Packet Marking for Integrated Load Control

M. Karsten
School of Computer Science
University of Waterloo, Canada
mkarsten@uwaterloo.ca

J. Schmitt
Distributed Computer Systems Lab
University of Kaiserslautern, Germany
jschmitt@informatik.uni-kl.de

Abstract

Combining low-complexity packet marking at internal nodes with path-based load observation at edge gateways enables a rich set of control mechanisms in a packet-switched network. In this paper, we propose a method to estimate the relative per-node load at internal nodes, based on only the aggregated marking signal available at edge nodes, without any knowledge of internal capacities. We present the design of an integrated network system that supports both admission control and per-node load estimation using only two bits in the packet header, which requires a specific encoding of the two distinct marking signals. Finally, we describe a prototype implementation and a few simulation and lab experiments as a proof-of-concept for this approach.

Keywords

network control, quality of service, packet marking, load estimation, routing

1. Introduction

The idea of exploiting ECN-like [1] packet marking at Internet routers for flow admission control at edge or end systems has been proposed as an effective way of stabilizing the resource allocation along each transmission path, without the need for any explicit interaction with intermediate nodes. Such an admission control scheme reduces computation overhead at internal nodes, compared to per-flow interaction, and provides structural independence between internal and edge modules, which only need to be agree on the aggregated marking signal. In particular, an admission control decision can often be made without knowing the absolute capacity of internal resources.

In this paper, we explore the same principles to perform per-node load estimation at the edge of a network domain, without explicit involvement of internal nodes, other than through binary packet marking. Per-node load information can be used for constraint-based routing, such as load balancing or traffic engineering. We present an integrated system design that allows to utilize packet marking for both admission control and per-node load estimation. Admission control and load estimation are both possible without explicit knowledge of the forwarding capacity of internal nodes, which makes the approach suitable for highly modular systems and also for systems where internal capacities are not known at the network edge.

The paper is organized as follows: In the next section, related work is surveyed. In Section 3, the system design is presented and discussed. Section 4 contains a brief

description of the software prototype, while results of the experimental evaluation are given in Section 5. The paper is concluded by a discussion and an outlook in Section 6.

2. Related Work

Traditional network control systems in telecommunication networks are highly distributed and essentially operate on each multiplexer node. One early example in the context of the Internet is given by the Integrated Services architecture [2]. However, given the flexibility/complexity of packet switching and the transmission speed of fibre optics, it turns out that node resources (memory, CPU, bus) are often the limiting factor, rather than link capacity. Control systems using a centralized resource broker such as [3,4] and a multitude of other proposals eliminate the processing at internal nodes, but at the expense of reliance on a central element for monitoring traffic characteristics per path at egress nodes and view the network as a black box. All measurement-based admission control schemes analyse the statistical nature of traffic and load measurements and propose suitable decision algorithms. In [5], an extensive comparison of measurement-based schemes finds that all perform fairly similar.

If internal nodes and edge gateways cooperate, for example through packet marking, very high utilization is possible. Pioneering work in this direction, has been done by Kelly et al. [6, 7]. Their analytical results show the basic stability of distributed admission control based on marking at resources even in the case of feedback delays [8]. Building on these results, there is work to shed light on the influence of delayed system reaction on stability, which presents bounds for the reaction delay [9,10]. In [11], a model for an Internet exclusively managed by end systems is presented and thoroughly analysed with respect to stability. In [12], a similar system design is presented, but with an admission control gateway carrying out probing for end systems. A simulation-based comparison of the basic design options for endpoint admission control with probing is presented in [13]. In particular, [13] reports a probing duration in the order of several seconds, whereas [14] argues for much lower values for the initial probing phase. A general framework for aggregated signalling and admission control is presented in [15]. One part of this proposal, termed *MBAC group*, introduces abstract signalling elements for edge-based admission control. Another part of that work [16] discusses different types of marking functions, termed *proportional* and *greedy*. The marking functions used in our work are examples of these basic types. Such a marking and admission control infrastructure can easily be used for dynamic pricing or even to facilitate resource auctions [17].

Routing in the current Internet primarily focuses on connectivity and thus mostly defaults to shortest path routing algorithms like in *Open Shortest Path First* (OSPF) [18]. However, the efficient use of network infrastructure is increasingly becoming more important to network providers to gain a competitive advantage. Therefore, many extensions to existing routing protocols, for example [19, 20], as well as totally new proposals – QoS routing schemes such as [21, 22, 23, 24] and load balancing routing schemes as proposed in [25, 26] – have been put forward. These efforts have been associated with the keyword traffic engineering, which is generally perceived as

mechanisms and strategies to optimize the use of an existing network infrastructure [27]. All these approaches depend in some form or another on information about the traffic matrix specifying traffic between ingress and egress routers as well as current load information of internal nodes. In fact, measuring and disseminating accurate load information is a critical component in the overall routing process [28].

3. System Design

The network control system presented here operates per network domain. Internal nodes use binary packet marking to convey load information to edge gateways, which in turn use this information for admission control and load estimation. The system interacts with adjacent network domains or clients via an inter-domain request signalling protocol. A conceptual overview of the system is shown in Figure 1. The system performs both admission control and per-node load estimation, but in different ways. Admission control is a pivotal element to ensure service guarantees, for example priority service during emergencies. The distributed admission control function is therefore implemented along the traffic path. Load-based routing on the other hand aims at improving resource utilisation, but is not essential. Consequently, load estimation can be a centralized service, despite the resulting single point of failure.

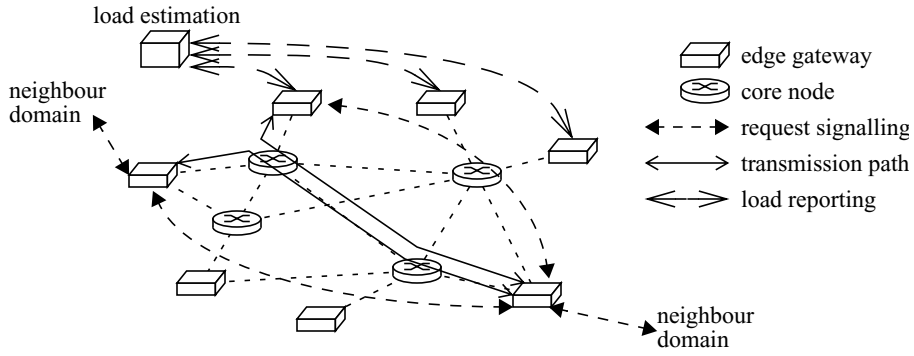

Figure 1: Network Domain Overview

3.1. Packet Marking and Load Observation

The focus of this work is on long-lived traffic with limited elasticity, comprised of individual large application flows or aggregates of smaller flows. In this context, there are two types of marking functions:

- A greedy marking algorithm marks all packets, if the local load is above a certain threshold and otherwise does not mark any packet. For long-lived and inelastic traffic, *Virtual Queue Marking* (VQM) [12] and its derivatives essentially behave like a simple threshold-based marking algorithm with a short time lag.
- A proportional marking algorithm randomly marks a fraction of packets in proportion to the local load. An example of such an algorithm is *Load-based Marking* (LBM) [29].

If marking at each resource is modelled as a Bernoulli experiment with distribution function $m(.)$ and l_k is the local load at resource k, then for path p the path marking rate $M(p)$, as observed by edge gateways, can be expressed as

$$M(p) = 1 - \prod_{k \in p} (1 - m(l_k)), \tag{1}$$

which immediately leads to the following observations:

- With greedy marking, the path marking rate is essentially a binary signal denoting whether at least one node along the path is loaded above its local load threshold.
- Multi-node proportional marking emits a monotonic and continuous path marking signal, which however may significantly deviate from individual marking rates.

3.2. Admission Control

The admission control part of the system is integrated with a request signalling protocol – in this case RSVP [30]. There are two interesting aspects about this. First, RSVP automatically supports signalling across RSVP-unaware nodes, therefore no changes are necessary to facilitate edge-to-edge signalling. Second, marking-based network control requires feedback from the load observation point back to the traffic regulation entity. This fits with the receiver-based reservation style of RSVP, such that load information can be piggybacked onto the reservation message, which is sent from the egress gateway to the ingress anyway. With a sender-based reservation protocol, the ingress gateway would need to explicitly query the egress for the current path load before continuing regular signalling. Edge gateways continuously monitor the path load, such that usually no probing phase is needed during connection establishment.

Admission control is a binary decision about the acceptance of a new service request. The goal is to determine whether any internal resource is at the limits of its capacity. Following the argument of [7], only the highest loaded resource is relevant for both utilization and fairness. Therefore, a marking signal for admission control should encode the load situation of the highest loaded resource. This can be accomplished by greedy marking, but not by proportional marking.

3.3. Per-Node Load Estimation

The goal of per-node load estimation is to indirectly calculate the local load at internal resources purely based on observations made at the network edges. If the capacity of internal resources is known and further, if it is known which paths traverse which resource, the load situation of each internal resource can be estimated directly based on path usage rates. Formally expressed, let $t_{i,j}$ be the measured usage rate for path $p_{i,j}$, P_k be the set of paths which traverse resource k. Based on the absolute forwarding capacity for resource k, c_k, its relative load l_k can be calculated as

$$l_k = \left(\sum_{p_{i,j} \in P_k} t_{i,j} \right) / c_k . \tag{2}$$

However, the absolute capacity of internal resources may in some cases be unknown or unsuitable for this simple approach, for example:

- capacity adaptation of service classes by an independent allocation system;
- wireless or overlay links with varying capacity;
- complex notion of load, for example a combination of processing and link load;
- heterogeneous notion of load at different nodes.

In such a case, the aggregated marking rates observed at edge gateways may still be used to approximate the relative one-dimensional local load of each internal resource. The mathematical foundations of this method are based on the fact that the path marking rate can be expressed as a function of the internal resources' marking rates. Thereby, the set of path marking rates from all edge gateways allows to set up a system of equations, the solution of which estimates the individual loads.

Basic Load Estimation

To formulate the model, let K be the set of all resources in the network and $m_k(l_k)$ the relative marking rate at resource $k \in K$ using the marking function $m_k(.)$ for a given relative load with m_k: $[0,1) \rightarrow [0,1)$. A path between a pair of edge gateways i and j is denoted as $p_{i,j} \subseteq K$. The set of equations given by the path marking probabilities as in (1) can then be transformed into an equivalent set of *linear* equations for all paths $p_{i,j}$

$$\ln(1 - M(p_{i,j})) = \sum_{k \in p_{i,j}} y_k \quad \text{with} \quad y_k = \ln(1 - m_k(l_k)), \tag{3}$$

which can be solved, if and only if the number of paths $p_{i,j}$ results in $n = |K|$ linear independent equations. Solving the linear system of equations (3) and assuming the marking function m_k to be invertible, the relative load l_k at each resource is obtained by

$$l_k = m_k^{-1}(1 - e^{y_k}). \tag{4}$$

Note that this procedure does not require all $m_k(.)$ to be uniform for all nodes. Nevertheless, based on the individual relative load for each resource, a one-dimensional representation of its capacity can be determined. Let $t_{i,j}$ be the measured usage rate for path $p_{i,j}$ and P_k be the set of paths which traverse resource k. Based on the relative load l_k, its capacity c_k can be calculated as a re-formulation of (2) as

$$c_k = \left(\sum_{p_{i,j} \in P_k} t_{i,j} \right) / l_k . \tag{5}$$

A necessary condition for the above calculations is that the marking function m must be invertible, which essentially requires it to be strictly monotonic and continuous. Furthermore, if all packets are marked along a path, the equation system cannot be solved anymore, since the logarithm of 0 is undefined. Consequently, a proportional marking function such as LBM is suitable for load and capacity estimation, while a

greedy marking function is not. Further, all such load estimation techniques requires global routing information for the network domain, therefore a central server is the easiest way to implement such a system, although a distributed solution is possible.

Since the real system is asynchronous in nature, the above procedure can only be used to calculate estimations of the real values. In general, any realistic network topology and routing policies will result in enough linearly independent equations (3), since otherwise there would be a resource that does not multiplex at all. In fact, the system of equations is likely to be over-specified and may not be solvable due to contradicting values caused by measurement inaccuracies and communication delays. A practical resort is to use only the last path measurements that yield enough linearly independent equations. The mathematical complexity of solving the linear system of equations is limited, since it can be represented as a binary matrix.

Hybrid Load Estimation

Most of the per-path information is somewhat stale at the time when the relative load is estimated. The set of linear equations (3) is generated by a logarithmic transformation of (1). Because of the multiplication in (1), it is likely that errors propagate and increase throughout the system of equations, which results in increased sensitivity to any kind of information inaccuracy. On the other hand, the aggregated usage rate per resource as in (2) is also available. Since calculating this value only involves addition, it is less sensitive to inaccuracies. Without knowing the resource capacity, however, only the variation of the usage rate can be used to incrementally adjust the load estimation per node. It seems very promising to combine the properties of both procedures to improve the quality of the load estimation. Specifically, for each resource k, let l'_k and u'_k denote the previously calculated relative load and usage while l_k and u_k denote the current raw values. Then, the new load estimation l_k^* is calculated as

$$l_k^* = \left(\alpha \cdot l'_k \frac{u_k}{u'_k} + l_k\right) / (\alpha + 1) \tag{6}$$

with α being the weighting factor between the influence of the usage rate and the raw load resulting from the basic load estimation procedure. Assuming that the incremental adaptation based on the usage rate is more robust against information inaccuracies, the weighting factor α represents a trade-off between the overall quality and the convergence speed of the load estimation. Convergence speed matters in case of capacity changes, because only the raw load estimation part is capable to detect them.

3.4. Integration

As discussed in the previous sections, admission control does not work well with proportional marking at multiple resources and instead requires a greedy marking function. On the other hand, per-node load estimation only works with a continuous and monotonic load signal, such as the one generated by LBM. Given the current IP header, it would be beneficial to encode both marking signals into the two ECN bits, while

still being able to discriminate between foreground and background traffic without using additional bits. Then, the load control system is fully independent and transparent to the notion of service classes. It supports the notion of foreground traffic that is covered by service agreements and background traffic, which is transmitted on a best effort basis within the respective service class. Incoming traffic passes through traffic regulation at each ingress gateway and only packets complying to an existing service contract are marked as foreground traffic. Background traffic is discriminated against by applying a smaller queue drop threshold at internal nodes, which is independent of any scheduling algorithm between service classes.

The real-world challenge is to encode three units of binary information (foreground, greedy, and proportional marking) with two bits. The resulting four code points are used similar to the ECN code points [1]. For background traffic, both bits are cleared and no marking takes place at internal nodes. Packets belonging to foreground traffic are marked by edge gateways by setting one of the two bits. The two alternatives are termed *AC* and *TE* packets. In our system, the ingress gateway strictly alternates between both code points per peer gateway and thus produces an equal number of AC and TE packets for each path through the network. At core nodes, packet marking amounts to setting the respective other bit. AC packets are marked by greedy marking, if necessary, while TE packets are marked with the proportional algorithm. We call the resulting packets AC_m and TE_m packets. Because $\mathrm{AC_m}$ and $\mathrm{TE_m}$ packets are indistinguishable, an egress gateway cannot directly observe the respective relative path marking rate. However, since ingress nodes strictly alternate between AC and TE, an egress gateway can assume to always have an equal number of packets originally marked as AC and TE in its observation buffer. Given a total number of marked packets c, as well as some unmodified AC and TE packets, e_{AC} and e_{TE} respectively, the number of applicable $\mathrm{AC_m}$ packets m_{AC} can be calculated indirectly as

$$m_{AC} = \frac{c + e_{AC} + e_{TE}}{2} - e_{AC} \tag{7}$$

and the number of $\mathrm{TE_m}$ packets can be calculated accordingly. Thereby, it is possible to decode both load signals from the packet stream.

4. Software Prototype

We have developed a software prototype implementing all operations presented above. The software is built in the framework of a publicly available RSVP implementation [31]. It uses and extends the alternate queuing (ALTQ) framework for FreeBSD [32] and has been ported to the ns-2 simulation environment [33]. We have chosen an integrated software development approach by porting the RSVP daemon code to the ns-2 environment. Also, all algorithmic code for traffic regulation, packet marking and load observation is kept general enough to be used both in the ALTQ kernel modules for prototype experiments and as part of the simulation. The operation of the simulated

technology can be compared to and calibrated by results from lab experiments, which should improve the validity of subsequent simulation results. The original software structure of the RSVP implementation alleviates its porting to the simulation environment by strictly separating system-dependent from system-independent code.

The per-node load estimation module is implemented according to the model presented in Section 3.3, but only in the simulation environment. To better approximate real-world conditions, periodic load reports to the central server are generated with a random update period. Initial experiments have shown that for the load-based estimation procedure, the fluctuation of results caused by stale information is also influenced by the length of the shortest path a node is traversed by. The longer this shortest path is the higher are the resulting fluctuations. As a workaround, the results of the load-based estimation procedure are subject to exponential smoothing according to:

$$load_{est} = (load_{old} * (a-1) + load_{new}) / a$$

with a being the length of the respective shortest path for the node. This smoothing step is used for both load-based and hybrid estimation.

5. Evaluation

We present a number of initial simulation and experimental results to demonstrate the validity of the presented approach for admission control and load estimation. The actual experiments reported here just illustrate the typical behaviour of the system. Although many experiments have been carried out with the software prototype, we have not yet collected enough data for a thorough statistical analysis.

5.1. Admission Control

The first series of experiments are carried out in a standard dumbbell topology of PentiumIII/450MHz PCs running FreeBSD, which are connected by dedicated Ethernet links at 10 Mb/s. Internal nodes use LBM and VQM to mark packets (only VQM marks are used for admission control) and the virtual queue is configured at 85% of the link speed. The system is loaded (and overloaded) with a number of VoIP-like ses-

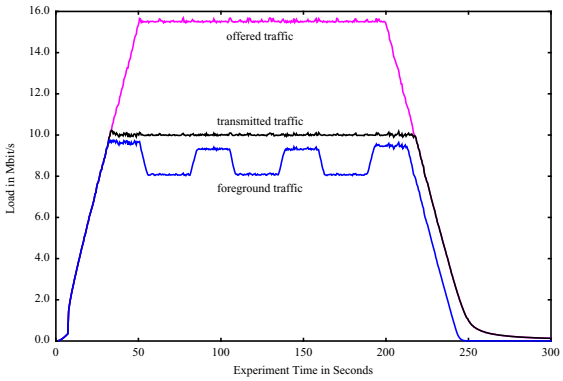

Figure 2: Deterministic Session Arrival and Duration

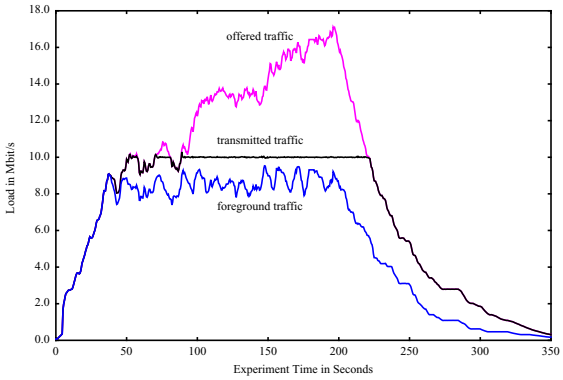

Figure 3: Exponential Session Arrival and Duration

sions, of which only a subset can be accepted by the admission control. The rest of the sessions therefore emit background traffic. Figure 2 shows the behaviour of the system with a deterministic session arrival process. The periodicity of accepted traffic load is caused by the deterministic session arrival process in combination with the reaction delay of the system. There is no packet loss for accepted sessions (not shown in the figure). It can be concluded that the system correctly accepts and rejects service requests and effectively discriminates between foreground and background traffic. At the same time, very high resource utilization is possible. To further confirm this conclusion, Figure 3 shows the system behaviour when the inter-arrival and duration times of sessions are not deterministic, but exponentially distributed. The same system behaviour is observed in simulations.

5.2. Per-Node Load Estimation

The topology for these simulation experiments is slightly more elaborated with multiple edge gateways and cross traffic at internal links. The virtual queuing system at all internal nodes is configured at 50% to limit the overall load and create more load fluctuations. The system is again loaded with exponentially distributed VoIP-like sessions, but only foreground traffic is taken into account for per-node load estimation. The behaviour of the system is shown in Figure 4 for usage-based estimation, Figure 5 for load-based estimation, and Figure 6 for hybrid estimation.The figures show the estimated load at an internal node, as calculated by the different estimation procedures, and the actual load observed locally at the node. The average update period for information distribution is set to 2 seconds. Relative to the link capacity, the average error of usage-based load estimation is 1.23%, which increases to 2.99% for basic load estimation and can be reduced to 1.76% by using hybrid load estimation. This confirms that hybrid estimation achieves a better approximation than basic estimation and almost the same quality as usage-based estimation, albeit without the need to know the internal links' forwarding capacities. In reality, the timescale of routing operations is

likely be much coarser than that of load estimation, therefore Figure 7 shows the same data on an larger time-scale to illustrate the accuracy of hybrid estimation.

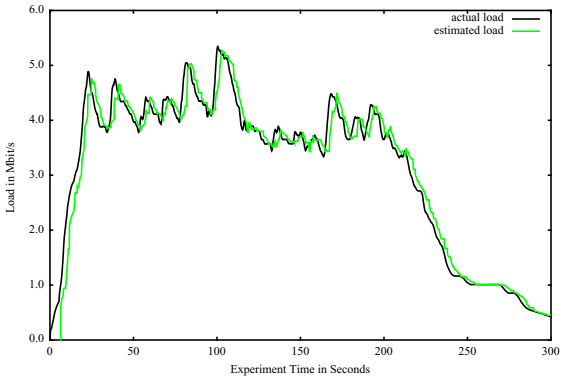

Figure 4: Load Estimation. - Usage-based (0.1s timescale)

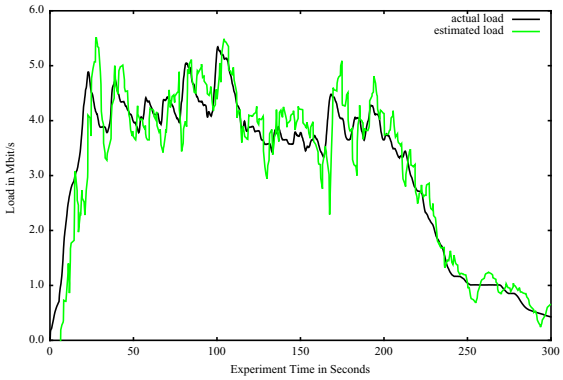

Figure 5: Load Estimation - Basic (0.1s timescale)

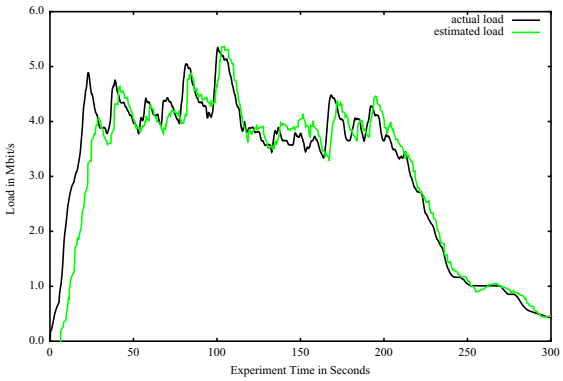

Figure 6: Load Estimation - Hybrid, $\alpha = 8$ (0.1s timescale)

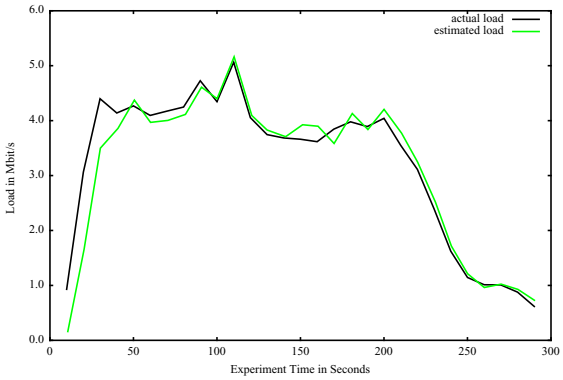

Figure 7: Load Estimation - Hybrid, $\alpha = 8$ (10s timescale)

5.3. Discussion

The traffic mix in our experiments is rather simple and not necessarily representative for all types of Internet traffic. Nevertheless, we believe that there is virtue in these experiments. First, in order to fully understand the system, it is necessary to limit the complexity of experiments. Without a basic understanding, it may not be possible to interpret experimental results from more complex scenarios. For example, the admission control test with deterministic session arrival shown in Figure 2 clearly exposes the system's reaction delay as the slope of the periodic increase and decrease of accepted foreground traffic. Second, the control system presented here may exist in multiple instances for multiple service classes. One of those service classes is likely to be a telephony class, in which case the experiment scenario is directly applicable.

6. Conclusions and Future Work

This paper presents the design and implementation of a network control system that employs low-complexity packet marking schemes at internal nodes, borrowed from active queue management research. An appropriate encoding of two distinct load signals in the available ECN bits allows to concurrently carry out admission control and generate input for per-node load estimation at edge gateways, without otherwise involving internal nodes. It is shown mathematically how per-node load approximations can be calculated without knowing the capacity of internal resources. A software prototype is presented and initial experimental results are presented. Clearly, the work presented here is not the final proof about the superiority of this approach over the many others. Rather than presenting a presumed perfect system, this work is intended as an experimental proof of concept for the basic ideas about reactive admission control and edge-based per-node load estimation. We have deliberately not proposed a specific routing system, since we believe that this is complementary to our work. Clearly, the integration with sophisticated routing systems and combined evaluations are interesting items for future work.

Independent of the actual routing being used, an in-depth analysis and more experimental evaluation is necessary for the admission control and load estimation parts of the system. The main benefits of the presented approach are its simplicity and modularity. For both claims, it is necessary to carry out in-depth experiments, preferably in very realistic networks, to assess whether the claims hold in reality.

Last not least, it is also necessary to devise a strategy and potentially detailed mechanisms to interact with the traditional use of ECN bits. The simplest solution is to confine both types of usage of those bits to different service classes. It is very likely that data-oriented TCP traffic requires different quality characteristics than, for example, interactive voice traffic, anyway. It is also worth noting that the system does not rely on the actual ECN bits, but can use any two bits in the packet header. However, the possible interaction of edge-based network control with end-system flow control remains interesting.

ACKNOWLEDGMENT

Frank Zdarsky has implemented the initial port to the ns-2 environment.

References

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

[2] R. Braden, D. Clark, and S. Shenker. RFC 1633 - Integrated Services in the Internet Architecture: An Overview, June 1994.

[3] K. Nichols, V. Jacobson, and L. Zhang. RFC 2638 - A Two-bit Differentiated Services Architecture for the Internet, July 1999.

[4] Z.-L. Zhang, Z. Duan, L. Gao, and Y. T. Hou. Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services. *ACM Computer Communication Review*, 30(4):71–83, October 2000. Proceedings of SIGCOMM 2000.

[5] 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.

[6] 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.

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

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

[9] 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.

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

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

[12] 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.

[13] 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.

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

[15] L. Westberg, A. Csaszar, G. Karagiannis, A. Marquetant, D. Partain, O. Pop, V. Rexhepi, R. Szabo, and A. Takacs. Resource Management in Diffserv (RMD): A Functionality and Performance Behavior Overview. In *Protocols for High Speed Networks: 7th IFIP/IEEE International Workshop, PfHSN 2002*, pages 17–34. Springer LNCS 2334, April 2002.

[16] A. Csaszar, A. Takacs, R. Szabo, V. Rexhepi, and G. Karagiannis. Severe Congestion Handling with Resource Management in Diffserv on Demand. In *Networking 2002*, pages 443–454. Springer LNCS 2345, May 2002.

[17] M. Karsten and J. Schmitt. Market-Based Resource Allocation for Packet-Switched Networks. In *Proceedings of the 10th International Conference on Telecommunication Systems Modelling and Analysis (ICTSM10), Monterey, USA*, pages 52–62, October 2002.

[18] J. Moy. RFC 2328 - Open Shortest Path First (OSPF) Routing, April 1998.

[19] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, and T. Przygienda. RFC 2626 - QoS Routing Mechanisms and OSPF Extensions, August 1999.

[20] C. Villamizar. OSPF Optimized Multipath (OSPF-OMP. Internet Draft, February 1999. Work in progress.

[21] Z. Wang and J. Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, September 1996.

[22] Q. Ma and P. Steenkiste. On Path Selection for Traffic with Bandwidth Guarantees. In *Proceedings 5th International Conference on Network Protocols (ICNP'97)*, pages 191–202. IEEE, October 1997.

[23] S. Chen and K. Nahrstedt. An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions. *IEEE Network Magazine*, 12(6):64–79, November 1998.

[24] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Quality of Service Routing: A Performance Perspective. *ACM Computer Communication Review*, 28(4):17–28, October 1998. Proceedings of SIGCOMM 98.

[25] N. Taft-Plotkin, B. Bellur, and R. Ogier. Quality-of-Service Routing Using Maximally Disjoint Paths. In *Proceedings International Workshop on Quality of Service (IWQoS'99)*, pages 119–128. IEEE, May 1999.

[26] S. Bak, A. Cheng, J. Cobb, and E. Leiss. Load-balanced Routing and Scheduling for Real-time Traffic in Packet-Switched Networks. In *Proceedings of IEEE Conference on Local Computer Networks (LCN 2000)*, pages 634–643. IEEE, November 2000.

[27] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. RFC 3272 - Overview and Principles of Internet Traffic Engineering, May 2002.

[28] R. Guerin and A. Orda. QoS Routing in Networks with Inaccurate Information: Theory and Algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350–364, June 1999.

[29] 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 2156, September 2001.

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

[31] M. Karsten, J. Schmitt, and R. Steinmetz. Implementation and Evaluation of the KOM RSVP Engine. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, pages 1290–1299. IEEE, April 2001.

[32] K. Cho. *The Design and Implementation of the ALTQ Traffic Management System*. PhD thesis, Keio University, Japan, January 2001. Software at http://www.csl.sony.co.jp/person/kjc/programs.html.

[33] K. Fall and K. Varadhan, editors. *The ns Manual*. April 2002. Software and Documentation available at http://www.isi.edu/nsnam/ns/.

Biography

Martin Karsten has received his diploma in Joint Computer Science and Business Economics from the University of Mannheim, Germany, in 1996. Afterwards, he has been a research scientist amd PhD candidate at Darmstadt University of Technology, Germany, and received his doctoral degree in Computer Science in 2000. After spending two more years as research group head and lecturer in Darmstadt, he took up an appointment with the University of Waterloo, Canada, where he is currently a faculty member in the School of Computer Science.

Jens Schmitt has received his diploma in Joint Computer Science and Business Economics from the University of Mannheim, Germany, in 1996. Afterwards, he has been a research scientist amd PhD candidate at Darmstadt University of Technology, Germany, and received his doctoral degree in Computer Science in 2000. After spending three more years as research group head and lecturer in Darmstadt, he took up an appointment with the University of Kaiserslautern, Germany, where he is currently a professor in the Computer Science Department.