As appeared in:Proceedings of the 1999 International Conference on Parallel Processing
(ICPP -'99), University of Aiku, Japan, Sept. 1999.

New Delay Analysis in High Speed Networks

Chengzhi Li Riccardo Bettati Wei Zhao

Department of Computer Science
Texas A & M University
College Station, TX 77843-3112
Phone 409 - 845 - 5098
Email: {chengzhi,bettati,zhao}@cs.tamu.edu

Abstract

The implementation of bounded-delay services over integrated services networks relies admission control mechanisms that in turn use end-to-end delay computation algorithms. For guaranteed-rate scheduling algorithms, such as fair queueing, delay computation based on Cruz' service curve model performs very well. Many currently deployed networks, be they packet-switched or ATM based, rely on non-guaranteed-rate disciplines, most prominently FIFO and static-priority disciplines. We show that for this class of disciplines the service curve model performs poorly. We propose the Integrated Approach as alternative to the service curve model to cluster servers for delay computaton purposes, and show in a series of evaluations that this new approach outperforms approaches based on the service curve model as well as other currently used approaches.

1. Introduction

A major challenge in the design of high-speed integrated services networks is the implementation of a *bounded-delay* service, that is, a communication service with deterministically bounded delays for all packets in a connection. In such a network, the number of connections with a bounded-delay service requirement that can be supported is mostly determined by (i) traffic characterizations used to describe the traffic of connections, (ii) the packet scheduling disciplines at each server or switch in the network, and (iii) the accuracy of the delay analysis used for connection admission control tests.

In order to guarantee that all the connections can meet their deadline requirements, an effective and efficient method to derive the upper bound for the end-to-end delay experienced by connection's traffic is needed. By a delay analysis method being *effective*, we mean that the method is able to produce delay bounds that are relatively tight. A method that overestimates the delay bounds reduces the utilization of the network. By *efficient*, we mean that the method is simple and fast in order to be used as part of online connection admission control. During the past decade, a number of service scheduling disciplines that aim to provide per-connection performance guarantees have been proposed in the context of high speed packet switching networks, such as Fair Queueing, Virtual Clock, Self Clocked Fair Queueing, Stop and Go Queueing, Earliest Deadline First, Static-Priority Scheduling, Rate Controlled Service Discipline, and SCED Scheduling. Along with these service scheduling disciplines, various delay analysis techniques have been devised to evaluate upper bounds for end-to-end delays experienced by connections in a network.

We can group these delay analysis techniques into two classes, depending on whether they decompose the network into isolated servers that are analyzed separately, or whether they integrate individual servers in the network into larger superservers. We distinguish therefore *decomposition-based* from *service-curve based* methods. A brief description of these methods is presented as following.

1.1 Decomposition-Based Methods

The basic idea for any decomposition-based method is to partition the network into isolated servers, and base the end-to-end delay analysis on the local delay analysis on the isolated servers. First, the local traffic is characterized on a per-connection basis at each server inside the network. The traffic is dependent on the source traffic for the connection and on the delay experienced by the traffic at previous servers. Next, the local delay bounds are independently computed. Finally, the upper bound for the end-to-end delay of the connection is computed as the sum of the local delay bounds at the individual servers on the path of the connection. The fundamental approach was proposed in [8, 9] and has been widely adopted in various forms.

Decomposition-based methods are very simple to use

and are suitable for networks with arbitrary topology. On the other hand, they often overestimate the end-to-end delay suffered by the connection's traffic and so reduce the network resource utilization. This is because this approach assumes that a packet suffers the worst-case delay at every server along its path. This assumption is conservative; while a packet may suffer the worst case delay at one server, it may not incur the worst case delay at a successive server. It follows that some real time connections may be rejected by a decomposition-based admission control algorithm even though the network can guarantee their QoS requirements.

1.2 Service-Curve Based Methods

The basic idea in service-curve based methods is to find a representation of a sequence of servers on the path of the connection as a single server. Successive servers are therefore integrated and dependencies between delays on successive servers can be taken into account. Servers are represented by their *service curve* $s_{i,k}(t)$, which defines the minimum amount of service (in bits transferred) that a server k can give to a particular connection i during time interval $[0, t)$ [10, 5].

Cruz [10] describes how the service curve can be used to effectively evaluate the end-to-end delay suffered by a connection. Suppose that Connection i passes through m servers and the k-th server offers the connection a service curve $s_{i,k}(t)$. Furthermore, suppose that the amount of traffic entering the network on Connection i during time interval $[0, t)$ is bounded by $F_i(t)$. Then the end-to-end delay of Connection i is bounded by

$$D_i = \max_{t \geq 0} \{S_i^{-1}(t) - F_i^{-1}(t)\}, \quad (1)$$

where $S_i(t)$ is called as *network service curve* of Connection i and is defined as

$$S_i(t) = \min\{\sum_{k=1}^{m} s_{i,k}(t_k) \parallel t_k \geq 0, \sum_{k=1}^{m} t_k = t\}. \quad (2)$$

Service curves can be used in two ways for delay computation, depending on whether scheduling algorithms are derived from pre-defined service curves, or whether service curves are derived from pre-defined scheduling algorithms.

Allocated Service Curve Method First, service curves are assigned to every connection at each server. Then, the end-to-end delay bound is derived based on the source traffic characterization and *network service curve*, which can be computed from the service curves of all servers on the path of the connection. The scheduling disciplines on the servers can be *synthesized* in a separate step from the service curves that were assigned earlier. See [10, 32] for some examples.

Theoretically this method fully utilizes the network resource and can be applied to networks with arbitary topology. However, the scheduling discipline synthesized from the service curves always relies on a dynamic priority assignment. Therefore, the scheduling overhead is not negligible, and will impair utilization of the network resource.

Induced Service Curve Method As opposed to the allocated service curve method, here servers are assigned scheduling disciplines first. Then, service curves are derived for each server based on the local server scheduling discipline. Next, the network service curve is derived based on these service curves. Finally, the end-to-end delay bound is derived based on the source traffic characterization and the network service curve [29].

Once the service curve is known for the scheduling disciplines in the system, delay analysis is straightforward. Unfortunately, except for guaranteed-rate scheduling algorithms [18], deriving service curves is very difficult, if not impossible. This is indeed the case for static-priority (SP) schedulers, simple earliest-deadline-first (EDF) schedulers, and first-in-first-out (FIFO) schedulers. In this paper we will derive an approximation for the service curve of a FIFO server and use it to compare the performance of a service-curve based approach with the integrated approach presented in this paper.

1.3 Integrating Servers

As noted above, both general approaches to end-to-end computation do not work well for non-guaranteed-rate scheduling disciplines. Decomposition-based approaches over-estimate end-to-end delays for all disciplines by not taking into consideration self-regulating effects as traffic traverses the network on common paths. Service-curve based approaches work fine for guaranteed-rate disciplines, but fail for other disciplines. Indeed, we will illustrate later with an example of a chain of FIFO servers (Section 4) that service-curve based approaches can perform substantially worse than decomposition based ones.

In this paper we propose an *integrated* approach to analyze networks of non-guaranteed-rate servers. The general approach is to determine an accurate integrated service description for a collection of servers. Similar to the network service curve described earlier, that allows for a computation of output traffic descriptors for connections leaving the collection of servers under consideration. End-to-end delays can then be computed by partitioning the network into collections of servers, and then applying a decomposition-based method collections of servers instead of individual servers, thus greatly reducing the amount of over-estimation occuring in the delay computation.

We will describe the new delay analyis method in Section 2 on for a simple subnetwork containing two servers. While the approach itself is generic for a large class of service disciplines, we will focus our attention to systems with FIFO servers. In Section 3 we will apply the results of Section 2 to define an algorithm for end-to-end delay computation. We provide a detailed evaluation of the new algorithm by comparing it with a decomposition-based and a service-curve based algorithm. Section 5 concludes the paper.

2 Integrated Delay Analysis for a Subsystem with Two Multiplexors

In this section, we study a subsystem with two multiplexors, the topology for this subsystem is illustrated in Figure 1. An integrated method for the delay analysis in this system is presented. Although the approach is generic in nature, we will assume that the multiplexors are use a FIFO scheduling policy.

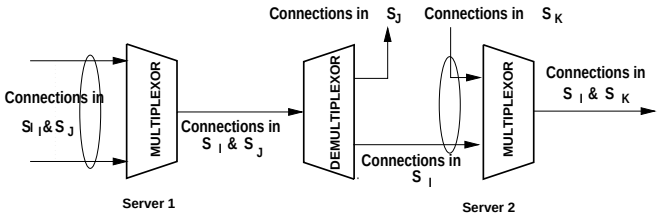

Figure 1. A Subsystem with two Multiplexors.

To evaluate the worst-case delay suffered by traffic, the description for network traffic is needed. We give the following definitions and notations for this purpose.

Definition 1. The *traffic arrival function* $f_{i,j}(t)$ of Connection i at Server j is defined as the amount of data arriving at Server j from Connection i during the time interval $[0, t)$.

Definition 2. We call function $b_{i,j}(I)$ the *traffic constraint function* of $f_{i,j}(t)$ if for any $t > 0$ and $I > 0$

$$f_{i,j}(t + I) - f_{i,j}(t) \leq b_{i,j}(I). \tag{3}$$

Similarly, we define the amount of traffic leaving the server as follows:

Definition 3. The amount of traffic leaving the Server j during the interval $[0, t)$ is denoted by $W_j(t)$. We call $W_j(t)$ the output traffic function at Server j.

Referring to the two-server subsystem depicted in Figure 1, we use $\mathcal{S}_{12}$ to denote the set of all connections that traverse both Server 1 and Server 2. We use $\mathcal{S}_1$ to denote the set of all connections that traverse Server 1 only and then leave the subsystem. We use $\mathcal{S}_2$ to denote the set of all connections that join the subsystem after Server 1 and traverse Server 2 only.

Throughout this paper, we will assume that the traffic of every connection is controlled at the source by a token bucket, that is, for $i \in S_{12} \cup S_1, j = 1$ or $i \in S_2, j = 2$

$$b_{i,j}(I) = \min\{I, \alpha_i + \rho_i * I\}. \tag{4}$$

2.1 Main Results

The delay at a server can be determined once the output traffic at that server is known. The following lemma, which was first presented in [1], addresses this.

Lemma 1. For a single FIFO server j, if the aggregated arrival traffic function $G_j(t)$ is known, its output traffic function $W_j(t)$ can be written as

$$W_j(t) = \min_{0 \leq s \leq t} \{t - s + G_j(s)\}, \tag{5}$$

where

$$G_j(t) = \sum_{k \in S_j} f_{k,j}(t), \tag{6}$$

where S_j is the set of connections that traverse Server j.

Once we know the output traffic of a server, we also know the arrival time for the data leaving at any particular point in time. The following lemma gives the relationship between the output traffic and the data arrival time.

Lemma 2. During the time interval $[0, t)$, if the total amount of data leaving Server j is $W_j(t)$, the time $H_j(t)$ when the $W_j(t)$-th bit arrives at Server j is given as

$$H_j(t) = G_j^{-1}(W_j(t)). \tag{7}$$

Note: $H_j(t) \leq t$.

Proof: The lemma follows from the definition of function $G_i(t)$. Q.E.D

Similarly, we can formulate when the arriving data will leave the server, as the following lemma shows.

Lemma 3. If the total amount of data arriving at Server j during the time interval $[0, t)$ is $G_j(t)$, then the $G_j(t)$-th bit leaves Server j at time $W_j^{-1}(G_j(t))$.

Proof: The lemma follows from the definition of function $W_j(t)$. Q.E.D

We can now apply these results to accurately determine the end-to-end delay suffered by traffic as it traversed the two-server subsystem depicted in Figure 1.

Lemma 4. The end-to-end delays of connections in S_{12} (that is, traversing both Server 1 and Server 2) are bounded by

$$d_{S_{12}} = \max_{t \geq 0} \{W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))\}. \tag{8}$$

Proof: During the time interval $[0, t)$, the total amount of traffic arriving at Server 2 is $G_2(t)$. According to Lemma 3, the $G_2(t)$-th bit leaves Server 2 at time $W_2^{-1}(G_2(t))$. Furthermore, these $G_2(t)$ contains $W_1(t)$ bits coming from Server 1. According to Lemma 2, the $W_1(t)$-th bit arrives at Server 1 at time $G_1^{-1}(W_1(t))$. Therefore, the delay suffered at time t by connections traversing both servers is given as $W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))$. So we have

$$d_{\mathcal{S}_{12}} = \max_{t \geq 0}\{W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))\}, \qquad (9)$$

Q.E.D

Unfortunately, Equation (8) is only of theoretical value. This is because it requires the knowledge of internal network traffic (in form of $G_2(t)$). Since the only information we assume are the traffic constraint functions at the sources, and the traffic is not reshaped internally, the internal network traffic is difficult, if not impossible, to describe. In order to provide a useful integrated method for delay analysis in this subsystem, we need to deeply analyze Equation (8).

The following central theorem in this paper provides an estimation for $d_{\mathcal{S}_{12}}$ in Lemma 4 To streamline the presentation of the theorem, we define the following auxiliary notations:

- $\bar{G}_1(t) = \sum_{i \in S_{12} \cup S_1} b_{i,1}(t)$.
- $\bar{W}_1(t) = \min_{0 \leq s \leq t}\{t - s + \bar{G}_1(s)\}$.
- $\bar{H}_1(t) = \bar{G}_1^{-1}(\bar{W}_1(t))$.
- $F_{12}(t) = \sum_{i \in S_{12}} b_{i,1}(t)$.
- $F_2(t) = \sum_{i \in S_2} b_{i,2}(t)$.

Theorem 1. The delay suffered by Connections in S_{12} is bounded by

$$\begin{aligned} d_{S_{12}} \quad \leq \quad & \max_{0 \leq s \leq \mathcal{B}_1}\{\max_{\mathcal{B}_1 + \mathcal{B}_2 \geq T \geq s}\{s + \\ & \min\{T - s, F_{12}(T - \bar{H}_1(s))\} + F_2(T - s)\} \\ & - \min\{T, \bar{G}_1^{-1}(T)\}\}\}, \end{aligned}$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ are the length of maximum busy periods on Server 1 and Server 2, respectively.

Proof: See [25]. Q.E.D

We note that, according to Theorem 1, the end-to-end delay $d_{S_{12}}$ of connections traversing both servers can be computed using only bounding functions for the traffic entering the subsystem. This eliminates the problems described earlier with Equation (8) and provides a practical method to analyze end-to-end delays, as we proceed to describe below.

3 New Delay Analysis Algorithm

A common method to analyze the end-to-end delays suffered by connections in networks, with or without traffic regulation at intermediate nodes, consists of two steps. In a first step, a single-server analysis technique is developed to estimated the local worst case delay and characterize the output traffic, provided characterizations of all input traffic of the server. In a second step, starting from characterizations of all source traffic, local delay analysis is successively performed on each server along the path of the connection. As described earlier, the main disadvantage of this method is that the delay dependencies in successive servers without traffic regulation on a connection's path is ignored. So the obtained end-to-end delay bounds are very loose and the bursts are overestimated.

3.1 A New algorithm

Algorithm *Integrated:*

Step 1: Partition the network into subnetworks, each of them consists at most of two servers..

Step 2: Chose the appropriate order for all subnetworks such that each input traffic of $(i+1)$-th subnetwork can be estimated by all input traffic of subsystems with order less than $(i+1)$-th.

Step 3: Traverse in the subnetworks in the topological ordering, performing the following steps for each subnetwork:

Step 3.1: Compute the delay bounds suffered by connections in the subnetwork.

Step 3.2: Estimate the output traffic of the subnetwork.

Step 4: Compute the end-to-end delays for each connection by summing up all local delays suffered at every subnetwork along its path.

Figure 2. Algorithm *Integrated*.

Equation (10) can be used as the basis for improved end-to-end analysis methods, which better take into account delay dependencies. Algorithm *Integrated*, described in Figure 2, computes end-to-end delays in a cycle-free network with FIFO servers. It first partitions the network into subnetworks of one or two servers each (Step 1). It then identifies a topological ordering of subnetworks (Step 2). Next, it computes the local delays (Step 3.1) and the output traffic (Step 3.2) at each subnetwork. Finally, it determines the end-to-end delays by summing up the previously computed local delays (Step 4).

4 Evaluation

In a suite of simulation experiments we compared the proposed new method for end-to-end delay analysis with that of two commonly used methods ([8, 9]), which we call Algorithm *Decomposed* and Algorithm *Service Function*. These methods were originally proposed by Cruz and adopted in various forms by many others. We compare their performance on a network with FIFO servers arranged in a feedforward topology. These experiments show that our new method generally computes tighter bounds on end-to-end delays than the other approaches.

4.1 Experiments

We evaluate the performance of the new approach by comparing Algorithm *Integrated* to the delay computation methods described by Cruz in [8, 9], which we call Algorithm *Cruz*. In this section, we first define the performance metric and then describe the system configuration considered. The performance results will be presented and discussed in the next section.

Topology and Traffic Descriptions. In our evaluation, we consider a simple tandem network with n 3×3 switches, which are connected in a chain. An example of such a

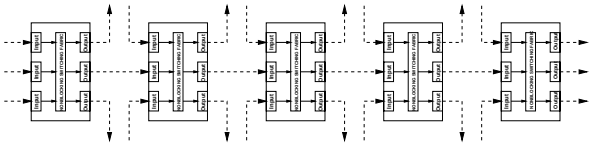

Figure 3. A Tandem Network.

tandem network with 5 switches is illustrated in Figure 3. There are $2n+1$ connections in this network. Connection 0 is the longest; it enters the network at the middle input port of the first switch and exits the network from the middle output port of the n-th switch. For $k = 0$ to $n-1$, the $(2k+1)$-th session enters the network by the upper input port of the k-th switch and exits the network from the upper output port of the $(k+1)$-th switch, the $(2k+1)$-th session enters the network by the lower input port of the k-th switch and exits the network from the upper lower output port of the $(k+2)$-th switch. The middle output port of each switch, excepted the first one, carries four connections, including Connection 0. In order to simplify the evaluation, we assume that every source traffic is controlled by a token bucket with a unit bucket size ($\alpha = 1$) and the token arrival rate $\rho = \frac{U}{4}$, where U is the work load of the network. While an increase of the traffic burstiness (larger value for α) increases the overall end-to-end delays, our experiments indicate that it does not affect the relative performance of the approaches evaluated in these experiments. In particular, increasing the traffic burstiness has no effect on the relative improvement $R_{X,Y}$ (defined below) for any pairing of methods.

Performance Metric. We quantify the performance of algorithms using two measures. One is the *end-to-end delay* $D_0^C(U)$ estimated by Algorithm X for the end-to-end delay suffered by the connection which travels the longest path in the network (Connection 0 in our case) under the work load U. The other is called the *relative improvement* $R_{X,Y}(U)$, which is used to compare two algorithms and is expressed as

$$R_{X,Y}(U) = \frac{D_0^X(U) - D_0^Y(U)}{D_0^X(U)}. \qquad (10)$$

4.2 Delay Computation

In [25] we summarize the formulas used for the delay calculation in the decomposition based and service-curve based approach as described in [8, 9, 10, 5, 32]. We use these formulas to derive closed forms for the worst-case delay for Connection 0 in the topology used in these experiments. We call the resulting delay computation algorithms Algorithm *Decomposed* for the decomposed approach and Algorithm *Service Curve* for the service-curve based approach.

Algorithm *Decomposed* We derive the worst-case end-to-end delay of Connection 0 by adding the local delays on the servers along its path. For this, we let E_k be the local delay suffered by traffic of Connection) at Server k. In [25], we derive the following equations for E_k:

$$E_1 = \frac{2\alpha}{1-\rho}; \quad E_2 = \alpha\frac{3-\rho+4\rho^2}{(1-\rho)^2}$$

$$E_k = 3\alpha + \rho E_{k-1} + 3\rho\frac{\alpha + \rho\sum_{i=1}^{k-1} E_i}{1-\rho}, k \geq 3$$

The end-to-end delay D_0^D for Connection 0 using Algorithm *Decomposed* is then obtained by adding the local delays:

$$D_0^D = \sum_{k=1}^{n} E_k$$

Algorithm *Service Curve* The delay calculation in this approach is based on the definition for the service curve given in [10]. As we compare the performance of the various approaches for a network with pre-defined servers (FIFO servers in this case), synthesizing scheduling algorithms from pre-defined service curves is not viable. We

must use an induced service curve approach, where we derive the service curve from the scheduling policy used in the server. The performance of such a method, however, greatly depends on how tight service curves can be defined for a given service discipline. In [25] we derive an *upper bound* on the service curve for a FIFO server, which in turn give raise to a lower bound for the end-to-end delay D_0^{SC} for Connection 0 with the service curve method. As we derive in [25], the worst case delay D_0^{SC} is lower-bounded by the following expression:

$$D_0 \geq \frac{2\alpha}{1-2\rho} + \frac{\alpha(3-2\rho)}{(1-\rho)(1-3\rho)} + \frac{(n-2)\alpha(3-\rho)}{(1-\rho)(1-3\rho)}.$$

It is important to emphasize at this point that the following comparisons are between *upper bounds* on end-to-end delays for both Algorithm *Integrated* and Algorithm *Decomposed*, and *lower bounds* for Algorithm *Service Curve*. The results for the performance of Algorithm *Service Curve*, both in terms of end-to-end delays and in terms of relative performance, must therefore be considered as optimistic.

4.3 Numerical Results and Observations

The results of our experiments comparing the performance of the three approaches are depicted in Figures (4), (5), and (6). Figure 4 compares the service-curve based approach to the decomposition-based approach and illustrates how the former is not well suited (as was to be expected) for analyzing non-guaranteed-rate service disciplines, in this case FIFO. As the network load increases, the inadequacy of modeling a FIFO server with a service curve becomes evident. For larger systems, this gets partly offset by the compounding effects of summing convervative local delay bounds in the decomposition-based approach.

From Figure 5 we see that Algorithm *Integrated* always outperforms Algorithm *Decomposed*. Furthermore, for loads up to 80%, the performance improvement increases with growing network size. This is expected as Algorithm *Integrated* takes delay dependences within server pairs into account.

While the performance improvement of Algorithm *Integrated* over Algorithm *Service Curve* can be inferred by transitivity, we show a comparison in Figure 6 for illustrative purposes. The results of this experiment show that the performance gains are significant, except for large systems under high load.

5 Conclusion

In this paper, we have proposed a new method for deriving end-to-end delay bounds for connections in tandem

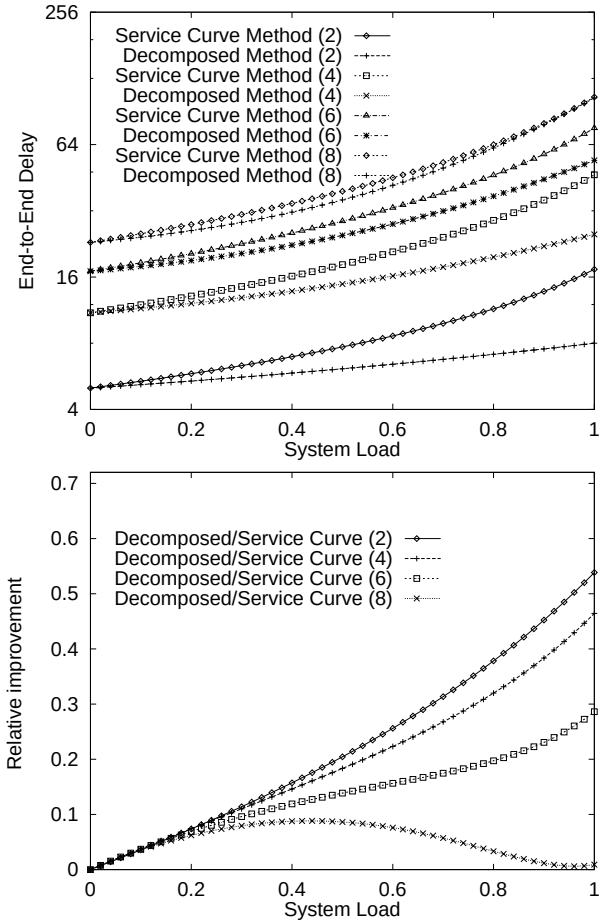

Figure 4. Comparison between Decomposed Method and Service Curve Method.

network, which uses a FIFO scheduling discipline. Our new method takes into account delay dependencies in successive servers along the path of a connection, which is in general very difficult for delay analysis, and achieves better performance than the method provided in [8, 9]. This can be observed through the extensive simulation experiments provided in previous section.

When servers do not have traffic regulation mechanisms (as is the fact with all work conserving servers), circular dependencies amond connections introduce feedback effects on local delays, which in turn show up as non-linearities in the local delay calculations. For this reason, the analysis method described in this paper is limited to sets of connections that do not generate cycles in the network. Based on our previous work on decomposition-based analysis with feedback effects of networks with both FIFO and static-priority servers ([23]), we currently working on extending the approach proposed in this paper to general networks.

Although the integrated approach for analyzing pairs of servers presented in this paper is generic in principle, we

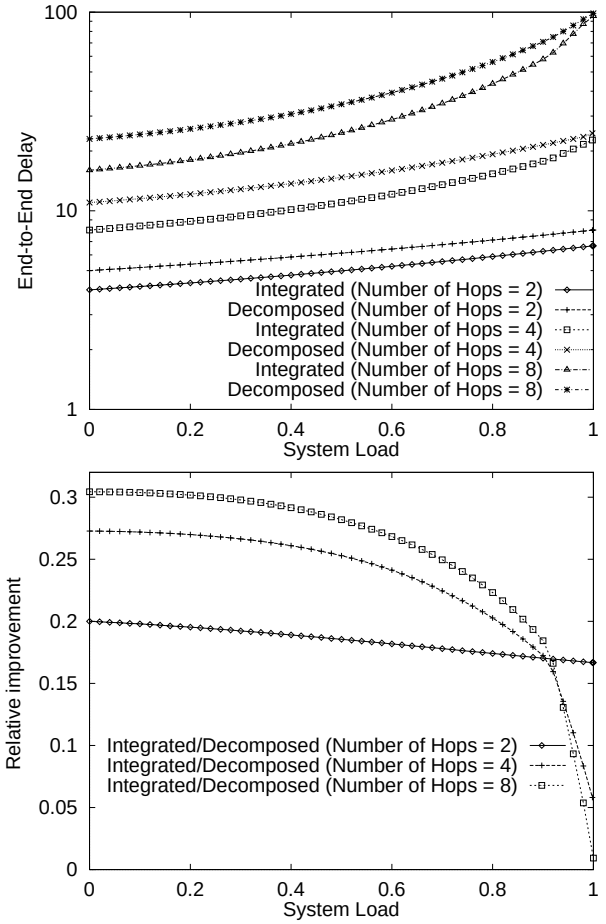

Figure 5. Comparison between Integrated Method and Decomposed Method.

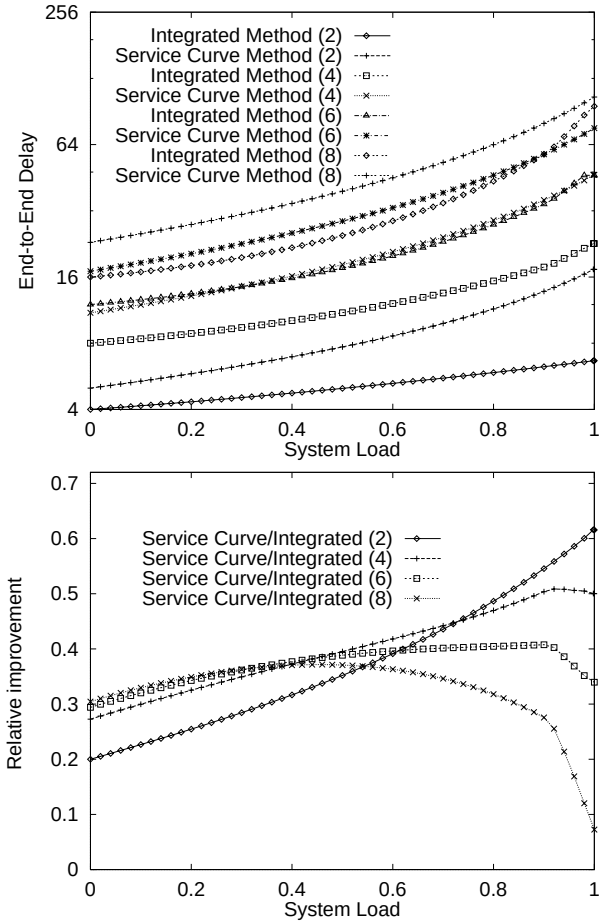

Figure 6. Comparison between Integrated Method and Service Curve Method.

derived the closed form delay formulas for the FIFO service discipline (in Theorem 1). We are currently extending the applicability of this approach to the static-priority discipline by deriving the appropriate closed form solutions of the delay formulas.

Acknowledgment

This work was partially sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number F49620-96-1-1076, by the Defense Advanced Research Projects Agency (DARPA) through the Honeywell Technology Center under contract number B09333438, and by Texas Higher Education Coordinating Board under its Advanced Technology Program with grant number 999903-204. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, Honeywell, DARPA, the U.S. Government, Texas State Government , Texas Higher Education Coordinating Board, or Texas A&M University.

References

[1] R. Agrawal and R. Rajan. Performance bounds for guaranteed and adaptive services. In *IBM research report RC 20469(91385)*, 1996.

[2] C. M. Aras, J. Kurose, D. S. Reeves, and H. Schulzrinne. Real-time communication in packet-switched networks. In *Proceedings of IEEE*, vol. 82, no. 1, 1994.

[3] A. Banerjea and S. Keshav. Queueing delays in rate controlled ATM networks. *Proceedinds of Infor-com'93*, 1993.

[4] J. Bennett and H. Zhang. WF^2Q: worst-case fair weighted fair queueing. *Proc. of IEEE INFOCOM'96*, 1996.

[5] J.-Y. Le Boudec. Application of network calculus to guaranteed service networks. *Accepted for Trans. on Information Theory*, 1997.

[6] J.-Y. Le Boudec, G. D. Veciana and J. Walrand. QoS in ATM: Theory and Practice. *Inforcom'98*, 1998.

[7] D. D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proceedings of ACM SIGCOMM'92*, pages 14–26, Aug. 1992.

[8] R. L. Cruz. A calculus for network delay. *IEEE Transactions on Information Theory*, 37(1):114–131, Jan. 1991.

[9] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, Jan. 1991.

[10] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, 1995.

[11] A. Dailianas and A. Bovopoulis. Real-time admission control algorithms with delay and loss guarantees in ATM networks. In *Proceedings of INFOCOM'94*, pages 1065–1072, 1994.

[12] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proceedings of ACM SIGCOMM'89*, pages 1–12, Sept. 1989.

[13] D. Ferrari. Client Requirements for real-time communication services. *IEEE Communication Magazine*, vol. 28, no. 11, 1990.

[14] N. R. Figueira and J. Pasquale. An upper bound on delay for the virtual clock service discipline. *IEEE/ACM Trans. Networking*, vol. 3, no. 4, 1995.

[15] V. Firoiu, J. Kurose, and D. Towsley. Efficient admission control for EDF schedulers. *Proceedings of Inforcom'97*, 1997.

[16] L. Georgiadis, R. Guérin, V. Peris, and K. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE ACM Transactions on Networking*, 4(4):482–501, Aug. 1996.

[17] S. J. Golestani. Network delay analysis of a class of fair queueing algorithms. *IEEE J. on Seleted Areas in Communications*, vol. 13, no. 6, 1995.

[18] P. Goyal, S. S. Lam, and H. Vin. Determining end-to-end delay bounds in heterogeneous networks. In *5th Int. Workshop on network and Op. Sys. support for Digital Audio and Video*, 1995.

[19] E. W. Knightly. On the accuracy of admission control tests. *Proceedings of ICNP'97*, 1997.

[20] E. W. Knightly. Enforceable quality of service guarantees for bursty traffic systems. *Proceedings of the IEEE Infocom'97*, 1997.

[21] K. Lee. Performance bounds in communication networks with variable-rate links. *Proceedings of SigComm'95*, 1995.

[22] C. Li, A. Raha, and W. Zhao. Stability in ATM networks. In *Proceedings of the IEEE Infocom'97*, 1997.

[23] C. Li, R. Bettati, and W. Zhao. Static priority scheduling for ATM networks. *Proceedings of the 18th IEEE Real-Time Systems Symposium*, 1997.

[24] C. Li, R. Bettati and W. Zhao. Response time analysis for distributed real-time systems with bursty job arrivals. *Proceedings of IEEE ICPP*, 1998.

[25] C. Li, R. Bettati and W. Zhao. New delay analysis in high speed networks. *Technical Report, Department of Computer Science, Texas A&M University*, 1999.

[26] J. Liebeherr, D.E. Wrege, and D. Ferrari. *"Exact admission control in networks with bounded delay services."*, to appear in *IEEE/ACM Transactions on Networking*.

[27] S. Low. Traffic management of ATM networks: service provisioning, routing, and traffic shaping. Ph.D Dissertation. UC Berkeley, 1992.

[28] R. Nagarajan, J. Kurose, and D. Towsley. Local allocation of end-to-end quality of service measures in high speed networks. In *IFIP Int. Workshop on Modeling of ATM Networks*, 1993.

[29] J. K. Ng, S. Song, C. Li, and W. Zhao. A new method for integrated end-to-end delay analysis in ATM networks. *Accepted by Journal of Communications and Networks*.

[30] A. K. J. Parekh and R. G. Gallager. A generalized processor sharing aApproach to flow control in integrated services networks: the single-node case. In *IEEE/ACM Trans. Networking*, vol. 1, no. 3, 1993.

[31] A. K. J. Parekh and R. G. Gallager. A generalized processor sharing aApproach to flow control in integrated services networks: the multiple-node case. In *IEEE/ACM Trans. Networking*, vol. 2, no. 2, 1994.

[32] H. Sariowan and R. L. Cruz. Scheduling for quality of service guarantees via service curves. *Proceedings of ICCCN'1995*, 1995.

[33] H. Sariowan. A service-curve approach to performance guarantees in integrated-service networks. Ph. D. Dissertation. University of California, San Diego, 1996.

[34] D. Stiliadis and A. Varma. Latency-rate server: a general model for analysis of traffic sceduling algorithms. *Proceedings of the IEEE Inforcom'96*, 1996.

[35] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for vbr video in packet-switching networks: Fundamental limits and practical tradeoffs. *IEEE/ACM Trans. on Networking*, 4(3), 1996.

[36] H. Zhang. Service disciplines for guaranteed performance service in packet switching networks. In *Proc. IEEE*, 1995.

[37] Q. Zheng and K. G. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Trans. on Communications*, vol. 42, no. 2, 1994.