Bottleneck Estimation for Load Control Gateways

Krishna Pandit*, Jens Schmitt*, Martin Karsten+, and Ralf Steinmetz*

Multimedia Communications Lab
Dept. of Electrical Engineering and Information Technology
Darmstadt University of Technology
64283 Darmstadt, Germany

+School of Computer Science
University of Waterloo
200 University Ave W
Waterloo, ON, N2L 3G1, Canada

{Krishna.Pandit, Jens.Schmitt, Ralf.Steinmetz}@kom.tu-darmstadt.de; kalli@bbcr.uwaterloo.ca

Abstract

Providing Quality of Service (QoS) to inelastic data transmissions in a cost-efficient, highly scalable, and realistic fashion in IP networks remains a challenging research issue. In [14], a new approach for a basic, domain-oriented, *reactive QoS system* based on so-called *Load Control Gateways* has been proposed and experimentally evaluated. These load control gateways base their load/admission control decisions on observations of simple, binary marking algorithms executed at internal nodes, which allows the gateways to infer knowledge about the load on each path to peer load control gateways. The original load control system proposal utilizes rather simple, conservative admission control decision criteria. In this paper, we focus on methods to improve the *admission control* decision by using probability theoretical insights in order to better estimate the load situation of a bottleneck on a given path. This is achieved by making assumptions on the probability distribution of the load state of the nodes and analyzing the effect on the path marking probability. We show that even with benevolent assumptions the exact calculation is mathematically intractable for a larger number of internal nodes and develop a heuristic in the form of a Monte Carlo based algorithm. To illustrate the overall benefit of our approach we give a number of numerical examples which provide a quantitative feeling on how the admission control decision can be improved. Overall, we believe the result of this paper to be an important enhancement of the admission control part of the original load control system which allows to make better usage of resources while at the same time controlling statistically the guarantees provided to inelastic transmissions.

Keywords: admission control, reactive resource allocation, ECN.

1. Introduction

1.1 Motivation

The quest to enhance the Internet to offer more sophisticated services than best effort for more demanding applications, mainly in the area of multimedia, continues. Quality of Service (QoS) in the Internet has become a chase to find the right mixture of complexity and functionality. Functionality here means some kind of predictability on the behavior of the data traffic. Extreme examples in this area of conflict are best effort service, which has hardly any complexity and functionality, and Guaranteed Service [23], which achieves per flow predictability at the cost of a high state complexity. Two of the key components of a QoS system are the admission control and the packet forwarding mechanism. The latter is responsible for enabling service differentiation while the admission control is inevitable if absolute guarantees are to be made.

One road to admission control is the knowledge of the load of each node in the domain. But when the number of nodes rises obtaining this exactly becomes a rather complex procedure. However, to obtain knowledge on the load of a path is possible with significantly less complexity. How this information can be obtained within a domain is shown in [14], where a new approach for a basic, domain-oriented, *reactive QoS system* based on so-called *Load Control Gateways* has been proposed and experimentally evaluated. In order to avoid scalability problems, the internal nodes (core routers) have to be kept simple. Therefore, the intelligence, i.e., the admission control decision, is pushed to the edge gateway. These load control gateways base their load/admission control decisions on observations of simple, binary marking algorithms executed at internal nodes, which allows the gateways to infer knowledge about the load on each path to peer load control gateways. The original load control system proposal utilizes rather simple, conservative admission control decision criteria. That is the assumption

· This work is partly funded by the German Research Network Society in the context of the LETSQoS project.

of the worst-case, i.e., that all marking actions stem from one node. As we will show later, this is reasonable when the number of nodes is small, but becomes way to conservative when the number of nodes increases. We propose a method to improve the *admission control* decision by using probability theoretical insights in order to better estimate the load situation of a bottleneck on a given path.

An extended version of this paper [20] is available.

1.2 System Model

In Figure 1, a model network is depicted. The edge gateways, internal nodes and traffic flows are denoted by E_i, I_i and T_i, respectively. A traffic flow enters the network at an ingress gateway, traverses several internal nodes, and leaves through an egress gateway. Every gateway can act as an ingress or egress gateway; this solely depends upon the direction of the traffic. The internal nodes use simple FIFO queueing and are ECN capable [21]. In contrast to using a queue-based marking scheme as, e.g., RED, they mark packets according to their load (Load Based Marking [22]]. Let $s_i \in [0, 1]$ be the load of node *i*. The mapping of load to ECN marks is determined by the marking function $u = f(s)$. An example is a linear marking function without offset, i.e., a node with load *s* marks a packet with probability $u = s$. This is depicted in Figure 2.

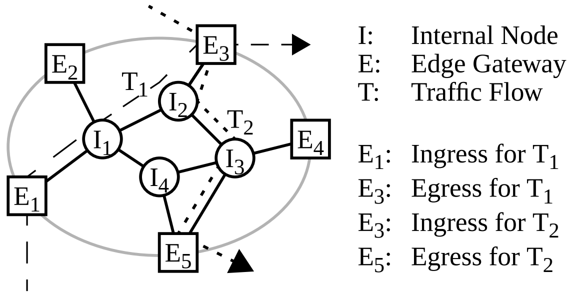

Figure 1: Topological system view

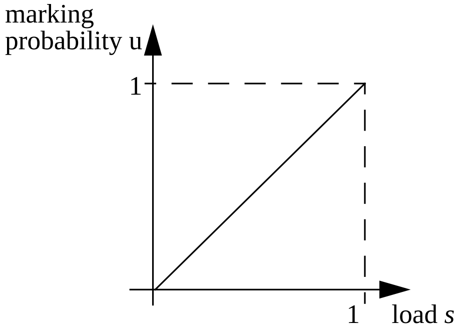

Figure 2: Marking function

In other words, if a node is loaded with e.g., 0.7, it marks 70 % of all its packets. Even though the marking function can be different in each node, we assume all marking functions to be similar.

For most of the paper, we assume every node to have a linear marking function without offset. As our target is to ensure QoS within one domain, we can assume full control over every router and therefore can choose the marking function.

1.3 Related Work

To our knowledge there exists no work on this particular problem, i.e., estimating the bottleneck load from the knowledge of the path load. However, there obviously exists a vast amount of work on providing performance assurances to network traffic. Here we only give a brief overview on work that is fundamental for our research. A more thorough discussion and its relevance to Load Control Gateways is given in [14]. The related work can be classified into the following categories: Flow control and active queue management (AQM), measurement-based admission control, distributed flow admission control and edge-base admission control.

Flow control and active queue management (AQM): The current control paradigm of the Internet is composed of flow control elements at end systems and active queue management schemes at routers. The goal of this distributed resource allocation system is to share available resources efficiently and fairly. The current flow control concept is dominated by TCP's congestion control algorithms and its different flavors [1]. AQM is the notion of how to make decisions on discarding or marking which packets under which conditions. The most prominent example of an AQM scheme is Random Early Discard (RED) [7], which uses an exponential weighted moving average of the queue size and a piecewise-linear probability function over this average queue size to determine the discarding or marking probability for a packet. Most AQM schemes base their decisions on a queue threshold which needs to be exceeded for packets to be discarded or marked (often randomly). The only AQM scheme known to us which provides direct load-based feedback is Load-Based Marking (LBM) [22] by calculating marking probabilities from the measured link load. A drawback of LBM, however, is that it is theoretically restricted to a single resource. All schemes use feedback and reaction as the main mechanism to fairly distribute resources in transient times of overload. Note that all of these schemes assume elastic traffic (with concave utility curves) such that no a-priori flow admission control is required. The other way round, there also is no way of carrying out reliable admission control as it would be desirable for inelastic traffic. In contrast, our work while similar to the above research focuses on admission control for inelastic traffic, which is an inherently different problem. Nevertheless, the admission control system described here is based on the same or very similar AQM mechanisms, which is considered an advantage over QoS systems requiring a completely different set of mechanisms.

Measurement-based admission control: In general, admission control schemes can be distinguished by how the admission decision is made:

- based on worst-case assumptions and resulting in deterministic guarantees,
- based on statistically relaxed assumptions and resulting in statistically controlled guarantees, or
- based on statistical measurements of flow behaviour and resulting in empirical guarantees.

The last approach is the one most related to our work and is commonly called measurement-based admission control. There has been a large amount of research on measurement-based admission control schemes. Different measurement-based admission control algorithms are ([5], [8], [10], [12]). In [3] an extensive comparison of measurement-based admission control schemes results in the conclusion that all schemes perform fairly similar with respect to the utilization they yield.

While our work is similar to measurement-based admission control by taking into account past system behavior, the admission decision here is based on indirect observations rather than direct measurements. Furthermore, in contrast to traditional approaches for measurement-based admission control, it is not a local decision for a single link but an admission decision for a whole path through a subnet including multiplexing with other paths and corresponding cross traffic effects – a much harder problem.

Distributed flow admission control: A further criterion to distinguish different admission control schemes is given by the location where the admission control decision is made: at each forwarding node, at edge nodes between domains, at a centralized server, or at the endpoints of communication. Traditionally, admission control is performed at each node and only if all nodes accept a request, it is granted by the network. More recent admission control schemes do not require to involve all nodes on a path. Distributed flow admission control algorithms are given in ([2], [4], [6], [11],[13], [15], [18], [16], [17], [19]).

Edge-based admission control: Our architectural choice is for edge-based admission control, i.e., we assume independent domains providing QoS for elastic and in particular inelastic traffic flows by using admission control gateways located at the edges of these domains. We are not the first to follow this architectural paradigm, yet the different proposals (including ours) differ very much in their details and in the way they are analyzed, whether being based on theoretical, experimental or just conceptual considerations. Examples are [9] and [24].

1.4 Outline

This paper is organized as follows. In section 2 we introduce our method. Then, in section 3, we analytically discuss a network with two nodes. In section 4 we show that the *N* node case is mathematically not tractable. We then develop a Monte Carlo Algorithm to analyze more nodes and present numerical examples in section 5. Finally, we conclude and point out future work.

2. On the Relationship between Bottleneck Node Load and Path Marking Probability

The egress gateway knows the number of hops from each connected ingress gateway. Further, it can easily determine the number of packets arriving marked, from which the path marking probability, which is denoted by *l*, can be estimated. All this information is sent to the ingress gateway in a so-called load report, where the admission control takes place. For details on how a suitable protocol could be designed to transfer this information from egress to ingress see [14]. There it is also shown that the implementation allows for independent markings, which is crucial for the following equation. The relationship between the path marking probability *l* and the single node marking probability u_i on a path consisting of *N* nodes is given by

$$l = 1 - \prod_{j=1}^{N} (1 - u_j) \qquad (1)$$

The objective of our work presented in this paper is to make a statistical statement on s_i, when *l* is given. More precisely, we want to calculate the probability

$$P\{(\exists i, s_i > s_t) | (l_1 < l < l_2)\} \qquad (2)$$

where l_1, l_2 are the lower and upper bound of the interval, respectively, for the measured *l*. I.e., we look for the probability, that one node has exceeded a certain load threshold under the condition that *l* was measured within a given interval. The reason for using an interval is that the measurement of *l* is subject to a statistical uncertainty. The exact size of the interval is controlled by the confidence into the measurement but shall not further be discussed here. Since s_i can be obtained from u_i by inverting the marking function, it is sufficient to calculate what we define as $P(X)$.

$$P(X) = P\{(\exists i, u_i > u_t) | (l_1 < l < l_2)\} \qquad (3)$$

When the linear marking function without offset is used, Eqs. (2) and (3) are equivalent.

3. The Two Node Case

Consider the excerpt of a network shown in Figure 3. E_1, E_2 denote the load control/edge gateways and I_1, I_2 internal nodes. Let a flow exist between E_1 and E_2 via I_1 and I_2. Additionally, I_1 and I_2 are loaded by arbitrary cross-traffic, which is indicated by the dashed line.

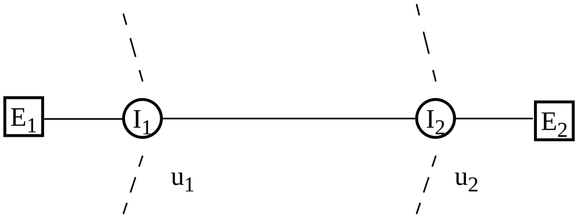

Figure 3: Two node excerpt

With $N = 2$ Eq. (1) yields

$$l = 1-(1-u_1)(1-u_2) = u_1+u_2-u_1u_2 . \qquad (4)$$

Rearranged and u_1, u_2 replaced by y, x for simpler notation and clearer appearance, Eq. (4) becomes

$$y(l,x) = 1-\frac{1-l}{1-x} = \frac{l-x}{1-x} . \qquad (5)$$

which is plotted in Figure 4.

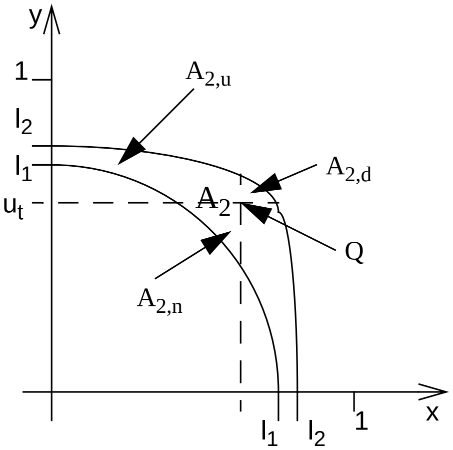

Figure 4: The two node case

The lower and upper curve denotes $y(l_1,x)$ and $y(l_2,x)$, respectively. A_2 denotes the area enclosed by the two curves. The numerical subscript 2 indicates that we are dealing with two nodes. $A_{2,u}$ is the area enclosed by the two curves above the desired threshold u_t. $A_{2,d}$ is the area between the two curves with the constraint that $x > u_t$ and $y > u_t$. Complementary, $A_{2,n}$ is the area between the two curves with the constraint that $x < u_t$ and $y < u_t$. The point of intersection is denoted by $Q = (u_t, u_t)$. We assume the load states of the single nodes to be uniformly distributed. This might be considered a controversial assumption, however we believe it can be enforced by choosing an appropriate topology and routing mechanism. It is the best assumption if there is no further knowledge of the topology, given that the network is designed such that no a priori bottlenecks or hot spots are intentionally created. The probability that one node is loaded beyond u_t under the condition that l is in the specified interval is then given by

$$P\{(u_1 > u_t)|(l_1 < l < l_2)\} = \frac{A_{2,u}}{A_2} . \qquad (6)$$

The probability $P(X)$ from Eq. (3) can therefore be given directly by

$$P(X)_2 = P\{(\exists i, u_i > u_t)|(l_l < l < l_u)\} = \frac{2A_{2,u}-A_{2,d}}{A_2} , \qquad (7)$$

or, by the complement,

$$P(X)_2 = 1-\frac{A_{2,n}}{A_2} . \qquad (8)$$

Which one is better suited for computation depends on where Q lies with respect to the curves $y(l_1,x)$ and $y(l_2,x)$, i.e. below, above or in between. In the extended version [20], results for all cases are shown. Here we limit the derivation to giving equations for each area mentioned in Eqs. (7) and (8) and symbolically showing the result for one case for one case in Eq. (15).
The denominator always is

$$A_2(l_1,l_2) = \int_0^{l_2}\frac{l_2-x}{1-x}dx - \int_0^{l_1}\frac{l_1-x}{1-x}dx . \qquad (9)$$

which can be given in a closed form as

$$A_2(l_1,l_2) = (1-l_2)\ln(1-l_2)-(1-l_1)\ln(1-l_1)+(l_2-l_1) . \qquad (10)$$

For $u_t < l_1$ (any other assumption would only be of theoretical nature)

$$A_{2,u}(l_1,l_2,u_t) = \int_{x_1}^{x_2}\int_{y_1}^{y_2} dxdy , \qquad (11)$$

where

$$x_1 = 0 \; ; \; x_2 = \frac{l_2-u_t}{1-u_t} \; ; \; y_1 = \max\left(u_t, \frac{l_1-x}{1-x}\right) \; ; \; y_2 = \frac{l_2-x}{1-x} .$$

This can also be given in closed form.

$$A_{2,u}(l_1,l_2,u_t) = \qquad (12)$$

$$(1-l_2)\ln\left(\frac{1-l_2}{1-u_t}\right)-(1-l_1)\ln\left(\frac{1-l_1}{1-u_t}\right)+(l_2-l_1)$$

$A_{2,d}$ and $A_{2,n}$ are given by, respectively,

$$A_{2,d}(l_1, l_2, u_t) = \int_{x_1}^{x_2}\int_{y_1}^{y_2} dxdy\,, \tag{13}$$

where

$$x_1 = u_t\,;\; x_2 = \frac{l_2 - u_t}{1 - u_t}\,;\; y_1 = \max\left(u_t, \frac{l_1 - x}{1 - x}\right)\,;\; y_2 = \frac{l_2 - x}{1 - x}$$

and

$$A_{2,n}(l_1, l_2, u_t) = \int_{x_1}^{x_2}\int_{y_1}^{y_2} dxdy\,, \tag{14}$$

where

$$x_1 = \frac{l_1 - u_t}{1 - u_t}\,;\; x_2 = u_t\,;\; y_1 = \frac{l_1 - x}{1 - x}\,;\; y_2 = \min\left(u_t, \frac{l_2 - x}{1 - x}\right)$$

Note that we define

$$\int_a^b f(x)dx = 0 \text{ for } a > b\,.$$

Due to the min and max operator, $A_{2,u}$ and $A_{2,d}$ generally can not be given in closed form and their calculation becomes intractable. For the two node case it certainly is possible, but since our goal is to analyze networks for much larger path lengths, say on the order of 10 internal nodes, the complexity is not acceptable. For the case depicted in Figure 4, applying Eqs. (10) and (14) to Eq. (8) yields

$$P(X)_2 = 1 - \frac{(1 - u_t)^2 - (1 - l_1)\left[1 + \ln\left(\frac{(1 - u_t)^2}{1 - l_1}\right)\right]}{(1 - l_2)\ln(1 - l_2) - (1 - l_1)\ln(1 - l_1) + (l_2 - l_1)}\,. \tag{15}$$

If the load state of the single nodes is not uniformly distributed, but given by a general density function $f_x(x)$ then instead of areas, volumes under the density function have to be calculated, i.e., all integrals in the form

$$A = \int_{x_1}^{x_2}\int_{y_1}^{y_2} dxdy$$

become

$$A = \int_{x_1}^{x_2}\int_{y_1}^{y_2} f_{x,y}(x, y)dxdy\,.$$

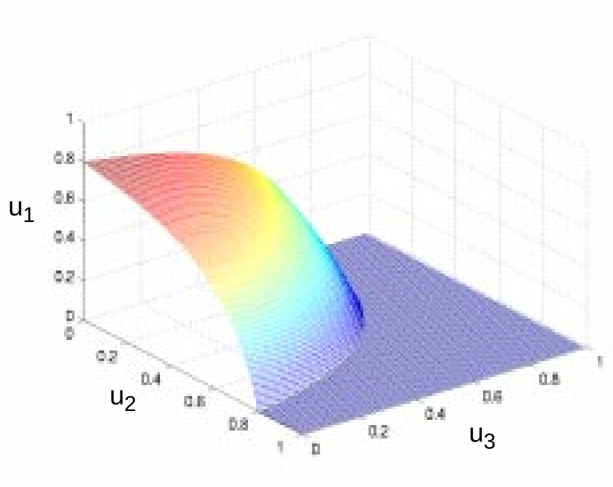

Figure 5: The 3 node case

4. The *N* Node Case

As the analysis of the two node case already indicates, the *N* node case becomes quickly intractable, even if uniform distribution of the load states is assumed. In this section we discuss the limitations of the analytical computation and then point out a heuristic. We first consider the three node case. Analogous to the two node case, the marking probability is given by

$$l = 1 - (1 - u_1)(1 - u_2)(1 - u_3)\,, \tag{16}$$

which by rearranging yields

$$u_3(l, u_1, u_2) = 1 - \frac{1 - l}{(1 - u_1)(1 - u_2)}\,.$$

A plot of Eq. (16) with $l = 0.8$ is shown in Figure 5. A_3 is then, analogous to Eq. (9)

$$A_3(l_1, l_2) = \int_{x_1}^{x_2}\int_{y_1}^{y_2}\int_{z_1}^{z_2} du_1 du_2 du_3 - \int_{x_3}^{x_4}\int_{y_3}^{y_4}\int_{z_3}^{z_4} du_1 du_2 du_3\,, \tag{17}$$

where

$$x_1 = y_1 = z_1 = x_3 = y_3 = z_3 = 0\,;$$

$$x_2 = l_2\,;\; y_2 = 1 - \frac{1 - l_2}{1 - u_1}\,;\; z_2 = 1 - \frac{1 - l_2}{(1 - u_1)(1 - u_2)}\,;$$

$$x_4 = l_1\,;\; y_4 = 1 - \frac{1 - l_1}{1 - u_1}\,;\; z_4 = 1 - \frac{1 - l_1}{(1 - u_1)(1 - u_2)}\,.$$

For the *N* node case, the marking probability is according to Eq. (1)

$$l = 1 - \prod_{j=1}^{N}(1 - u_j)\,.$$

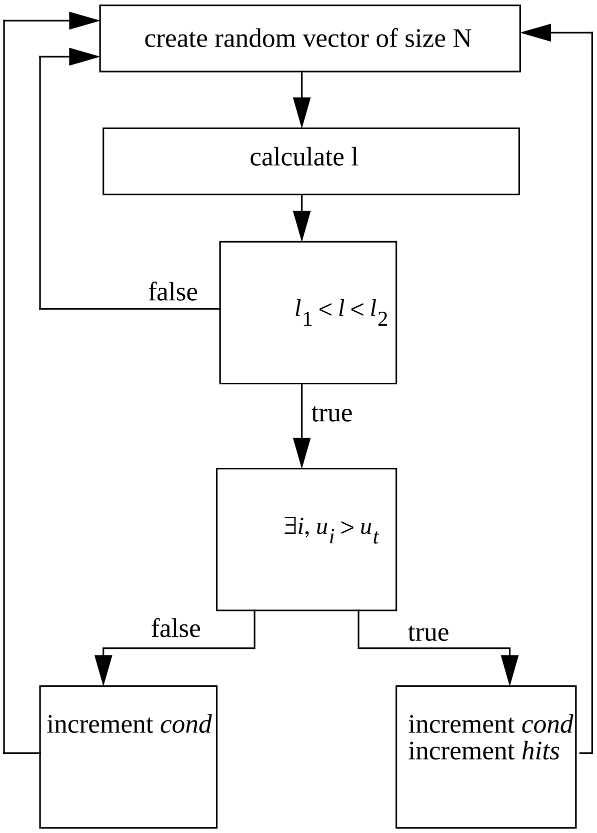

Figure 6: Monte Carlo algorithm flow chart

Solving for u_N yields

$$u_N(l, u_1, u_2, ..., u_{N-1}) = 1 - \frac{1-l}{\prod_{j=1}^{N-1}(1-u_j)} . \tag{18}$$

Following the pattern of Eq (17), we find that calculating Eq (3) for the N node case requires to solve an integral in the form

$$A_N(l) = \int_0^{g_1}\int_0^{g_2} ... \int_0^{g_N} du_1 du_2 ... du_N \quad , \tag{19}$$

where

$$g_i = 1 - \frac{1-l}{\prod_{j=1}^{i}(1-u_j)} . \tag{20}$$

To our knowledge this is analytically impossible. Arbitrary probability density functions certainly do not make it easier. Therefore, we continue with a heuristic in form of a Monte Carlo based algorithm, which has no problem of accommodating whatever kind of density function on the load states of internal nodes. The flow chart of our algorithm is shown in Figure 6.

At first, a vector of N random numbers is created. Each number represents the marking probability of a node. E.g., for the linear marking function without offset, each number is uniformly distributed between 0 and 1. Then l, i.e., the marking probability of a packet that takes a path with the loads according to the random vector, is calculated with Eq. (1). If l is outside the interval $[l_1, l_2]$ a new random vector is created. If the condition holds then it is checked whether at least one of the nodes is loaded beyond the desired threshold. In the variable *cond* the number of events in which the condition holds is counted, while the variable *hits* takes account of the number of events in which the threshold is exceeded.

Our Monte Carlo algorithm can be verified by comparing it to the analytical results of the two node case in section 3. The difference between the analytical and Monte Carlo result for the probability $P(X)$ from Eq. (3) was less than 0.02 for various settings and repetitions where each experiment was run with 5×10^5 iterations.

Of course this algorithm has a complexity which makes real-time calculations in load control gateways impossible. But they are not necessary, as it is possible to a priori compute a comprehensive look-up table. This look-up table has three dimensions: The number of nodes on the path, the interval of the measured path marking probability, as well as the desired bottleneck node load.

5. Numerical Examples

In this section, we give results obtained with the Monte Carlo based algorithm. As motivated before, we assume uniform distribution of load states throughout this section. Note that it is not necessary for the distribution of load states to be exactly uniform, but the actual distribution has to be such that it causes less conservative admission control decisions than the uniform distribution. First we use the linear marking function without offset. We considered the cases with 2 through 7 nodes and ran 5×10^5 iterations each. This should cover the space of possibilities sufficiently, which is 10^9 for 7 nodes and a discretization of 0.01. As it can be seen in the algorithm, only the number of iterations in which the condition holds, as well as the number of hits is statistically relevant. These numbers decline quickly as the number of nodes rises. E.g., when $0.89 < l < 0.91$, the condition holds 4094 times and out of those 2 times $u_t > 0.85$. Therefore, whether 5×10^5 iterations are sufficient to obtain statistically sound results is subject to future work. Nevertheless the tendencies when tuning the parameters are clearly visible.

In Figures 7 and 8 $P(X)$ vs. the number of nodes is plotted. The threshold u_t is increased in steps of 0.5. The ranges of u_t and $[l_1, l_2]$ are chosen according to the desired utilization network providers target.

Comparing these results with the original method from [14], we find a significant improvement.

E.g., when the targeted path marking probability is 0.95 (Figure 8), the probability that the bottleneck is loaded 0.85 is less than 0.1. This means that, unlike in the original method, a considerable amount of additional traffic can be admitted safely.

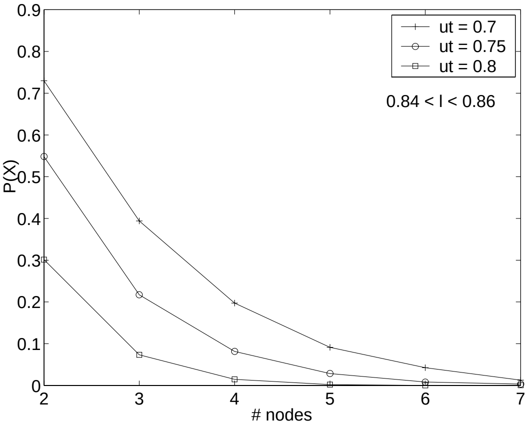

Figure 7: Bottleneck estimate - l approx. 0.8

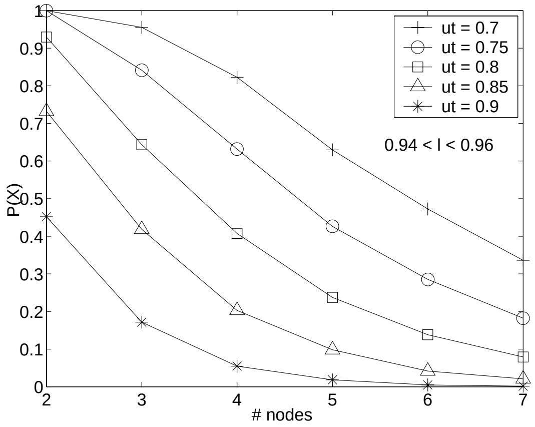

Figure 8: Bottleneck estimate - l approx. 0.95

In Figure 9, we show how *P(X)* behaves if *N* is held constant and u_t is varied. Finally, we show in Figure 10 an example with a different marking function. We use now

$$u = \frac{e^{ks} - 1}{e^{k} - 1}. \qquad (21)$$

Note that the loads of the nodes remain uniformly distributed. The constant *k* we arbitrary set to 5. Using such a marking function yields better results when the number of nodes increases and the intervals get closer to 1.

This implies that the appropriate marking function depends on the number nodes and the interval. This is subject to further research.

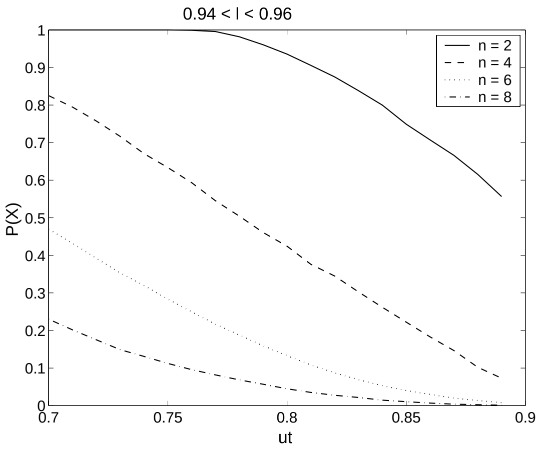

Figure 9: Bottleneck estimate - P(X) vs. ut

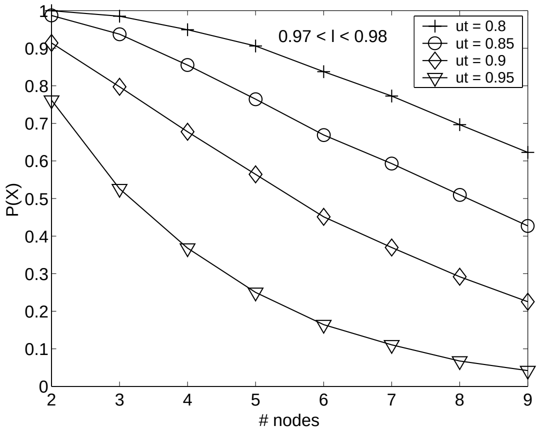

Figure 10: Bottleneck estimate - l approx. 0.975

6. Conclusion

We have devised a method to estimate the load of the bottleneck node in a domain guarded by load control gateways. Based on the knowledge of the path marking probability efficient admission control decisions for inelastic traffic streams can be achieved. We showed that the exact calculation is mathematically intractable and thus developed a Monte Carlo based algorithm to solve the problem. By numerical examples we have shown it to constitute a significant improvement of the original admission control method.

There are several pointers for future work. The assumption on the probability distribution of load states to be uniform, has to be further verified. This could be done experimentally. Of course, it may also be the case that such investigations yield other probability distribution functions. Furthermore, some statistical issues remain open, such as the appropriate size of the interval and the goodness of the Monte Carlo algorithm. Also marking functions are an interesting area of further research. An interesting observation is that the appropriate marking function seems to depend on the number of nodes and the interval.

To attempt to utilize the sequence of the marked packets, i.e., treat an 101010 sequence, where 0 and 1 denote an unmarked and marked packet, respectively, differently than a 111000 sequence, is a long shot. Last but not least, the performance of the method is to be verified in an actual implementation.

References

[1] M. Allman, V. Paxson, and W. R. Stevens. RFC 2581 - TCP Congestion Control. Standards Track RFC, April 1999.

[2] G. Bianchi, A. Capone, and C. Petrioli. Throughput Analysis of End-to-End Measurement-Based Admission Control in IP. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 1461–1470. IEEE, March 2000.

[3] L. Breslau, S. Jamin, and S. Shenker. Comments on the Performance of Measurement-Based Admission Control Algorithms. In Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000), pages 1233–1242. IEEE, March 2000.

[4] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance. *ACM Computer Communication Review*, 30(4):57–69, October 2000. Proceedings of SIGCOMM'2000 Conference.

[5] S. Crosby, I. Leslie, B. McGurk, J. Lewis, R. Russel, and F. Toomey. Statistical Properties of a Near-Optimal CAC Algorithm. In Proceedings IEEE ATM'97, pages 103–112. IEEE, June 1997.

[6] V. Elek, G. Karlsson, and R. Rönngren. Admission Control Based on End-to-End Measurements. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 623–630. IEEE, March 2000.

[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1(4):397–413, August 1993.

[8] R. Gibbens, F. Kelly, and P. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. IEEE Journal on Selected Areas in Communications, 13:1102–1114, 1995.

[9] R. Gibbens and F. Kelly. Distributed Connection Acceptance Control for a Connectionless Network. In *Proceedings of 16th International Teletraffic Congress - ITC 16, Edinburgh, Scotland*, 1999.

[10] R. Gibbens and F. Kelly. Measurement-based Connection Admission Control. In *Proceedings of the 15th International Teletraffic Congress* - ITC 15, Washington, DC, USA, pages 879–888, 1997.

[11] R. Gibbens and F. Kelly. Resource Pricing and the Evolution of Congestion Control. *Automatica*, 35:1969–1985, 1999.

[12] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70, August 1997.

[13] R. Johari and D. Tan. End-to-End Congestion Control for the Internet: Delays and Stability. *IEEE/ACM Transactions on Networking*, 9(6):818–832, December 2001.

[14] M. Karsten and J. Schmitt. Admission Control based on Packet Marking and Feedback Signalling – Mechanisms, Implementation and Experiments. Technical Report TR-KOM-2002-03, *http://www.kom.tu-darmstadt.de/publications/abstracts/KS02-5.html*, Darmstadt University of Technology, May 2002

[15] F. Kelly. Models for a self-managed Internet. In *Philosophical Transactions of the Royal Society A358*, pages 2335–2348, 2000.

[16] F. Kelly, P. Key, and S. Zachary. Distributed Admission Control. *IEEE Journal on Selected Areas in Communications*, 18(12):2617–2628, December 2000.

[17] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[18] T. Kelly. An ECN Probe-Based Connection Acceptance Control. *ACM Computer Communication Review*, 31(3):14–25, July 2001.

[19] L. Massoulie. Stability of Distributed Congestion Control with Heterogeneous Feedback Delays, 2000. Microsoft Research Technical Report, 2000-11.

[20] K. Pandit, J. Schmitt, M Karsten and R. Steinmetz. Bottleneck Estimation for Load Control Gateways - Extended Version, Technical Report TR-KOM-2003-06, *http://www.kom.tu-darmstadt.de/publications/abstracts/PSKS03-1.html*, Darmstadt University of Technology, July 2003.

[21] K. Ramakrishnan, S. Floyd, and D. Black. RFC 3168 - The Addition of Explicit Congestion Notification (ECN) to IP. Standards Track RFC, September 2001.

[22] V. Siris, C. Courcoubetis, and G. Margetis. Service differentiation in ECN networks using weighted window-based congestion control. In *Proceedings of Quality of Future Internet Services Workshop 2001, Coimbra, Portugal*, pages 190–206. Springer LNCS, September 2001

[23] S. Shenker, C. Partridge and R. Guerin. RFC 2581 - Specification of Guaranteed Quality of Service. Standards Track RFC, September 1997.

[24] L. Westberg, M. Jacobsson, G. Karagiannis, S. Oosthoek, D. Partain, V. Rexhepi, and R. Szabo. Resource Management in DiffServ Framework. Internet Draft, February 2002. Work in progress.