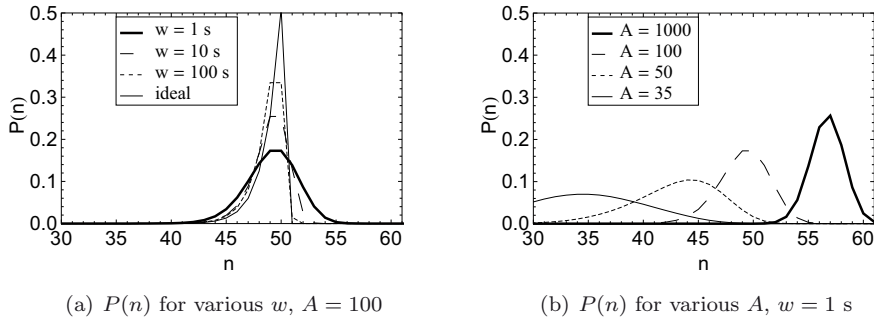


Figure 5.2: Flow probability distribution $P(n)$ when $n_{max} = 50$: (a) For various window sizes, w and $A = 100$ erlang. (b) For various flow loads A and $w = 1 \text{ s}$

increases, the measurements become more and more accurate and the distribution of number of flows approaches the Erlang-B distribution. This is illustrated in Fig. 5.2(a) for a flow load $A = 100$ erlang. At this high load, when the controller is *ideal*, the probability of being in state $N = n_{max}$, is very high. When the window size is large, flows are correctly rejected when the system is in state n_{max} resulting in a quick drop to zero for $N = n_{max} + 1$. Notice how the distribution becomes more and more symmetric around n_{max} when the measurement window is reduced. In this example, when $w = 1 \text{ s}$, the system is half the time in the

Chapter 5. MBAC: Impact of the Measurement Error on Key Performance Issues

rejection region and half the time in the acceptance region. For larger window sizes the deviation from the ideal is due to the fact that flows are rejected 50% of the time when the system is in state $N = n_{max} - 1$.

Fig.5.2(b) shows how the distribution $P(n)$ changes for various offered loads A . For larger A the system spends more and more time in the rejection region.

5.3.2 Performance Analysis

In this section provisioning to fulfill some predefined performance criteria will be demonstrated with an example.

First we shall show how the performance measures are impacted when the offered flow load, A is varied. At this stage we use no slack in the bandwidth (e.i. $l = 0$). Keeping the window size constant at $w = 1$ s, Fig. 5.3(a) shows how the performance measures P_{FAcc} , P_{FRej} and P_B are affected when the flow load increases.

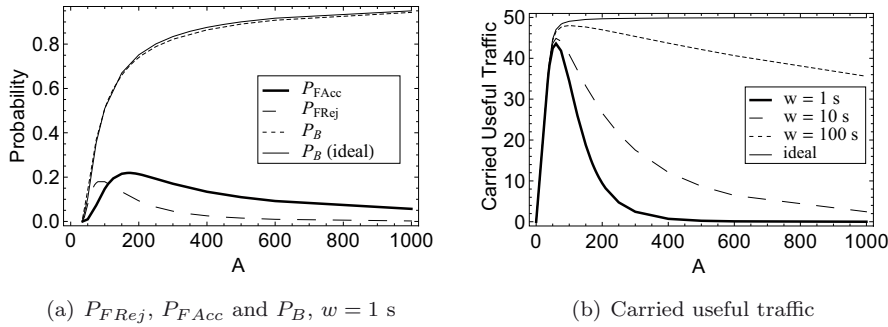


Figure 5.3: Performance as the load increases: (a) Probability of false acceptance, false rejection and overall blocking probability when $w = 1$ s. (b) Carried useful traffic for different window sizes

Low loads result in negligible false acceptance. Instead false rejections cause a slight increase in blocking probability as compared to the *ideal*. At a load of about $A = 60$ erlang, $P_{FAcc} = P_{FRej}$. Then as the load increases, P_{FAcc} increases resulting in a slightly lower blocking probability as compared to the *ideal*. As the load increases towards infinity, P_{FAcc} becomes zero. The reason for this is that the system moves into the rejection region (see Fig. 5.2(b)) and will eventually only carry useless traffic. This is illustrated in Fig. 5.3(b), which shows that as the load increases the carried useful traffic approaches zero. Also shown, is that for larger window sizes, the MBAC approaches the *ideal* and the carried useful traffic falls off slower.

5.3.3 Carried Useful Traffic

We are interested in balancing the trade-off between the overall probability of false rejections and the probability of false acceptances. For the user, the only

5.3. Case study using MMRP source models

concern is the probability of false acceptance and in this example the requirement is $P_{FAcc} < 0.01$. Let the offered flow load be $A = 100$ erlang and let the window size be $w = 1$ s.

The performance plots shown in Fig. 5.4(a), 5.4(b), 5.4(c), and 5.4(d) illustrate the trade-off between blocking and accepting flows when $P(FAcc | N = n_{max})$ is varied.

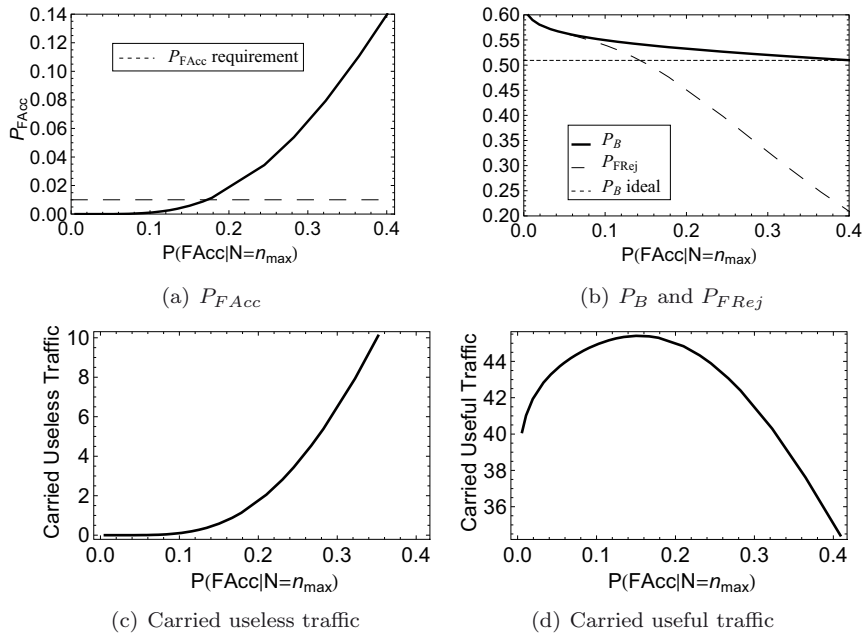


Figure 5.4: The performance measures as $P(FAcc | N = n_{max})$ varies for $A = 100$ erlang and $w = 1$ s: (a) Probability of false acceptance. (b) Probability of false rejection and overall blocking probability. (c) Carried useless traffic. (d) Carried useful traffic

Consider first the performance in the light of the customer. To fulfill the requirement of $P_{FAcc} < 0.01$, Fig.5.4(a), shows that $P(FAcc | N = n_{max}) < 0.17$. At this value the blocking probability is about 5% larger than for the *ideal*.

For the service provider, the concern is false rejections and useless traffic. As $P(FAcc | N = n_{max})$ increases, false rejections fall off (see Fig. 5.4(b)) and the useless traffic increases (see Fig.5.4(c)). Observe in Fig.5.4(d) that a value $P(FAcc | N = n_{max}) = 0.15$, maximizes the carried useful traffic. Reducing the value and the admission controller becomes too strict due to the increase in P_{FRej} . Increasing the value passed this point on the other hand and the admission controller accepts too much useless traffic. In this case, using $P(FAcc | N = n_{max}) = 0.15$, also ensures that $P_{FAcc} < 0.01$. This is an interesting fact since it shows that the carried useful traffic can be maximized by correctly tuning $P(FAcc | N = n_{max})$.

Chapter 5. MBAC: Impact of the Measurement Error on Key Performance Issues

A value of $P(FAcc | N = n_{max})$ can be directly mapped to a safeguard in terms of levels l given by (5.13). This again, means that the carried useful traffic can be increased by adding a safeguard. However if the safeguard is too large, the carried useful traffic will decrease because the MBAC becomes too strict.

The value of l which maximizes the carried useful traffic, will depend on the window size and the offered flow load. Using a higher flow load A and the value of l which maximizes the carried useful traffic will increase. This is shown in Fig. 5.5(a). For example, using the extreme load of $A = 1000$ erlang results in $l = 9$. Decreasing the load, will have the opposite effect, and eventually as the load is reduced further, adding a safeguard will only decrease utilization.

When the measurement error is reduced by increasing the window size w , the required number of levels which optimizes the carried traffic is also reduced (see Fig. 5.5(b)). For large w , the measurement error will become so small that an additional safeguard will only decrease the carried useful traffic.

By observing the shapes of the plots in Fig. 5.5(a) and 5.5(b), we see that the carried useful traffic will drop fast if the number of levels is too low. Adding a too large safeguard is thus better than adding a too small.

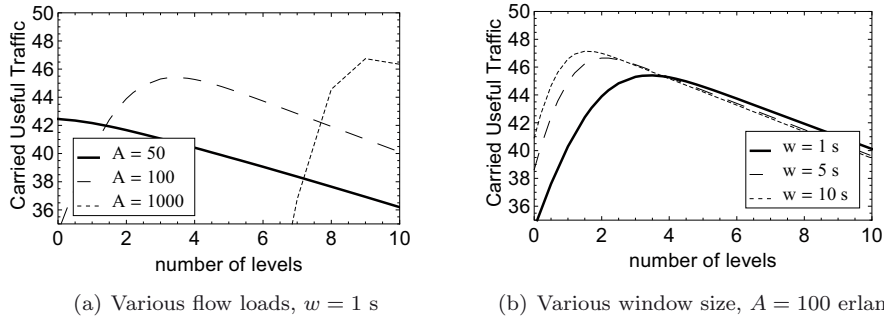


Figure 5.5: The carried useful traffic can be maximized by choosing the right safeguard: (a) The required number of levels for various offered flow loads A , when $w = 1$ s. (b) The required number of levels for various window sizes, w , when $A = 100$ erlang

5.3.4 Impact of Multiple Rejections

When predicting the performance measures above, we considered the offered flow load $A = \lambda/\mu$, without considering the actual values of the flow arrival rate and flow lifetime. When using simulations or a real network setting, the impact of multiple arrivals will impact performance. Clearly, many arrivals within a measurement window will increase the blocking probability since the MBAC only admits at most one flow after a measurement update. As an illustration consider an example where the mean time between flow arrivals is $1/\lambda = 10$ s and the expected flow lifetime of the flows is $1/\mu = 1000$ s. The safeguard in terms of levels

5.3. Case study using MMRP source models

is set to $l = 1$. Without considering P_{MRej} , and only using $A = 100$ erlang in the analytical formulas, the carried useful traffic will only increase as the window size increases. If P_{MRej} is included, this is no longer the case. Fig. 5.6 compares the result when P_{MRej} is not considered (*Predicted*) and when it is included in the analysis, (*MRej*). For window sizes below 7.5 s, the carried useful traffic is higher than what is predicted. This is because the P_{MRej} reduces the probability of false acceptance. As the window size increases past 7.5 s, the useful traffic drops below what is predicted because the increase in the P_{Mrej} increases the probability of false rejections. The MBAC becomes overly pessimistic.

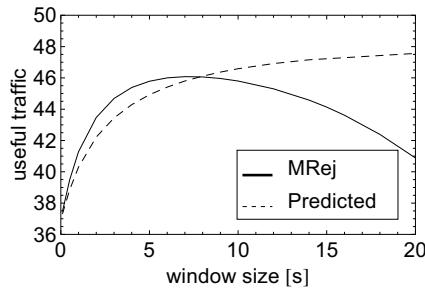


Figure 5.6: Carried useful traffic vs window size when multiple rejections are considered (*Mrej*) and when they are not considered (*Predicted*)

5.3.5 Comparison with Simulation

We check the analytical performance measures using simulation. The parameters used are $\alpha = \beta = 2 \text{ s}^{-1}$, $r = 2 \text{ Mbps}$, $w = 1 \text{ s}$, $n_{max} = 50$ and offered flow load $A = 100$ erlang. When the assumptions stated in Section 5.1.1 hold, it is expected that the simulated result will match closely to what is theoretically predicted.

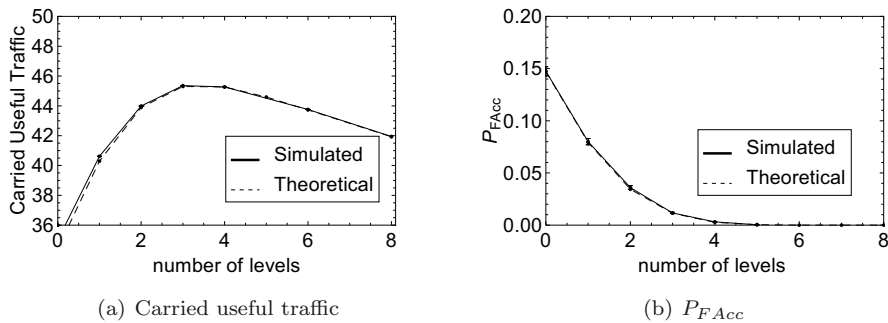


Figure 5.7: Comparing the simulated and theoretically predicted performance as the number of levels increases: (a) Carried useful traffic vs number of levels. (b) Probability of false acceptance P_{FAcc} vs number of levels

Chapter 5. MBAC: Impact of the Measurement Error on Key Performance Issues

The mean time between new flow arrivals is $1/\lambda = 500$ s and expected flow lifetime is $1/\mu = 50000$ s. Fig. 5.7(a) and Fig. 5.7(b), show that the simulated performance matches closely with the theoretically predicted performance.

5.3.6 Sensitivity of the Distribution to the Parameters in Process

In Section 4.4, simulation was used to study ON-OFF sources with Pareto distributed ON-OFF times. By choosing proper values for the shape and location parameter, the distribution can be made very heavy tailed. The results showed that heavy-tailed distribution had a very negative impact on the probability of false acceptance, and hardly showed improvement with increasing window size. However, only the static system remaining in state n_{max} was considered.

In the following example, flow dynamics are also considered. Simulations are used to see the effect heavy-tailed distributions of the ON-OFF states have on the overall probability of false acceptance and carried useful traffic as the safeguard is varied. Two systems are compared: System NegExp consist of homogenous MMRP sources. System Pareto2 consist of Pareto ON-OFF sources. The parameters are the same as used in Section 4.4 and repeated in Table 5.1.

Table 5.1: Source types with parameters

Source name	shape	mean ON	mean OFF	peak rate
NegExp	-	0.5s	0.5 s	2 Mbps
Pareto2	1.2	0.5s	0.5 s	2 Mbps

For both systems, the remaining parameters are $w = 1$ s, $n_{max} = 50$, mean time between flow arrivals is $1/\lambda = 10$ s, and flow lifetime is $1/\mu = 1000$ s.

Fig 5.8(a) shows how the probability of false acceptance decays as the safeguard in terms of number of levels increases. As expected, when the sources are heavy tailed, the probability of false acceptance is higher. What is also seen is that the probability of false acceptance decays slower than the system with MMRP sources. The effect this has on the carried useful traffic is seen in Fig. 5.8(b).

The performance with respect to carried useful traffic for the Pareto2 is much worse but also note how stable the performance remains for a large range of safeguard sizes. This result strengthens the argument that adding a too large safeguard is better than adding a safeguard that is too small.

5.4 Conclusion

This chapter gave an in-depth understanding of how measurement uncertainties and flow dynamics impact the MBAC admission decision.

The probability of false acceptance can be reduced by adding a slack in bandwidth to work as a safeguard. However, if the slack is too large, flows are blocked unnecessarily. With some appropriate performance measures, we showed

5.4. Conclusion

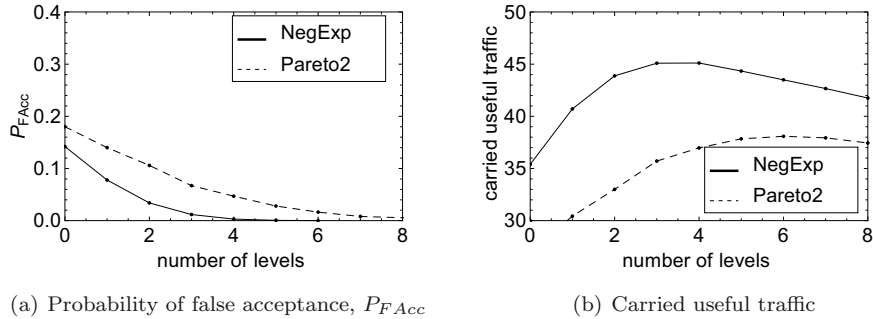


Figure 5.8: Performance comparison of a system with MMRP sources (NegExp) and a system with Pareto ON-OFF sources (Pareto2)

how the system can be provisioned to meet a predefined performance criteria. An interesting fact is that there is a value of the slack which maximizes the carried useful traffic. This means that adding a safeguard of proper size will be of benefit to both the user and the service provider. We also observed that choosing a safeguard that is too large is better than choosing a too small safeguard.

In this chapter, we assumed that the flows were homogenous. In Chapter 8, we develop the analytical models needed to extend this work to the non-homogenous case.

Chapter 6

MBAC and Performance at the Flow Level: A Simulation Study

Most simulation studies in the literature are evaluated with respect to a QoS measure such as a packet loss/delay probability bound versus utilization (or indirectly by the overall blocking probability). In this respect, to deduct the performance of MBAC algorithms there are two main performance questions: 1) *How well the MBAC algorithm can actually meet the QoS target* specified by the requesting flow and the existing flows and 2) *How well the MBAC algorithm can utilize the network resources*. Simulation is then used to verify the actual performance of an MBAC algorithm with respect to the rate level QoS measures such as delay and loss probabilities. After studying various MBAC algorithms, [56] concluded that all algorithms achieve nearly the same performance and have similar deviations from the *ideal* controller, see Fig. 2.6. Due to the large number of system/traffic parameters involved in the MBAC system, pure simulation studies are generally confronted with a too large parameter space and have the difficulty in neatly clarifying the influence and significance of individual parameters.

In the following simulation study we elaborate on the performance measures introduced in Section 3.7 and studied in Chapter 5. We like to investigate how multiple arrivals within a measurement window impact the MBAC performance and gain some insight into how the MBAC can be made more robust due to changes in the offered flow load. Robustness related to types of traffic such as *self-similar* and bursty traffic have been considered [40], [53]. In fact [56] indicates that MBAC may outperform non-measurement based admission control algorithms in the presence of long range dependent traffic.

Less focus has been on the MBAC's ability in being robust when the flow load changes and the rate characteristics of the individual flows are unchanged. According to the definition, the offered flow load is given by the product of the flow arrival rate and the mean flow lifetime. One may expect that when varying the offered flow traffic, it is indifferent if this is done by increasing the mean flow lifetime or increasing the arrival rate but keeping the product of the two constant. This could be the reason why most literature studies of MBAC do not consider

changing the flow arrival rate or the flow lifetime.

By including both a so called *Back-off strategy* and a *Peak-rate strategy* an MBAC can be made more robust in the handling of unexpected high arrival rates and long lasting flows. In this simulation study we investigate further the effect of flow level dynamics and admission control strategies. Instead of studying the performance of MBAC with respect to a packet loss or delay probability, we study how false acceptance and false rejections impact the performance. We greatly reduce the parameter space by assuming that the maximum utilization the system can handle is given and only focus on the question: *Is the current state above or below this maximum utilization?* The impact of the measurement error on the MBAC decision process can then be analyzed in isolation.

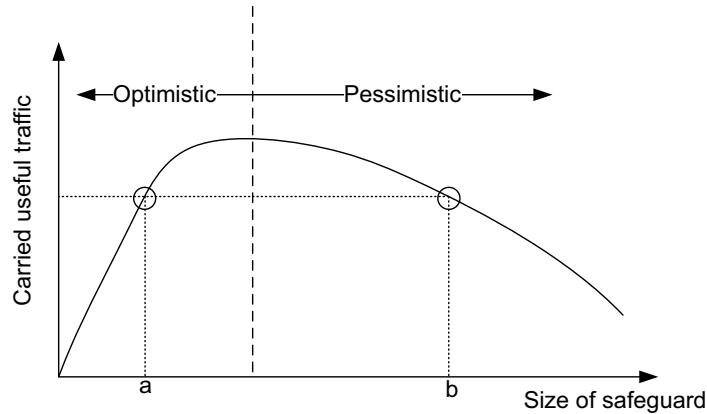


Figure 6.1: Carried useful traffic as the window size increases

Recall from Section 5.3 that the carried useful traffic can be maximized by adding a safeguard. This is illustrated in Fig. 6.1 where the MBAC balances between false rejection and false acceptance. If the safeguard is too small, the carried useful traffic is reduced because the probability of false acceptance is too large and we say that the MBAC is *optimistic*. On the other hand, if the safeguard is too large, the useful traffic is reduced because the probability of false rejection is too large and we say the MBAC is *pessimistic*. The values safeguard size a and b , in Fig. 6.1 show two values of a chosen safeguard which will result in the same carried useful traffic. We argue that the size a will give a better performance. Even though the overall utilization will be lower at this point, the MBAC will carry less *useless traffic* since the probability of false acceptance is smaller. In addition, note also the gradient of the curve, which shows that the carried useful traffic is more sensitive to a change in the safeguard when the size is a .

6.1 Description of MBAC and MBAC Strategies

The system consists of homogenous flows competing for a link of limited capacity c . These flows have peak rate r and mean rate ξ . An MBAC (described in more detail in Section 3.1), is put in place to control access to this link and prevent the average aggregate rate from exceeding its upper limit, uc . As before, in this simulation study, uc is given. When a new flow arrives to the MBAC, it will be accepted if:

$$\hat{R} + \xi \leq uc. \tag{6.1}$$

Additional flows arriving within the measurement window will be accepted or rejected depending on which *admission control strategy* is adopted by the MBAC.

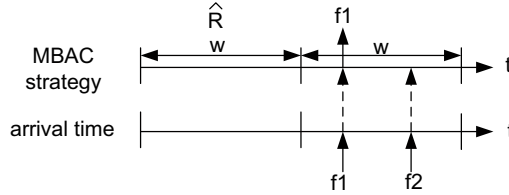


Figure 6.2: Relationship between measurement window updates and flow arrival times. Here, flow $f1$ is accepted if $\hat{R} + \xi \leq uc$ and flow $f2$ will be lost or accepted depending on *strategy*

We consider the following strategies found in the literature:

- **The block all strategy.** With this strategy, the MBAC simply blocks additional flow arrivals within the window. In Fig. 6.2 , flow $f2$ is lost
- **The accept all strategy.** With this strategy, the MBAC does not keep track of flow arrivals within a window and will thus treat all flow arrivals within a window in the same manner. For example, in Fig. 6.2 , if flow $f1$ is accepted, then also flow $f2$ will be accepted.
- **The peak rate strategy.** With this strategy (also called the pessimistic policy in [39], [81], [82]), the MBAC artificially increases the aggregate measurement with the peak-rate r_i of admitted flows within a measurement window. The MBAC algorithm will accept a flow if $\hat{R} + \xi + \sum_i r_i \leq uc$, where $\sum r_i = 0$ at the start of a measurement window. For example in Fig. 6.2, $f2$ will be accepted if $\hat{R} + \xi + r_{f1} \leq uc$, where r_{f1} is the peak rate of flow $f1$.
- **The back-off strategy.** This strategy (termed the back-off policy in [39], [81], [82]) introduced in [37] and later adopted by [33], works by turning down subsequent arrivals after one flow has already been denied admission. When a flow has been rejected admission, no flows will be admitted until a flow has

6.2. The Simulation Setup

left the system. The *Back-off* strategy can only be used if the MBAC can keep track of flow departures. In reality this may not be possible, [20], [40]. Alternatively, a deterministic waiting interval can be added before another flow is accepted [37].

Of the above mentioned strategies it is not difficult to envisage stability problems with the *Accept-all* strategy when the arrival rate is high. In fact the authors in [56] added the *Peak rate* strategy to all the MBAC algorithms under study arguing that the *Peak rate* strategy is a needed feature for an MBAC to be robust to high arrival rates. The *Peak rate* strategy is often considered as being pessimistic [56], [39]. However, when stating this strategy as being pessimistic, it must be an alternative that is better. Clearly, in comparison to the *Block-all* strategy, it is expected that the performance should improve in terms of utilization. One could imagine that by summing up the mean rates of the flows accepted within a window instead of the peak rates, will perhaps improve utilization. We do not discuss this option, since we have not seen examples in the literature of MBAC adopting this implementation. Also, in real systems it is reasonable to assume that only the *Peak rate* of the flows will be known to the MBAC [56]. It is intuitive to expect that employing the *Accept-all* strategy will give a more optimistic performance and thereby increasing the overall utilization. However we draw into question whether this will be too optimistic in that the MBAC will accept more flows than it actually can handle, thus degrading the overall MBAC performance.

The motivation for using the *Back-off* strategy is that even though the MBAC made the correct decision in rejecting a flow at one instance, with a high number of arriving flows, a flow will be accepted as soon as the measured values are under-estimated. It is analytically shown that the back-off strategy is robust to high offered flow loads [33]. Note, that in [65], [37] and [33] the flow load is only increased by increasing the flow lifetime.

6.2 The Simulation Setup

In this simulation each flow is modeled by a two-state MMRP process, which is described in detail in Section 3.8. The parameter settings for these rate processes are $\alpha = \beta = 2 \text{ s}^{-1}$ and $r = 2 \text{ Mbps}$. The maximum number of flows the system can handle is $n_{max} = 50$. The offered flow load will be fixed at $A = 100$ erlang. This high offered load is comparable to what is used in [56]. The reason for using MMRP sources is that these are simple models and the purpose of this study is to solely give an illustration of how certain MBAC strategies impact the MBAC performance. If it is possible to find the maximum number of flows the system can handle (analytically or experimentally), one can easily extend this work to also test an MBAC scheme with other source types.

We add a safeguard in terms of levels (see Section 5.2). Unless otherwise specified, we use $l = 1$, which with the above setting corresponds to replacing the mean rate ξ in (6.2) with the peak rate r of the arriving flows. The performance

Chapter 6. MBAC and Performance at the Flow Level: A Simulation Study

measures are defined in Section 3.7. Specifically in this simulation study, the focus is on maximizing the carried useful traffic.

An ideal admission controller will never admit more than n_{max} flows and all carried traffic is useful. With $A=100$ erlang, the carried traffic given an ideal system can be directly found by the Erlang loss formula and will be 49.06 erlang. Clearly the MBAC can never achieve a higher value than this. We will not use the ideal controller as a benchmark but instead use the analytical framework defined in Section 5.1 to predict the performance, under the assumptions given in Section 5.1.1. As this performance is not based on simulations, we refer to this as the *theoretical* performance.

6.3 Flow Arrival Rate and Admission Decision

In Section 5.3.4, we saw how the impact of multiple rejections becomes more evident with increasing window size because the probability of multiple arrivals within a window increases. If we increase the arrival rate (and decrease the flow lifetime), the effect of multiple blocking within a window will increase. As an illustration, we use the *Peak rate* strategy and run two experiments. In the first experiment, $\lambda = 1/10 \text{ s}^{-1}$ and $1/\mu = 1000 \text{ s}$ and in the second, $\lambda = 1 \text{ s}^{-1}$ and $1/\mu = 100 \text{ s}$. The results are shown in Fig. 6.3(a) which shows that for a window

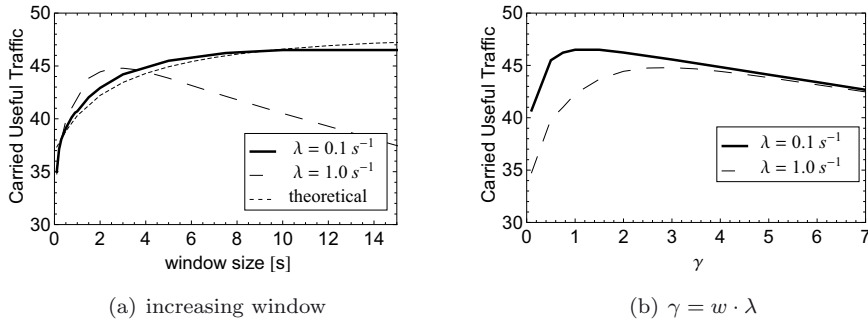


Figure 6.3: Comparing carried useful traffic of high arrival rate ($\lambda = 1 \text{ s}^{-1}$) to low arrival rate ($\lambda = 1/10 \text{ s}^{-1}$): (a) Carried useful traffic as the window size increases. (b) Carried useful traffic as γ increases

size less than 15 s, the low arrival rate results in a performance which closely resembles what is theoretically predicted. A high arrival rate outperforms the theoretically predicted performance with respect to carried useful traffic when the window size is below 4 s. This is because multiple rejections within the window reduce the probability of false acceptance and thus makes up for measurement error. The figure shows that for high arrival rates, a large window size will have a very negative effect on performance. It is evident that the performance depends on the average number of flow arrivals per measurement window, γ :

$$\gamma = \lambda \cdot w \quad (6.2)$$

6.3. Flow Arrival Rate and Admission Decision

Fig. 6.3(b) plots the carried useful traffic for increasing γ , where it can be seen that the MBAC behavior depends mainly on the product $\lambda \cdot w$.

In preference to the *Block-all* strategy, the *Peak rate* strategy results in a better performance at least when the average number of flow arrivals per measurement window is high.

Intuitively, the *Accept-all* strategy results in a lower overall blocking probability and higher utilization. However, if the utilization is higher than what can actually be handled by the network, the traffic can be considered useless. Is it at first glance intuitive to decide which strategy *Accept-all* or *Peak rate* should result in higher carried useful traffic?

Let the arrival rate of flows be $\lambda = 1 \text{ s}^{-1}$ and the flow holding time $1/\mu=100$ s. Fig. 6.4(a) shows the probability of false acceptance vs window size and Fig. 6.4(b) the carried useful traffic vs window size for the three strategies: *Block-all*, *Peak rate*, and *Accept-all*.

For the *Block-all* strategy, the pessimistic behavior of blocking all flows within a window results in a rapid fall in carried useful traffic as the window size increases. For smaller window sizes, the *Block-all* strategy out-performs the *Peak rate* strategy because blocking due to multiple rejections, makes up for the measurement error.

Fig. 6.4(b) shows that for all three strategies the carried useful traffic will reach a maximum for a certain window size. Then, as the window size increases, the performance deteriorates, however not for the same reason. For the *Accept-all* strategy, the degrading performance is due to the MBAC being too optimistic, seen by an increase in false acceptance, see Fig. 6.4(a). For the *Peak rate* and *Block-all* strategy the degrading in performance is due to the MBAC becoming too pessimistic. The *Accept-all* strategy will never be able to carry as much useful

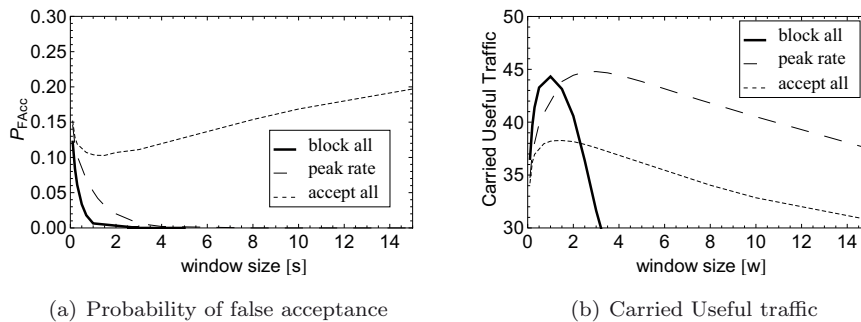


Figure 6.4: Comparing the performance of the *Block-all*, *Peak rate* and *Accept-all* strategy

traffic as neither the *Block-all* strategy nor the *Peak rate* strategy. Even worse, the *Accept-all* strategy will show an increased probability of false acceptance as the window size increases and more and more of the traffic carried will thus become useless.

6.3.1 The Back-Off Strategy

The *Back-off* strategy was proposed to make up for the measurement error and it will thus be interesting to see how adding this strategy will effect performance. In the following the MBAC adopting the *Peak rate* strategy will be compared to the MBAC which adopts both the *Back-off* strategy and the *Peak rate* strategy. In these experiments, the arrival rate is set to $\lambda = 1/10 \text{ s}^{-1}$ and flow lifetime to $1/\mu = 1000 \text{ s}$. First the performance is studied with respect to the window size, when the safeguard is $l = 1$. Fig. 6.5(a) shows that with the *Back-off* strategy, the performance is improved for small window sizes (measurement error high).

The next experiment considers the performance with respect to the size of the safeguard. The window size is $w = 1 \text{ s}$ such that the average flow arrival rate is low and the effect of the *Peak rate* strategy is very small. Fig. 6.5(b) shows that with the *Back-off* policy, when a safeguard is added, the performance deteriorates.

The above examples indicate that the *Back-off* strategy has some effect on improving performance in terms of carried useful traffic when the measurement error is large.

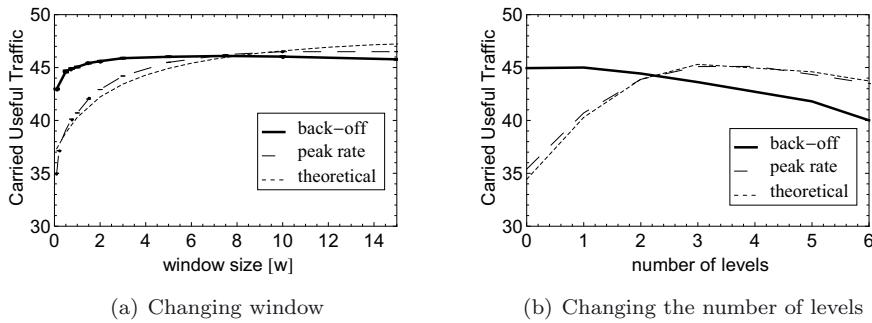


Figure 6.5: Comparing carried useful traffic when *Back-off* is used (back-off) and without the *Back-off* strategy (peak rate)

6.4 Conclusion

When the flow arrival rate to the MBAC increases, the chance of making a false acceptance increases. In addition when the average number of flows arriving within a measurement window increases, the handling of multiple arrivals within a window also becomes important. In this simulation study, we have focused on the impact different admission control strategies will have on the performance of the MBAC decision process. In the literature, the so called *Peak rate* strategy and the *Back-off* strategy are often used to make the admission controller more robust to high flow loads. The authors of these schemes have noted that these strategies improve robustness at the cost of degrading performance [56], [33]. However, we have come to the contradictory conclusion. The flow level performance measures

6.4. Conclusion

carried useful traffic and *probability of false acceptance* show that these strategies in fact improve performance. With the *Back-off* strategy there is a penalty in the form of degraded utilization if a safeguard is added and there are indications that this strategy can make up for measurement errors. This simulation study has also demonstrated how flow level performance measures can be used to study specific MBAC features. A similar MBAC performance analysis should be provided for other types of traffic and other MBAC specific strategies.

Chapter 7

MBAC: The Measurement Error with Non-Homogenous Flows

In this chapter the focus is on controlling the probability of false acceptance when the flows are non-homogenous. Non-homogeneous flows cause increased complexity for the MBAC admission decision algorithm and also the measurement process. We introduce the concept of *similar flows*, which is a restriction to simplify the analytical expressions in a non-homogeneous flow environment.

This chapter is organized as follows: Before the analysis of the measurement error for non-homogeneous flows is given, analytical means to describe the flow rate process and flow mix are needed. Similar flows will be used to describe the rate process of non-homogeneous flows and this concept is introduced in Section 7.1. To describe the flow mix, the multi-dimensional knapsack model is given in Section 7.2, and the measurement error is characterized in Section 7.3. Section 7.4 follows up with a case study to demonstrate the use of the similar flows concept. A conclusion is given in Section 7.5.

7.1 System Assumptions and the Concept of Similar flows

The system under study is described in Section 3.1 and assumptions regarding the flows are given in Section 3.2.

Multiple classes of flows complicate the analytical error analysis. However, assuming that flows are homogeneous (i.e. they belong to the same class) is very restrictive, even if flows are of same type e.g. only video applications. The concept of *similar flows* which is a special case of non-homogeneous flows, is introduced to simplify this analysis. Flows are said to be *similar* if they obey some restrictions on their rates and maximum variances and all have the same auto-correlation function $\Psi(t)$. It is reasonable to assume that a common correlation structure can be found and that it is representative if the number of similar flows is large.

The rate process $K_i(t)$ of a similar flow belonging to class i with mean ξ_i and

7.2. Ideal Admission Controller and the Stochastic Knapsack

variance σ_i^2 has the following requirements:

$$\kappa \leq \xi_i \leq r_{max} \quad (7.1)$$

$$\sigma_i^2 \leq \sigma_{max}^2 \quad (7.2)$$

$$\Psi(t) = \frac{cov(K_i(t), K_i(t + \tau))}{\sigma_i^2} \quad (7.3)$$

where $\kappa > 0$ is a lower bound on the mean rate and σ_{max}^2 is an upper bound on the variance. The maximum number of flows belonging to class i , that can be aggregated on the link is controlled by the mean value ξ_i . The lower bound restriction κ , is necessary in that it limits the number of flows and thereby the variance of the aggregated flows.

7.2 Ideal Admission Controller and the Stochastic Knapsack

Before looking into the measurement error, consider a system where the admission controller has perfect knowledge of the aggregate mean rate. In this system, there is no measurement error and \bar{R} is replaced with the true value $E(R)$ in (3.1). This admission controller is referred to as the *ideal* controller. This *ideal* controller will always accept a flow from class i , when the system is in the *acceptance region*, $E(R) \leq uc - \xi_i$. When $E(R) > uc - \xi_i$ the system is in the *rejection region* and a flow is always rejected. Thus for this system $E(R)$ will never exceed uc . Fig. 7.1 gives an illustration of the two class dependent regions.

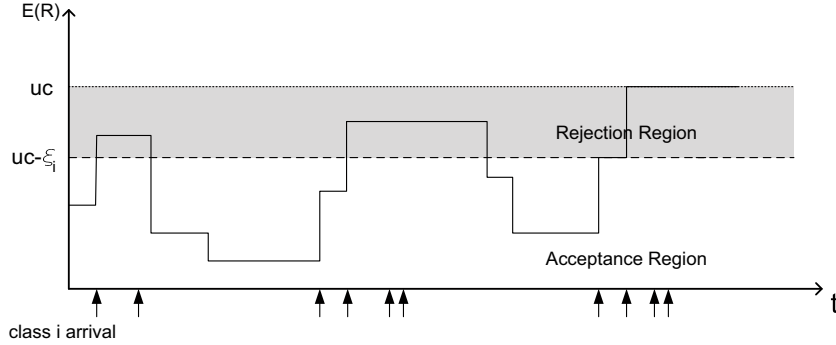


Figure 7.1: Illustration of the rejection region and acceptance region for class i

Let new flows belonging to class i arrive following a Poisson process with parameter λ_i . If the flow is accepted it stays in the system for a negative exponentially distributed lifetime with mean $1/\mu_i$. A flow that is not accepted is lost. The *offered flow load* from class i , is the Erlang load [15] denoted by A_i :

$$A_i = \frac{\lambda_i}{\mu_i} \quad (7.4)$$

Chapter 7. MBAC: The Measurement Error with Non-Homogenous Flows

The system can now be modeled by means of a *stochastic knapsack* and supporting literature for this section can be found in chapter 2 of [57].

Convert uc into l_{max} discrete resource units of size ξ , where ξ is the largest common denominator of all ξ_i such that $uc = l_{max}\xi$. Also convert the mean rate ξ_i of class i , into $b_i, 1 < b_i < l_{max}$ units of size ξ . The stochastic knapsack then consists of l_{max} resource units, where a flow from class i will require b_i resources. If there are enough resources available, the flow is accepted and will occupy b_i resource units throughout the duration of the flow.

Conditioned on the system being in a particular state $\mathbf{n} = (n_1, n_2, \dots, n_k)$, the total amount of resources currently in use is given by \mathbf{bn} , where $\mathbf{b} = (b_1, \dots, b_k)$.

$$\mathbf{bn} = \sum_{i=1}^k n_i b_i \quad (7.5)$$

Define the system state space:

$$\mathcal{S} = \{(n_1, \dots, n_i, \dots, n_k) : \mathbf{bn} \leq l_{max}\} \quad (7.6)$$

For a class i flow, the acceptance region is the set of states \mathbf{n} where the knapsack will admit a class i flow:

$$\mathcal{A}_i = \{\mathbf{n} \in \mathcal{S} : \mathbf{bn} \leq l_{max} - b_i\} \quad (7.7)$$

For a class i flow, the rejection region is the subset of states where a class i flow will be rejected:

$$\mathcal{Q}_i = \{\mathbf{n} \in \mathcal{S} : l_{max} - b_i < \mathbf{bn}\}$$

Denote the equilibrium state probability $\pi(\mathbf{n})$ as the probability of the system being in state \mathbf{n} . The state probabilities are known to have a product form solution [57]:

$$P(\mathbf{N} = \mathbf{n}) = \pi(\mathbf{n}) = G^{-1} \prod_{i=1}^k \frac{A_i^{n_i}}{n_i!} \quad (7.8)$$

where G is the normalization constant:

$$G = \sum_{\mathbf{n} \in \mathcal{S}} \prod_{i=1}^k \frac{A_i^{n_i}}{n_i!} \quad (7.9)$$

Note, that the product form solution is insensitive to the distribution of the flow lifetime and only depends on the mean [57].

Let \mathbf{q}_i be a particular state within the rejection region of class i , $\mathbf{q}_i \in \mathcal{Q}_i$. Define now the conditional blocking probability $P_{\mathcal{Q}_i}(\mathbf{q}_i)$, as the probability of being in a rejection state \mathbf{q}_i given that the system is in the rejection region for class i .

7.3. Measurement Error and Similar Flows

$$P_{\mathcal{Q}_i}(\mathbf{q}_i) = P(\mathbf{N} = \mathbf{q}_i \mid \mathbf{N} \in \mathcal{Q}_i) = \frac{\pi(\mathbf{q}_i)}{\sum_{\mathcal{Q}_i} \pi(\mathbf{n})} \quad (7.10)$$

With (7.8) inserted, $P_{\mathcal{Q}_i}(\mathbf{q}_i)$ does not contain the normalization constant G which is difficult to determine.

7.3 Measurement Error and Similar Flows

Now return to the system controlled by MBAC, where due to measurement errors, flows will be accepted also when the system is in the rejection region (Fig. 7.1) and drive the system above uc . This will again put all the flows at risk of QoS violations. When the QoS requirement is violated, the network provides little or no utility to the end user and the network resources can be considered wasted. From a flow point of view, the probability of false acceptance should be kept low and this is the focus of this analysis.

The critical situation arises as soon as the system transits from the acceptance region into the rejection region as this is where a false acceptance is first made. In this analysis the state space above uc is omitted. Needless to say, if the probability of false acceptance is unacceptable at the boundary of uc , it is also unacceptable when the system resides above uc .

We shall use the stochastic knapsack defined in the previous section to approximately model the state space within the rejection region. The assumption is then that the impact of the measurement error is not significant when determining the conditional blocking probabilities within this region.

Consider a flow from class i , arriving to this system when the system is in one particular state in the rejection region. Define $P_{FAcc|\mathbf{q}_i}$ as the probability of false acceptance given that the system is in the rejection state $\mathbf{q}_i \in \mathcal{Q}_i$. Let this probability be bounded by the *performance target*, ε_i and define the *conditional performance requirement*:

$$\begin{aligned} P_{FAcc|\mathbf{q}_i} &= P(\text{False acceptance} \mid \mathbf{N} = \mathbf{q}_i, \mathbf{q}_i \in \mathcal{Q}_i) \\ &= P(\hat{R} + b_i\xi \leq l_{max}\xi \mid \mathbf{q}_i) \leq \varepsilon_i \end{aligned} \quad (7.11)$$

$P_{FAcc|\mathbf{q}_i}$ increases as the measurement window size decreases. Because the window size in general is very limited, it may be impossible to meet the above performance target. To cope with this, for a class i flow a safeguard of size $l_i\xi$ is added to make up for the measurement error. Viewing each level as a system resource, where l_{max} is the reserved number of resources to the flows, this implies that a flow from class i will see l_i resources as *unavailable resources*. With an added safeguard l_i a new flow belonging to class i , will only be admitted if

$$\hat{R} + b_i\xi \leq (l_{max} - l_i)\xi, \quad l_i = 0, 1, \dots, l_{max} \quad (7.12)$$

Including the safeguard, the conditional performance requirement is rewritten:

Chapter 7. MBAC: The Measurement Error with Non-Homogenous Flows

$$P(\hat{R} + b_i \xi \leq (l_{max} - l_i) \xi \mid \mathbf{N} = \mathbf{q}_i) \leq \varepsilon_i \quad (7.13)$$

Assume now that the flows are similar according to the definition in Section 7.1. This implies that they all share the auto-correlation function $\Psi(t) = cov(K_i(t), K_i(t + \tau)) / \sigma_i^2$. Without the use of similar flows the covariance for every flow must first be determined to find $\zeta_i^2(w)$ separately for each class. Using the property of similar flows significantly simplifies the determination of the variance of the time average of the aggregate rate, which now is directly found by applying (3.12):

$$\zeta_{\mathbf{n}}^2(w) = \sum_{i=1}^k n_i \sigma_i^2 \frac{2}{w^2} \int_0^w (w-t) \Psi(t) dt \quad (7.14)$$

With the assumption from section 3.4.2, that $\hat{R} \sim \mathcal{N}(\xi_{\mathbf{n}}, \zeta_{\mathbf{n}}^2(w))$:

$$\left(\frac{\hat{R} - \xi_{\mathbf{n}}}{\zeta_{\mathbf{n}}(w)} \leq z_{\varepsilon_i} \right) = 1 - \varepsilon_i \quad (7.15)$$

Rearranging and using the symmetric properties of the normal distribution:

$$P(\hat{R} \leq \xi_{\mathbf{n}} - \zeta_{\mathbf{n}}(w) z_{\varepsilon_i}) = \varepsilon_i \quad (7.16)$$

Comparing (7.16) and (7.13), the performance target will be met if l_i and $\zeta_{\mathbf{n}}(w)$ satisfy:

$$\xi(l_i + b_i - l_{max}) + \xi_{\mathbf{n}} = \zeta_{\mathbf{n}}(w) z_{\varepsilon_i} \quad (7.17)$$

With a predefined confidence interval and a fixed window size of w , the required value for l_i given this flow mix is:

$$l_i + b_i = \left\lceil \frac{\zeta_{\mathbf{n}}(w) z_{\varepsilon_i}}{\xi} \right\rceil + b_r \quad (7.18)$$

where $b_r = l_{max} - \frac{\xi_{\mathbf{n}}}{\xi}$ is the number of levels spanning the rejection region, $0 \leq b_r < b_i$.

Given that the system is in a particular rejection state \mathbf{q}_i , formula (7.18) can be used to determine the minimum l_i which meets the performance requirement $P_{F_{acc}|\mathbf{q}_i} \leq \varepsilon_i$.

Chapter 4, gives a detailed analysis of false acceptance when flows are homogeneous. In the homogeneous case, the rejection region only consists of one state, $n = l_{max}$ (i.e $n_{max} = l_{max}$).

For this non-homogeneous case, the analysis is more complex since an arriving flow may have several rejection states, where the probability of erroneous decisions depends on the flow mix of the currently accepted flows. In the above analysis l_i is determined by conditioning on the system being in a particular state. In the real system, MBAC has no other information regarding the system state than the measurement, \hat{R} , thus l_i must be valid for any rejection state. Relevant provisioning methods are:

7.3. Measurement Error and Similar Flows

- **Approximate provisioning:** The safeguard for class i , is the smallest l_i which meets the performance requirement:

$$\begin{aligned} P_{F|Q_i} &= P(\text{False acceptance} \mid Q_i) \\ &= \sum_{\mathbf{q}_i \in Q_i} P_{F|\mathbf{q}_i} P_{Q_i}(\mathbf{q}_i) \leq \varepsilon_i \end{aligned} \quad (7.19)$$

- **Approximate critical state provisioning:** The safeguard for class i is based on the state \mathbf{q}_i which over the long term results in the highest number of false acceptances:

$$\arg \max_{\mathbf{q}_i \in Q_i} \{P(\mathbf{N} = \mathbf{q}_i) P_{F_{Acc|\mathbf{q}_i}}\} \quad (7.20)$$

- **Largest safeguard provisioning:** The safeguard for class i , is based on the rejection state which requires the highest value of l_i :

$$\arg \max_{\mathbf{q}_i \in Q_i} \left\{ \left\lceil \frac{\zeta_{\mathbf{n}}(w) z_{\varepsilon_i}}{\xi} \right\rceil + b_r \right\} \quad (7.21)$$

- **Largest variance state provisioning:** The safeguard for class i , is based on the state within the rejection region resulting in the largest variance of the time average of the aggregate mean:

$$\arg \max_{\mathbf{q}_i \in Q_i} \zeta_{\mathbf{n}} \quad (7.22)$$

In the case where all states in the rejection region have approximately the same mean rate, provisioning using largest safeguard provisioning and largest variance state is the same. Otherwise, largest safeguard provisioning will be the most pessimistic provisioning method. However, in the case where the state probabilities are not known, this may be the safest method for determining l_i .

Section 7.4.1, will give a demonstration of the above provisioning methods.

7.3.1 MBAC With No Knowledge Regarding Flow Class

We have assumed that the MBAC knows which class a flow belongs to upon flow arrival. It may be that the MBAC cannot distinguish between classes of flows. Flows will then be treated by the MBAC as belonging to the same class, thus the chosen size of slack bandwidth must be common for all classes. In addition, the MBAC is typically fed with the peak rate of the arriving flow instead of the mean rate. Assuming peak rate, r_i of the arriving flow, adds a pessimism to the MBAC which can be translated to a slack bandwidth of $r_i - \xi_i$. This slack in bandwidth can then be converted to levels.

7.4 Similar Flows Generated by MMRP Sources

Consider k classes of Similar flows where a flow from class i , is described by the rate process $K_i(t)$. In accordance with the definitions of Similar flows, all flows belonging to the same Similar flow class, have the same auto correlation function $\Psi(t)$ given by (7.3).

Let now $K_i(t)$ be generated by a two state MMRP process shown in Fig. 7.2. If $\beta_i = h - \alpha_i$, all flows will have the same $\Psi(t) = e^{-ht}$ and the flows can be classified as similar. Flows belonging to class i is distinguished by the parameters α_i and r_i .

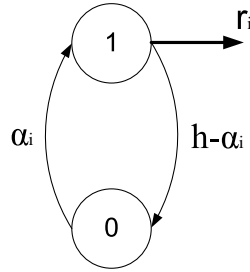


Figure 7.2: MMRP source model

The size of the measurement error is expressed through the variance of the estimated mean rate. From Section 3.8 and using the above parameters, we have that the auto-covariance $cov(K_i(t), K_i(t + \tau))$ of a flow from class i is given by:

$$cov(K_i(t), K_i(t + \tau)) = \sigma_i^2 e^{-\tau(\alpha_i + \beta_i)} = \sigma_i^2 e^{-\tau h} \quad (7.23)$$

where the variance, σ_i^2 , is:

$$\sigma_i^2 = \frac{r_i^2 \alpha_i \beta_i}{(\alpha_i + \beta_i)^2} = \frac{r_i^2 \alpha_i (h - \alpha_i)}{h^2} \quad (7.24)$$

If the state vector \mathbf{n} is known, the variance of the time average is given by inserting (7.24) in (7.14):

$$\zeta_{\mathbf{n}}^2(w) = \sum_{i=1}^k n_i \zeta_i^2(w) = \frac{2}{w^2 h^3} \left(w - \frac{(1 - e^{-wh})}{h} \right) \sum_i n_i r_i^2 \alpha_i (h - \alpha_i) \quad (7.25)$$

7.4.1 Case study using the two-state MMRP Source Model

In this section we will demonstrate the provisioning methods defined in Section 7.3 with a simple example where the flows are similar. Let there be two classes i , $i = 1, 2$ of flows representing real-time video applications (video 1 and video 2) competing for a link controlled by MBAC which admits a flow according to (7.12).

7.4. Similar Flows Generated by MMRP Sources

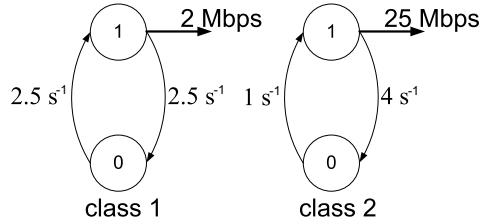


Figure 7.3: Parameter settings for class 1 and class 2

The flows are generated by MMRP processes with parameters shown in Fig. 7.3, which results in $\xi = 1 \text{ Mbps}$, $b_1 = 1$ and $b_2 = 5$.

According to the definition, the flows can be classified as similar flows. The maximum allowable average rate on this link is $\xi l_{max} = 25 \text{ Mbps}$, and the estimate of the average aggregate rate is based on continuous observation over a window size of, $w = 10 \text{ s}$. The task is to control the probability of false acceptance given that the system is in the rejection region, $P_{F|Q_i} < \varepsilon_i$. In this example $\varepsilon_i = 0.025$ for both classes.

Fig. 7.4 shows the state diagram for this system, where a given state is specified by (n_1, n_2) . Flows representing the video 1 class, have a rejection region of 6

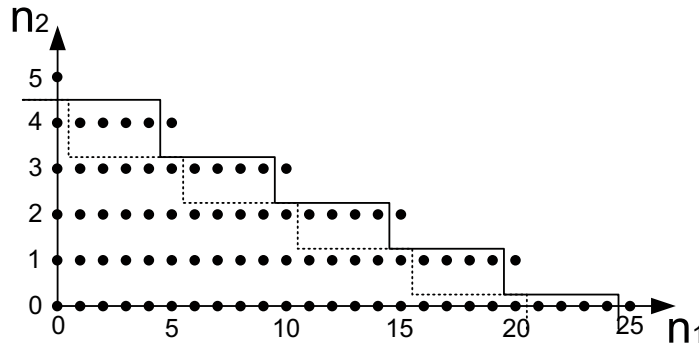


Figure 7.4: The rejection region for class 1 are the states above the solid line. The rejection region for class 2 are the states above the broken line

states, corresponding to the states above the solid line:

$$Q_1 = \{(0, 5), (5, 4), (10, 3), (15, 2), (20, 1), (25, 0)\}$$

The video 2 class, has a rejection region of 26 states, corresponding to the states above the broken line in Fig. 7.4.

Let the offered flow load from the video 1 sources and video 2 sources be $A_1 = 20 \text{ erlang}$ and $A_2 = 5 \text{ erlang}$, respectively. Fig. 7.5 shows the probability of being in the different rejection states for the video 1 class conditioned on the system being in the rejection region.

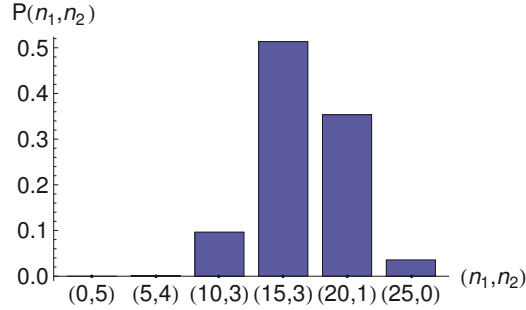


Figure 7.5: Probability of being in the different rejection states conditioned on the system being in the rejection region for the video 1 class

For the video 2 class the conditional probability distribution is presented in Fig. 7.6.

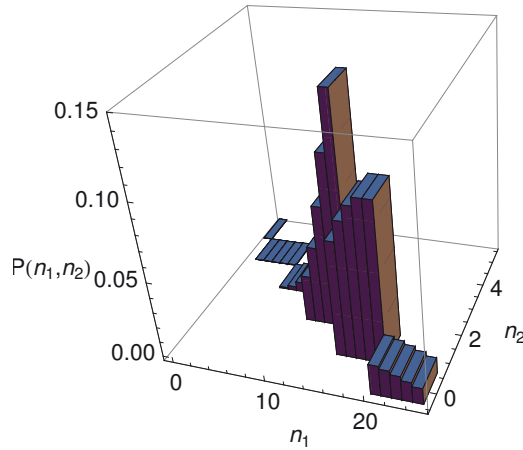


Figure 7.6: Probability of being in the different rejection states conditioned on the system being in the rejection region for the video 2 class

Table 7.1, shows the required safeguard and the resulting $P_{F|Q_i}$ for each of the classes using the different provisioning methods defined in Section 7.3.

Consider first provisioning for the video 1 class. If the system only consisted of video 1 sources, only one reduction level would be required in order to meet the performance target, $P_{F|Q_1} < 0.025$. However, the additional video 2 flows add significant amount of uncertainty to the acceptance decision.

For the video 1 class, since all rejection states have the same mean, the rejection state with the largest variance will also be the rejection state which requires the highest value of l_1 . This will be the state consisting of solely video 2 flows, state (0,5). For largest safeguard provisioning which requires no knowledge of state

7.4. Similar Flows Generated by MMRP Sources

Table 7.1: Required safeguard l_i using different provisioning methods with the corresponding $P(\text{False acceptance} \mid \mathcal{Q}_i)$.

Provisioning method	Class 1, (video 1)	Class 2, (video 2)
Approximate provisioning	$l_1 = 5, P_{F \mathcal{Q}_1} = 0.015$	$l_2 = 3, P_{F \mathcal{Q}_2} = 0.012$
Approximate critical state provisioning	$l_1 = 5, P_{F \mathcal{Q}_1} = 0.015$	$l_2 = 4, P_{F \mathcal{Q}_2} = 0.0048$
Largest safeguard provisioning	$l_1 = 8, P_{F \mathcal{Q}_1} = 9.9E^{-4}$	$l_2 = 7, P_{F \mathcal{Q}_2} = 2.1E^{-4}$
Largest variance state provisioning	$l_1 = 8, P_{F \mathcal{Q}_1} = 9.9E^{-4}$	$l_2 = 4, P_{F \mathcal{Q}_2} = 0.0048$

probabilities, Table 7.1 shows that 8 levels are required corresponding to a 32% drop in system utilization.

System utilization is improved with some knowledge of the state probabilities. The state resulting in the highest number of false acceptances is in this case the most probable state, state (15, 2). Approximate critical state provisioning will thus require a safeguard of size $l_1 = 5$. In this case, the method of approximate provisioning (7.19) will result in the same $l_1 = 5$ as approximate critical state provisioning.

Fig. 7.7 shows how the conditional probability of false acceptance, $P_{F|\mathcal{R}_1}$, is reduced as the safeguard increases for both classes. From Fig. 7.7, it can be seen that $l_1 > 4$ to meet the requirement. Using the approximate or approximate critical state provisioning will both meet the performance target and improve utilization.

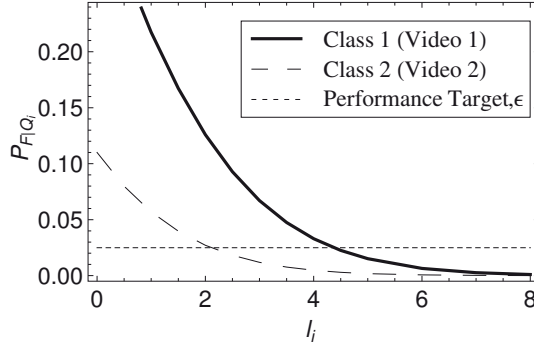


Figure 7.7: For class 1 and class 2: the probability of false acceptance given that the system is in the class dependent rejection region $P_{F|\mathcal{Q}_i}$ for different values of the safeguard l_i , $i = 1, 2$

Now, move to the video 2 class. For video 2, $b_2 = 5b_1$ and one can think of this as a "pessimism" associated with b_2 corresponding to 5 levels of reduction, 4 more levels than video 1 sources. Due to this effect, as can be seen in Table 7.1,

Chapter 7. MBAC: The Measurement Error with Non-Homogenous Flows

only a safeguard of size $l_2 = 4$ is required when provisioning based on the largest variance state (0,5). On the other hand now b_r in (7.18) is no longer always zero.

For video 2, the state which requires the highest value of l_2 , is state (1, 4) and from Table 7.1, we have that the largest safeguard provisioning requires $l_2 = 7$ levels. Approximate critical state provisioning based on the state resulting in the highest number of false acceptances of video 2 sources, state (1, 16), requires $l_2 = 4$. A further improvement in terms of utilization can be achieved if approximate provisioning is used. Table 7.1, shows that the required number of levels is $l_1 = 3$. According to Fig. 7.7, for the video 2 class, to meet the performance requirement, $l_2 > 2$. Using approximate state provisioning will as for *video1*, both meet the performance target and improve utilization.

In this example, the value $P_{F|Q_i} < 0.025$ was given and the sole purpose was to control the probability of false acceptance with the condition that the system was in the class dependent rejection region.

Whether this is an acceptable value can only be determined if the complete state space is considered, not just the rejection region at or below uc . The distribution of accepted flows will depend on the flow load A_i from the different classes together with the size of the measurement error. For a given flow load, the task is then to find a safeguard l_i which balances false acceptances and false rejections. The performance study in Chapter 5 can be expanded to also include similar flows.

7.5 Conclusion

In this chapter, the focus has been on the probability of false acceptance in a system with non-homogeneous flows. Most critical are the system states, within the *rejection region*, where accepting a flow will drive the system to a level beyond its limits. The system can then no longer guarantee QoS to the flows and the service provided to the users becomes inferior.

By conditioning on being in this rejection region, the task to limit the probability of false acceptance by adding a slack in bandwidth. With a given probability and window size, the size of this slack can be stated up front for analytically tractable sources with a known covariance function.

By introducing the concept of *similar flows*, the error analysis with non-homogeneous flows is simplified substantially. Similar flows share a common correlation structure and the error analysis becomes straightforward. In contrast, without this restriction, the correlation structure of each flow must be used which again results in a more complex analysis. In order to determine a proper value for the slack bandwidth, the impact of the measurement error on the distribution of accepted flows must be taken into account. If the slack is too large, the probability of false rejections increases and the system utilization decreases. If the slack is too small, the probability of false acceptance impacts the state transitions such that in reality also the state space above the system limits may be visited. Chapter 5 includes flow dynamics and studies the trade-off between rejecting too many flows thus wasting resources, and accepting too many flows resulting in QoS violations

7.5. Conclusion

and non-usable carried traffic. The study with flow dynamics and the framework defined in chapter 5, can be expanded to also include similar flows.

Chapter 8

Measurement Error when the Variance is Unknown

We state the question: *How long do we have to measure in order to accept a flow with a certain degree of confidence?* We have seen how this question can be answered up front, if the sources are analytically tractable with a known covariance function. In the case when the covariance is unknown this must also be estimated thus requiring a longer observation period before we can reach a certain degree of confidence. In this chapter we show how the variance can be estimated and also motivate the use of similar flows.

If the auto-correlation is known, the auto-covariance of the similar flows can be directly found by measuring the variances for the different classes of flows. This makes for a much easier analytical analysis than would be the case if the auto-covariance was unknown for all lags. By assuming that flows are similar and that the auto-correlation is known, we can analytically state the uncertainty of the estimated variance up front.

8.1 Estimating the variance

In the following we will see how the uncertainty of the measurement error can be found when the variance of the sources is unknown.

Consider just one flow. With continuous observation over the measurement window, an estimate of the variance $\hat{\sigma}^2$ is given by:

$$\hat{\sigma}^2 = \frac{1}{w} \int_0^w (K(t) - \xi)^2 dt \quad (8.1)$$

Before we study the accuracy of this measurement expressed by its variance, let us return to the method of equidistant sampling from Section 3.4.1.

The sample variance $\hat{\sigma}^2$ of the observed sample $X = X_1, X_2, \dots, X_m$ is given by:

8.2. Case Study with ON-OFF Sources

$$\hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^m (X_i - \xi)^2 \quad (8.2)$$

The variance of $\hat{\sigma}^2$ is then:

$$\begin{aligned} \text{var}(\hat{\sigma}^2) &= E[(\hat{\sigma}^2)^2] - E^2[\hat{\sigma}^2] = E \left[\left(\frac{1}{m} \sum_{i=1}^m (X_i - \xi)^2 \right)^2 \right] - E^2 \left[\frac{1}{m} \sum_{i=1}^m (X_i - \xi)^2 \right] \\ &= \frac{1}{m^2} E \left[\sum_i \sum_j (X_i - \xi)^2 (X_j - \xi)^2 \right] - (\text{var}(X_i))^2 \\ &= \frac{1}{m^2} \sum_i \sum_j E [(X_i - \xi)^2 (X_j - \xi)^2] - \sigma^4 \end{aligned} \quad (8.3)$$

Assuming a covariance stationary process (8.3) reduces to:

$$\text{var}(\hat{\sigma}^2) = \frac{1}{m^2} \sum_{h=1}^{m-1} (m-h) \psi_h - \sigma^4 \quad (8.4)$$

Where

$$\psi_{ij} = E [(X_i - \xi)^2 (X_j - \xi)^2] \quad (8.5)$$

With continuous observation, the variance of the estimated continuous time variance, $\theta^2(w)$, is given by:

$$\begin{aligned} \theta^2(w) &= \lim_{\Delta \rightarrow 0 | w=m\Delta} \text{var}(\hat{\sigma}^2) = \lim_{\Delta \rightarrow 0 | w=m\Delta} \left(\frac{\Delta}{w} \right)^2 \sum_{i=1}^{m-1} (m-i) \psi_i - \sigma^4 \\ &= \frac{1}{w^2} \int_{-w}^w (w-t) E [(X_0 - \xi)^2 (X_t - \xi)^2] dt - \sigma^4 \\ &= \frac{2}{w^2} \int_0^w (w-t) E [(X_0 - \xi)^2 (X_t - \xi)^2] dt - \sigma^4. \end{aligned} \quad (8.6)$$

The coefficient of variation, θ_{CV} is given by (see (3.15)):

$$\theta_{CV} = \frac{\theta^2(w)}{\sigma^2} \quad (8.7)$$

8.2 Case Study with ON-OFF Sources

In the following, we shall look at the variance of the estimator $\theta^2(w)$ with ON-OFF sources described in Section 3.8. When sampling the stationary ON-OFF source continuously over the measurement window, $\hat{\sigma}^2$ is given by:

$$\hat{\sigma}^2 = \frac{1}{w} \int_0^w (K(t) - \xi)^2 dt = \frac{r^2}{w} (1-2p) \int_0^w I(t) dt + \xi^2 \quad (8.8)$$

Chapter 8. Measurement Error when the Variance is Unknown

The variance of $\hat{\sigma}^2$, can then be directly found from (8.8), by taking the variance:

$$\theta^2(w) = \text{var}(\hat{\sigma}^2) = \frac{r^4}{w^2}(1 - 2p)^2 \text{var} \left(\int_0^w I(t)dt \right) = k\zeta^2(w) \quad (8.9)$$

where k is the *variance k-factor*:

$$k = r^2(1 - 2p)^2 \quad (8.10)$$

From Section 3.8 have that:

$$\zeta^2(w) = \frac{r^2}{w^2} \text{var} \left(\int_0^w I(t)dt \right) \quad (8.11)$$

so with ON-OFF sources, we can bypass (8.6).

8.2.1 System with Similar Flows

Let the system consist of similar flows. In state $\mathbf{N} = \mathbf{n}$ we have from Section 7.3, that the variance of the time average is given by:

$$\zeta_{\mathbf{n}}^2(w) = \sum_{i=1}^k n_i \sigma_i^2 \frac{2}{w^2} \int_0^w (w-t)\Psi(t)dt \quad (8.12)$$

where $\Psi(t) = \text{cov}(K_i(t), K_i(t + \tau)) / \sigma_i^2$ is the auto-correlation.

If the auto-correlation is known, the auto-covariance of the similar flows can be directly found by estimating the variances, σ_i^2 's for the different classes of flows. These estimates then replace the known value σ_i when determining the confidence interval. The confidence interval will only be an approximation due to the uncertainty of the estimated variance.

The variance is the covariance with zero lag. In the case when the covariance is unknown the covariance must be estimated. When determining the uncertainty of this estimate, the procedure is primarily the same as above, but it is not difficult to envisage the added complexity in the analytical expressions.

8.2.2 The MMRP sources

In the following we will give an illustration of the uncertainty of the estimated variance using the MMRP ON-OFF source model. Just like the variance of the time average, $\zeta^2(w)$, the variance of the estimated variance, $\theta^2(w)$, is reduced as the window size increases. This is illustrated in Fig. 8.1(a) which shows $\theta^2(w)$ for two different settings: setting $\alpha + \beta = 4 \text{ s}^{-1}$, where $\alpha = 1 \text{ s}^{-1}$ and $\beta = 3 \text{ s}^{-1}$ and setting $\alpha + \beta = 10 \text{ s}^{-1}$, where $\alpha = 4 \text{ s}^{-1}$ and $\alpha = 6 \text{ s}^{-1}$. For both settings the peak rate $r = 10 \text{ Mbps}$.

It is the k -factor (8.10), that determines the scaling of $\zeta^2(w)$. Fig.8.1(b) shows how the shape of $\theta^2(w)$ varies as the activity parameter changes. The fact the $\zeta^2(w)$ is zero when $p = 0.5$, is plausible, interesting and at the same time intuitive.

8.2. Case Study with ON-OFF Sources

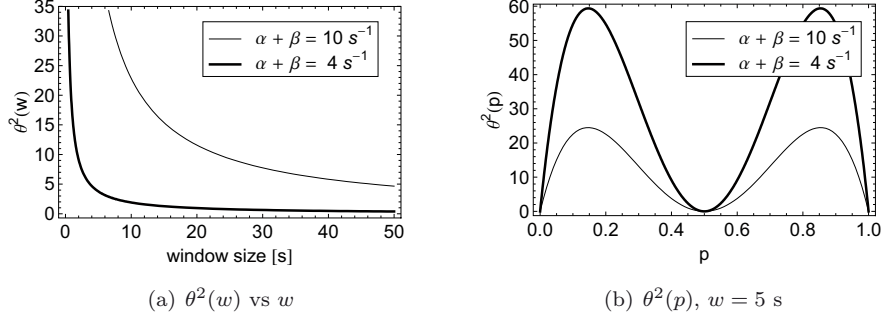


Figure 8.1: Variance of the estimated variance: (a) $\theta^2(w)$ for increasing w , for two different setting of α and β , $\alpha = 1 \text{ s}^{-1}$, $\beta = 3 \text{ s}^{-1}$ and $\alpha = 4 \text{ s}^{-1}$, $\beta = 6 \text{ s}^{-1}$. (b) $\theta^2(p)$ as the activity parameter p is varied for two different settings of $\alpha + \beta$, $w = 5 \text{ s}$

When $p = 0.5$, the estimated variance will always have the same value, since the distance to the mean value, ξ , is the same regardless of the source being in state ON or state OFF. This is in contrast to the variance of the time average which will have its largest value when $p = 0.5$. To compare the variance of the time average with the variance of the estimated variance, we use the coefficient of variation to characterize the relative measurement error.

As a reference, Fig. 8.2 shows the plot of the coefficient of variation of the time average ζ_{CV} together with the coefficient of variation of the estimated variance θ_{CV} , as the activity parameter p is varied. The window size is $w = 5 \text{ s}$, $r = 10 \text{ Mbps}$ and $\alpha + \beta = 4 \text{ s}^{-1}$.

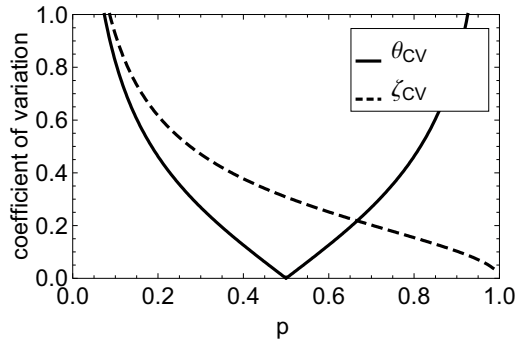


Figure 8.2: Variance of the estimated variance, $\theta^2(p)$ and variance of the time average $\zeta^2(p)$ as the activity parameter p varies when $w = 5 \text{ s}$ and $\alpha + \beta = 4 \text{ s}^{-1}$

8.3 Conclusion

The uncertainty of the admission decision can be stated upfront for analytically tractable sources with a known covariance function. When the covariance function is no longer known, this must also be estimated. This chapter has shown how the similar flow concept can be used to simplify the analysis when the auto-correlation structure is known. In this case the uncertainty can be found by estimating the variances for the different classes of flows. This makes for a much easier analytical analysis than would be the case if the auto-covariance was unknown for all lags. An example with ON-OFF sources shows that the added uncertainty caused by the estimated variance depends on the ON-OFF process parameter settings. An interesting observation is that there will be no added uncertainty in the case where the ON and OFF times have the same duration and the activity parameter is $p = 0.5$.

Chapter 9

Concluding Remarks

This thesis addresses the estimation process and the inherent measurement errors and how these errors impact the admission decision.

By assuming that the maximum utilization the system can handle is given, the focus has been on the question: *When a new requesting flow is accepted into the system, will the system state be above or below the maximum utilization?* The certainty of the answer to this question directly translates to the certainty of the admission decision.

In the literature, the infinite time-scale is used when evaluating the performance of MBAC. The work in this thesis is fundamentally different from previous work in that it considers what happens at a short time-scale governed by measurement updates and flow dynamics.

We have derived analytical expressions to characterize the measurement error. Most critical are the system states, where accepting a flow will drive the system to a level beyond the maximum utilization. For a system in a critical state, the probability of false acceptance can be controlled by adding a safeguard. The size of the safeguard can be stated up front for analytically tractable sources with a known covariance function. Simulations have demonstrated the validity of the defined formula for determining the probability of false acceptance. Not surprisingly, the simulations show that positive correlation between consecutive windows increases the probability of false acceptance.

By defining flow level performance measures we are able to capture the effect measurement errors will have on the MBAC performance when flow dynamics are included. Though it is widely recognized that a safeguard must be added to make up for measurement error, the general belief is that the performance is degraded as the safeguard increases. This is true if one thinks of performance in terms of overall utilization. However, long term utilization will hide what happens at shorter time-scales. For example, an MBAC which alternates between being in a state of very heavy overload following a period of underload may have an overall utilization which is similar to an MBAC which rarely accepts and rejects flows in error.

We have introduced the performance measure *carried useful traffic* which is

a measure of useful utilization. The idea behind this measure, is that when the system accepts more flows than it can handle, the QoS of the flows is at stake and the traffic that is carried is then considered *useless*. Only when the system operates in states at or below its limits, will the carried traffic be considered *useful*. The carried useful traffic can be maximized by a proper setting of the safeguard.

If the safeguard is too small, the carried useful traffic is reduced because the probability of false acceptance is too large. On the other hand, if the safeguard is too large, the useful traffic is reduced because the probability of false rejection is too large. Analytical analysis and simulation show that the *safeguard vs carried useful traffic curve* has a much steeper gradient for shorter safeguards. Adding a too large safeguard is thus better than adding a too small.

This is a promising finding, since in a real network scenario, the safeguard must be roughly estimated. Also, some applications may over-declare their peak rates when requesting admission. This over-declaration can translate to additional safeguard reduction. Due to the MBAC's insensitivity to large safeguards, we expect that over-declarations up to a reasonable level will not have negative effect on MBAC performance.

A simulation study was used to further demonstrate how the defined performance measures can be used to study different admission control strategies and how these strategies impact performance of the MBAC decision process. This study also demonstrated how the concept of carried useful traffic can be used to determine a proper measurement window size. A similar MBAC performance analysis should be provided for other types of traffic and other MBAC specific strategies.

Assuming that flows are homogenous, even if they are of same type of application is very restrictive. By introducing the concept of *similar flows*, the error analysis with non-homogeneous flows is simplified substantially. Similar flows share a common correlation structure and the error analysis becomes straightforward. In contrast, without this restriction, the correlation structure of each flow must be used which results in a more complex analysis.

The concept of similar flows can be further appreciated when estimating the confidence of the admission decision in the case when the covariance is no longer known. In this case, the covariance must also be estimated. Using similar flows, with a known auto-correlation structure, the uncertainty can be directly stated by estimating the variances for the different classes of flows. This makes for a much easier analytical analysis than would be the case if the auto-covariance was unknown for all lags.

The similar flow concept was only analyzed in the static case, for a system remaining in the rejection region. More insight into how an environment with non-homogenous flows will effect MBAC performance can be gained by expanding the work on flow dynamics to include similar flows.

The methodology and framework defined can be used to study a variety of cases to gain insight into MBAC behavior. This insight can be used for developing robust MBAC algorithms.

Bibliography

- [1] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, “Quality of Service Terminology in IP Networks,” *IEEE Communications Magazine*, vol. 41, no. 3, pp. 153–159, 2003.
- [2] E. TC-NA, “Network Aspects: General Aspects of Quality of Service and Network Performance,” ETSI Technical Report ETR 003 (ref. RTR/NA-042102), Tech. Rep., 1994.
- [3] S. Shenker and J. Wroclawski, “RFC2216: Network Element Service Specification Template,” *IETF, Request for Comments*, 1997.
- [4] D. Miras *et al.*, “A Survey on Network QoS Needs of Advanced Internet Applications,” *Working Document of Internet 2 QoS Working Group*, 2002.
- [5] ITU-T, “G.1000: Communication Quality of Service: A Framework and Definitions,” *International Telecommunication Union*, August 2001.
- [6] J. Evans and C. Filsfil, “Deploying DiffServ at the Network Edge for Tight SLAs, part 1,” *IEEE Internet Computing*, pp. 61–65, January · February 2004.
- [7] ITU-T, “G.114: One-Way Transmission Time,” *International Telecommunication Union, Geneva, Switzerland*, vol. 2, 1996.
- [8] R. Braden, D. Clark, and S. Shenker, “RFC1633: Integrated Services in the Internet Architecture: An Overview,” *IETF, Request for Comments*, 1994.
- [9] S. Blake and et al, “RFC2475: An Architecture for Differentiated Services,” *IETF, Request for Comments, August*, Dec. 1998.
- [10] J. Soldatos, E.Vayias, and K. George, “On the Building Block of Quality of Service in Heterogeneous IP Networks,” *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1, pp. 70–89, First Quarter 2005.
- [11] R. M. Mortier, “Multi-Timescale Internet Traffic Engineering,” *Communication Magazine, IEEE*, vol. 40, no. 10, pp. 125–131, October 2002.
- [12] J. W. Roberts and S. Oueslati-Boulahia, “Quality of Service by Flow-Aware Networking,” *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1773, pp. 2197–2207, 2000.

BIBLIOGRAPHY

- [13] S. Oueslati and J. W. Roberts, "A New Direction for Quality of Service: Flow Aware Networking," in *Proc. EuroNGI 1st Conference on Next Generation Internet Networks - Traffic Engineering*, 2005.
- [14] J. Y. Hui, B. C. Res, and N. J. Morristown, "Resource Allocation for Broadband Networks," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1598–1608, 1988.
- [15] V. B. Iversen, "Teletraffic Engineering and Network Planning," *COM, DTU*, 2007.
- [16] V. Jacobson, K. Nichols, and L. Zhang, "RFC2638: A Two-Bit Differentiated Services Architecture for the Internet," *IETF, Request for Comments*, 1999.
- [17] C. N. Chuah, L. Subramanian, R. H. Katz, and A. D. Joseph., "Resource Provisioning Using a Clearing House Architecture," in *Proc. IEEE IWQoS*, 2000.
- [18] Z. Duan, Z. Zhang, and Y. Hou, "Fundamental Trade-Offs in Aggregate Packet Scheduling," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 12, pp. 1166–1177, 2005.
- [19] Y. Jiang, P. J. Emstad, V. Nicola, and A. Nevin, "Measurement-Based Admission Control: A Revisit," in *17th Nordic Teletraffic Seminar (NTS-17)*, 2004.
- [20] J. W. Roberts, "Internet Traffic, QoS and Pricing," in *Proceedings of the IEEE*, vol. 92, no. 9, 2004.
- [21] A. Kortebi, S. Oueslati, and J. W. Roberts, "Cross-Protect: Implicit Service Differentiation and Admission Control," 2004, pp. 56 – 60.
- [22] R. Mortier, I. Pratt, C. Clark, and S. Crosby, "Implicit Admission Control," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 12, pp. 2629 –2639, dec 2000.
- [23] Y. Jiang, A. Nevin, and P. J. Emstad, "Implicit Admission Control for a Differentiated Services Network," in *Next Generation Internet Design and Engineering, 2006. NGI '06. 2006 2nd Conference on*, 0-0 2006, pp. 8 pp. –365.
- [24] D. Ferrari, "Client Requirements For Real-Time Communication Services," *IEEE Communications Magazine*, vol. 28, pp. 65–72, 1990.
- [25] A. Xu, W. Woszczyk, Z. Settel, B. Pennycook, R. Rowe, P. Galanter, J. Bary, G. Martin, J. Corey, and J. R. Cooperstock, "Real-Time Streaming of Multi-channel Audio Data over Internet," *Journal of the Audio Engineering Society*, vol. 48, pp. 1–022, 2000.
- [26] E. W. Knightly and N. B. Shroff, "Admission Control for Statistical QoS: Theory and Practice," *IEEE Network*, pp. 20–29, March/April 1999.

BIBLIOGRAPHY

- [27] J. Roberts, U. Mocchi, and J. Virtamo, *Broadband Network Teletraffic*, ser. Lecture notes in computer science. Springer, 1996, vol. 1155.
- [28] A. W. Berger and W. Whitt, “Extending the Effective Bandwidth Concept to Networks with Priority Classes,” *Communications Magazine, IEEE*, vol. 36, no. 8, pp. 78–83, aug 1998.
- [29] R. G. Addie, M. Zukerman, and T. D. Neame, “Broadband Traffic Modeling: Simple Solutions to Hard Problems,” *Communications Magazine, IEEE*, vol. 36, no. 8, pp. 88–95, aug 1998.
- [30] R. Cruz, “A Calculus for Network Delay. I. Network Elements in Isolation,” *Information Theory, IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, jan 1991.
- [31] Y. Jiang and Y. Liu, *Stochastic Network Calculus*. Springer, 2008.
- [32] S. Shenker and J. Wroclawski, “RFC2215: General Characterization Parameters for Integrated Service Network Elements,” *IETF, Request for Comments*, 1997.
- [33] R. J. Gibbens, F. Kelly, and P. Key, “A Decision-Theoretic Approach to Call Admission Control in ATM Networks,” *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 13, no. 6, pp. 1101–1114, Aug. 1995.
- [34] M. Grossglauser and D. N. C. Tse, “A Framework for Robust Measurement-Based Admission Control,” *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 293–309, June 1999.
- [35] J. W. Roberts, “Traffic Theory and the Internet,” *Communications Magazine, IEEE*, vol. 39, no. 1, pp. 94–99, Jan 2001.
- [36] D. Tse and M. Grossglauser, “Measurement-Based Call Admission Control: Analysis and Simulation,” in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, Apr 1997, pp. 981–989 vol.3.
- [37] N. G. Bean, “Robust Connection Acceptance Control for ATM Networks with Incomplete Source Information,” *Ann. Op. Res*, vol. 48, pp. 357–379, 1994.
- [38] Z. Dziong, M. Juda, and L. G. Mason, “A Framework for Bandwidth Management in ATM Networks - Aggregate Equivalent Bandwidth Estimation Approach,” *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 134–147, Feb. 1997.
- [39] A. W. Moore, “Measurement-Based Management of Network Resources,” Technical Report, University of Cambridge, Cambridge CB3 0FD, United Kingdom, April 2002.

BIBLIOGRAPHY

- [40] M. Grossglauser and D. N. C. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 550–563, Aug. 2003.
- [41] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Services Packet Networks," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 56–70, Feb. 1997.
- [42] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed Admission Control," *IEEE JSAC*, vol. 18, no. 12, pp. 2617–2628, Dec. 2000.
- [43] B. K. Choi and R. Bettati, "Endpoint Admission Control: Network-Based Approach," in *IEEE International Conference on Distributed Computing Systems*, 2001.
- [44] C. Cetinkay, V. Kanodia, and E. W. Knightly, "Scalable Services Via Egress Admission Control," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 69–81, Mar. 2001.
- [45] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," in *In Proceedings of ACM Sigcomm 2000*, 2000, pp. 57–69.
- [46] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks," *IEEE JSAC*, vol. 9, no. 7, pp. 968–981, Sept. 1991.
- [47] S. Floyd, "Comments on Measurement-Based Admission Control for Controlled-Load Services," Lawrence Berkeley National Laboratory, Tech. Rep., 1996, technical Report.
- [48] S. Crosby, I. Leslie, B. McGurk, J. T. Lewis, R. Russell, and F. Toomey, "Statistical Properties of a Near-Optimal Measurement-Based CAC Algorithm," in *IEEE ATM'97*, 1997.
- [49] R. J. Gibbens and F. P. Kelly, "Measurement-Based Connection Admission Control," in *Proc. the 15th International Teletraffic Congress (ITC)*, 1997.
- [50] D. Y. Eun and N. B. Shroff, "A Measurement-Analytic Approach for QoS Estimation in a Network Based on the Dominant Time Scale," *IEEE/ACM Trans. Networking*, vol. 11, no. 2, pp. 222–235, Apr. 2003.
- [51] Y. Jiang, P. J. Emstad, A. Nevin, V. Nicola, and M. Fidler, "Measurement-Based Admission Control for a Flow-Aware Network," in *Proc. EuroNGI 1st Conference on Next Generation Internet Networks - Traffic Engineering*, 2005.
- [52] E. W. Knightly, "Second Moment Resource Allocation in Multi-Service Networks," in *ACM Sigmetrics*, 1997.

BIBLIOGRAPHY

- [53] J. Qiu and E. W. Knightly, "Measurement-Based Admission Control with Aggregate Traffic Envelopes," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 199–210, April 2001.
- [54] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber, "Admission Control and Routing in ATM Networks Using Inferences from Measured Buffer Occupancy," *Communications, IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1778–1784, feb/mar/apr 1995.
- [55] G. Mao, "A Real-Time Loss Performance Monitoring Scheme," *Computer Communications*, vol. 28, no. 2, pp. 150–161, 2005.
- [56] L. Breslau, S. Jamin, and S. Shenker, "Comments on the Performance of Measurement-Based Admission Control Algorithms," in *IEEE INFOCOM*, 2000.
- [57] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, P. J. Hancock, Ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1995.
- [58] D. Mitra, M. Reiman, and J. Wang, "Robust Dynamic Admission Control for Unified Cell and Call QoS in Statistical Multiplexers," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 5, pp. 692–707, jun 1998.
- [59] R. van de Meent, "Network Link Dimensioning - A Measurement and Modeling-based Approach," Ph.D. dissertation, Univ. of Twente, 2006.
- [60] L. Kaining, J. Zhigang, and Z. Jun, "A New Admission Control Approach Based on Prediction," *Journal of Electronics*, vol. 19, no. 2, pp. 209–214, April 2002.
- [61] W.-S. Soh and C.-K. Tham, "Modular Neural Networks for Multi-Service Connection Admission Control," *Computer Networks*, vol. 36, no. 2-3, pp. 181–202, 2001.
- [62] C. Casetti, J. F. Kurose, and D. F. Towsley, "A New Algorithm for Measurement-Based Admission Control in Integrated Services Packet Networks," in *PfHNS '96: Proceedings of the TC6 WG6.1/6.4 Fifth International Workshop on Protocols for High-Speed Networks V*. London, UK, UK: Chapman & Hall, Ltd., 1997, pp. 13–28.
- [63] D. Liu and Y. Cai, "A Heuristic Approach for Measurement-Based Admission Control with Variable-Size Window," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 4, pp. 2537–2541 vol.4, 2001.
- [64] A. M. Abdelaal, H. H. Ali, and H. Sharif, "A Coarse-Grain Analysis for the Performance of Measurement-based Admission Control," *Journal of Computational Methods in Science and Engineering*, no. 6, pp. 349–358, 2006.

BIBLIOGRAPHY

- [65] N. G. Bean, “Statistical Multiplexing in Broadband Communication Networks.” Ph.D. dissertation, University of Cambridge, 1993.
- [66] Y. S. Sun, C. C. Chuang, and Y. D. Lin, “Dynamic Resizing of Utilization Target in Measurement-Based Admission Control,” *Computer Communications*, vol. 24, no. 11, pp. 1097 – 1104, 2001.
- [67] N. G. Duffield, “Asymptotic Sampling Properties of Effective Bandwidth Estimation for Admission Control,” in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, Mar 1999, pp. 1532–1538 vol.3.
- [68] D. H. G. Mao, “Loss Performance Analysis for Heterogeneous ON-OFF Sources With Application to Connection Admission Control,” *IEEE/ACM Trans. On Networking*, vol. 10, no. 1, pp. 125–138, February 2002.
- [69] R. van de Meent, M. Mandjes, and A. Pras, “Gaussian Traffic Everywhere?” in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 2, June 2006, pp. 573–578.
- [70] C. Williamson, “Internet Traffic Measurement,” *IEEE Internet Computing*, vol. 5, no. 6, pp. 70–74, 2001.
- [71] S. Floyd and V. Paxson, “Difficulties in Simulating the Internet,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 4, pp. 392–403, 2001.
- [72] R. O. Onvural, *Asynchronous Transfer Mode Networks Performance Issues*. Artech House, 1995.
- [73] F. Brichet, M. Mandjes, and M. F. Sanchez-Canabate, “Admission Control in Multiservice Networks,” COST 257, Mid-term Seminar,” Interim Report, Decemeber 1998.
- [74] I. VIM, “International Vocabulary of Basic and General Terms in Metrology (VIM),” *International Organization*, vol. 2004, pp. 09–14, 2004.
- [75] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, 2nd ed. Springer, 2002.
- [76] S. G. Rabinovich, *Measurement Errors and Uncertainties*, 2nd ed. Springer-Verlag New York, 2005.
- [77] R. Guerin, H. Ahmadi, and M. Naghshineh, “Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 7, pp. 968 –981, sep 1991.
- [78] M. Schwartz, *Broadband Integrated Networks*, P. Becker, Ed. Prentice Hall, 1996.

BIBLIOGRAPHY

- [79] G. Birtwistle, *DEMOS-a System for Discrete Event Modelling on Simula*, School of Computer Studies, University of Leeds, Woodhouse Lane, Leeds, England LS2 9JT, July 1997.
- [80] S. Jamin and S. Shenker, “Measurement-Based Admission Control Algorithms for Controlled-load Service: A Structural Examination,” in *IEEE/ACM Transactions on Networking*, 1995, pp. 56–70.
- [81] A. Nevin, Y. Jiang, and P. J. Emstad, “MBAC Robustness to Traffic Variations: A Simulation Study,” in *Proc. ECUMN*, 2007.
- [82] —, “Robustness Study of MBAC Algorithms,” in *IEEE Symposium on Computers and Communications, ISCC*, 2008.

