

# QoS for Storage Subsystems Using IEEE-1394

CHIH-YUAN HUANG, TEI-WEI KUO, and AI-CHUN PANG  
National Taiwan University

---

IEEE-1394 is widely adopted in various commercial products for computing, communication, and entertainment. Although many services with Quality-of-Service (QoS) supports are now available in systems over IEEE-1394, little work is done for QoS-based resource allocation. In this article, we aim at the design of a bandwidth reservation mechanism and its policy for isochronous requests, such as those from cameras. We then address the QoS support issue for asynchronous requests, such as those from disks, and an analytic framework for probability-based QoS guarantees. This work is concluded by the proposing of a topology configuration algorithm for IEEE-1394 devices. The capability of the proposed methodology and the analytic framework are evaluated by a series of experiments over a Linux-based system prototype.

Categories and Subject Descriptors: C.0 [**Computer Systems Organization**]: General—*Hardware/software interfaces; system architectures*; B.4.2 [**Input/Output and Data Communications**]: Input/Output Devices—*Channels and controllers*

General Terms: Design, Management, Performance

Additional Key Words and Phrases: I/O subsystem, quality-of-service, real time, IEEE-1394

## ACM Reference Format:

Huang, C.-Y., Kuo, T.-W., and Pang, A.-C. 2009. QoS for storage subsystems using IEEE-1394. *ACM Trans. Storage*, 4, 4, Article 12 (January 2009), 17 pages. DOI = 10.1145/1480439.1480441 <http://doi.acm.org/10.1145/1480439.1480441>

---

## 1. INTRODUCTION

The emergence of various digital devices in the market has triggered a strong demand on I/O interface products. Such a demand is aggravated by the rapidly increasing amount of digital contents. Typical designs, based on serial, parallel, or even USB 1.1 buses, could no longer meet the demands of users and the market. The massive amount of digital contents would impose serious

---

This work is supported in part by research grants from the RoC National Science Council under Grants NSC 93-2752-E-002-008-PAE, 94-2213-E-002-007, and 96-2219-E-002-004.

Authors' address: C.-Y. Huang, T.-W. Kuo, A.-C. Pang, Department of Computer Science and Information Engineering, Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan 106, ROC; email: {ktw,acpang}@csie.ntu.edu.tw.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org). © 2009 ACM 1553-3077/2009/01-ART12 \$5.00 DOI 10.1145/1480439.1480441 <http://doi.acm.org/10.1145/1480439.1480441>

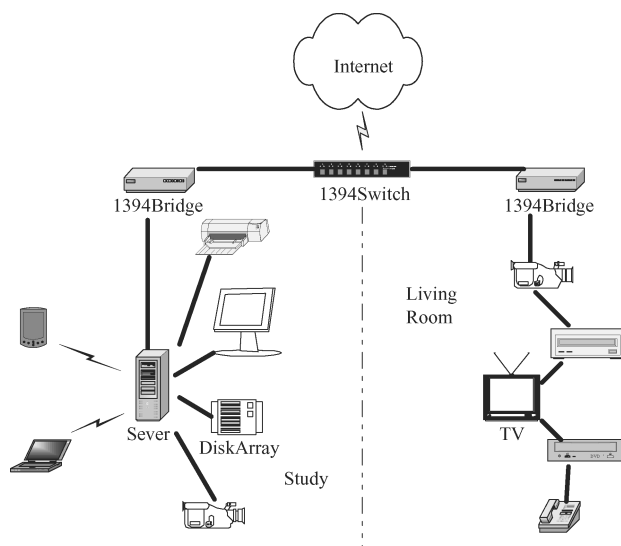


Fig. 1. An example IEEE-1394-based home network.

challenges on the resource supply and management of I/O interface products, especially if multiple services with QoS supports (e.g., video on demand) compete with each another over the same I/O bus. Among the popular I/O buses in the market, many well-known products are based on USB 2.0, SCSI, and IEEE-1394. They provide not only convenience in device connection but also high bandwidth sharing among devices (or consumer electronics) and computers/servers. For example, IEEE-1394 [IEEE 1996], that could serve as the home backbone network or for remote instrumentation and control, is widely adopted as an interface for many digital home devices, such as digital TV's, digital camcorders, and digital video cassette recorders. An example IEEE-1394-based home network is as shown in Figure 1.

Exploration of QoS supports over I/O buses is not only important for personal computer systems, but also critical for various forms of embedded systems because different forms of traffic with different QoS requirements would be competing on a finite amount of bandwidth. Such competition could be even more serious if the demands of massive amounts of digital content were to flow over these buses. Although many results have been proposed in applying I/O buses for home usage, such as a network file system in an IEEE-1394 home network [Igarashi et al. 1999] and an IEEE-1394 home network architecture [Saito et al. 2000; Ku et al. 2002; Zhang et al. 2001], little work is done to provide QoS guarantees for user requests over I/O buses. In particular, Huang and Kuo [2003] and Huang et al. [2004] explored QoS-based designs for USB 2.0 and 1.1 bandwidth management. An integrated real-time driver architecture was proposed over existing USB implementations. Lim et al. [2000] and Park and Kang [2000] proposed an IEEE-1394 device driver architecture that could support priority-based packet queuing at the driver level and thus improve the predictability in servicing high-priority applications.

The objective of this research is to explore the design and implementation of QoS supports for IEEE-1394 device management. Although the IEEE-1394 bus management protocol partitions the bandwidth usages of devices of different types (i.e., isochronous and asynchronous transfer types) and lets each isochronous request ask for the maximum number of bytes being transferred for a device type in each cycle (i.e.,  $100\mu s$  in  $125\mu s$ ), QoS supports over IEEE-1394 suffer some major problems: (1) Popular devices, such as storage devices, might suffer unpredictable performance when the bus is (temporarily) heavily loaded; (2) the QoS support for requests of isochronous data transfers should consist of the minimum number of bytes guaranteed to be transferred for a device in each cycle. Different from past work in priority-driven scheduling of application packets, we aim at resource reservation for requests and their analytic framework for QoS guarantees. In this article, we first propose the design of a bandwidth reservation mechanism and its policy for isochronous requests, such as those from cameras or for digital TV's. Each isochronous request is provided with a hard QoS guarantee by reserving an amount of IEEE-1394 bandwidth for data transfers. We then consider the mixture of isochronous and asynchronous requests in admission control for QoS services. An analytic framework for probability-based (soft) QoS guarantees is then proposed for asynchronous requests, since asynchronous requests are serviced in a best-effort fashion over IEEE-1394. A topology configuration algorithm is also proposed for IEEE-1394 devices to better improve the bandwidth utilization of an IEEE-1394 bus. The capability of the proposed methodology and the analytic framework are evaluated and demonstrated by a series of experiments over a Linux-based system prototype.

We must point out that many applications that rely on QoS supports from storage systems usually need to access data with response-time requirements. It is of paramount importance to know the bandwidth requirements of devices (or their storage subsystems) and to satisfy their needs. Our proposed methodology helps in the derivation of the mean waiting time of each device so as to guarantee the QoS guarantees of devices based on the bandwidth requirements (please see formula (11)). As a result, the analysis is one aspect of QoS of storage subsystems.

The rest of this article is organized as follows: Section 2 provides an IEEE-1394 overview and provides motivations for this work. In Section 3, a real-time IEEE-1394 driver architecture is proposed with admission control. A bandwidth reservation mechanism is presented for isochronous requests. An analytic framework is then proposed for asynchronous requests. A topology configuration algorithm for IEEE-1394 devices is also proposed. Section 4 provides a performance evaluation of the proposed methodology. Section 5 is the conclusion.

## 2. AN IEEE-1394 SUBSYSTEM AND MOTIVATION

### 2.1 IEEE-1394

The IEEE-1394 serial bus is designed to function as both a cable-connected virtual bus and a backplane bus with a speed of up to 400Mbps. Compared

to many serial and parallel I/O buses, IEEE-1394 offers a rich set of advanced features, including online connection/disconnection of devices in an IEEE-1394-based cable I/O system or boards to an IEEE-1394-based backplane. The IEEE-1394 standard supports both asynchronous and isochronous transfer modes of communication in an I/O system, where asynchronous and isochronous transfer modes correspond to sporadic and periodic transfers, respectively. The isochronous transfer mode can be used to guarantee a certain amount of bandwidth for a device with stringent response time requirements, such as those for streaming video or audio. The asynchronous transfer mode is appropriate for typical PC peripherals, such as printers or disk drives. Asynchronous transfers support bursty traffic and do not require constant-rate data delivery. In such a transfer mode, bus accesses are done in a contention fashion, and there is no bandwidth reservation. On the other hand, each data transfer in the isochronous transfer mode is done at a higher priority than that in the asynchronous transfer mode, and it is done at a constant interval. The destination of a data transfer in the isochronous transfer mode is defined as a channel number so that isochronous data streams could be broadcast to more than one node (i.e., active devices), instead of that in the synchronous transfer mode.

The IEEE-1394 serial bus supports peer-to-peer transactions. Arbitration must be done in a fair fashion, for transfer requests to determine which node has ownership of the bus. Prior to any data transfer, the transmitting node must obtain the ownership of the IEEE-1394 bus such that only one node at a time can transmit data over the wire. Data transmission could be done with three speeds: 100Mbps, 200Mbps, and 400Mbps. During the system initialization, the bus manager collects self-id packets from nodes to derive the network topology, where self-id packets were sent by nodes with source capabilities and sink requirements. A speed map based on the maximum transmission speed between every two nodes is thus built. The topology of an IEEE-1394 network resembles a tree structure, where the maximum number of nodes is 63, and 64,000 devices can be configured. Each node in an IEEE-1394 network functions as a repeater by forwarding signals to the next node in the daisy-chain.

As shown in Figure 2, upper-layer applications pass requests to the software driver. Based on the request type (asynchronous or isochronous), the driver sends the request to the corresponding interface to transfer data streams over the IEEE-1394 bus. The IEEE-1394 bus is a layered structure, where four layers, namely, serial bus management, transaction, link, and physical layers, are defined in the protocol stack. The serial bus management layer supports bus configuration and management activities for each device node. The transaction layer, that is the implementation of the request-response protocol specified in the standard [ISO 2009], supports asynchronous commands such as writes, reads, and locks. The transaction layer also provides a retry mechanism to handle situations where resources are busy. The link and physical layers are the main building blocks of the IEEE-1394 stack. The link layer communicates with the transaction layer with acknowledged datagrams (i.e., one-way data transfers with request confirmation) and handles packet transmission/reception and the cycle control for isochronous channels. The physical layer

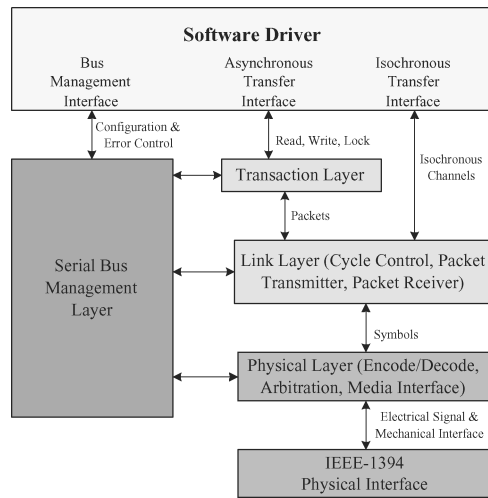


Fig. 2. An IEEE-1394 protocol stack.

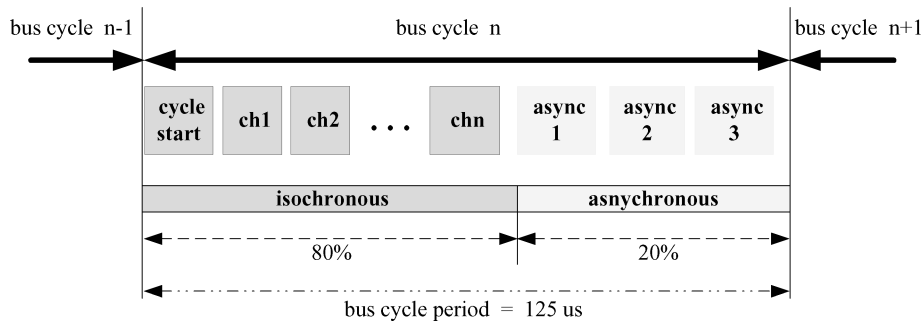


Fig. 3. An IEEE-1394 bus cycle.

passes data streams to the physical interface (also referred to as the transceiver) and provides initialization and arbitration services to ensure that only one node sends data at any time. The physical interface translates serial data streams and signal levels before they are sent to the link layer. The physical-interface devices function as repeaters by allowing data to pass from one device to another without going through the link layer.

## 2.2 Motivation

Data transfers over an IEEE-1394 bus are done in cycles, where each cycle is of  $125\mu s$  typically (i.e., in a frequency  $8k Hz$ ). Each cycle starts with a cycle-start packet, which contains the value of the clock counter of the cycle master, where the master (determined during the bus initialization) does arbitration for the bus. Each device attached to the bus must update its local clock counter based on the clock counter recorded in the cycle-start packet. Transfers in a cycle are done in two time segments, as shown in Figure 3, where the first and second time segments are for isochronous and asynchronous data transfers, respectively.

Isochronous data transfers are done over pre-assigned channels in a broadcasting fashion and are suitable to periodic data transfers. The asynchronous mode is for data transfers in a handshaking fashion. The asynchronous transactions operate on the IEEE-1394 bus based on a rotational priority scheme called the *fairness interval*. The *fairness interval* ensures that each node intending to initiate a transaction obtains a fair access to the bus. The principle is that all nodes that have one or more pending asynchronous transactions are permitted to obtain the bus ownership only one time for each fairness interval.

The Quality-of-Service (QoS) support for IEEE-1394 is mainly achieved in two parts: (1) The IEEE-1394 bus management protocol guarantees that isochronous data transfers could only use up to 80% of the total bandwidth in each cycle (i.e.,  $100\mu s$  in  $125\mu s$ ) so that asynchronous data transfers could still receive a reasonable amount of service time. (2) A request for isochronous data transfers could ask for the maximum number of bytes being transferred for a device type in each cycle. When the traffic load is reasonable, the worst-case delay time for each bus request and the buffer size requirements could be derived. Such supports could suffer two major technical problems in QoS services: (1) Popular devices, such as disks, might suffer from unpredictable performance when the bus is (temporarily) heavily loaded. There is no QoS support or analysis framework for such devices. (2) The QoS support for requests of isochronous data transfer should be the minimum number of bytes guaranteed to be transferred for a device in each cycle. There is little work being done for such QoS supports. Proper resource reservation mechanisms are needed based on the IEEE-1394 standard.

We must point out that the configuration problem remains, even with the aforementioned two technical problems resolved, where the configuration problem determines which node should be directly connected to which node to obtain the best performance or to meet the requirements of applications (i.e., a topology design problem). The configuration problem exists due to the fact that the maximum speed for data transfers between two directly connected nodes is determined by the minimum capability of the two nodes. Note that nodes connected (directly or indirectly) by IEEE-1394 could send data to each another. Nodes in a path in the passing of data sent from one node to another node would function as repeaters. An improper design of a topology could easily result in hot spots with a bad capability such that the QoS requirements of applications fail. These observations motivate our work in the proposing of proper resource scheduling methods for IEEE-1394 and the study of the configuration problem.

### 3. QOS SERVICES FOR ISOCHRONOUS AND ASYNCHRONOUS TRANSFERS

#### 3.1 Overview

The purpose of this section is to develop a resource scheduling policy of an integrated QoS service for isochronous and asynchronous requests and its device topology configuration problem. The topology of an IEEE-1394 system can be

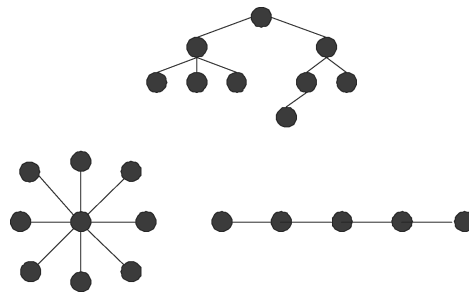


Fig. 4. IEEE-1394 system topologies.

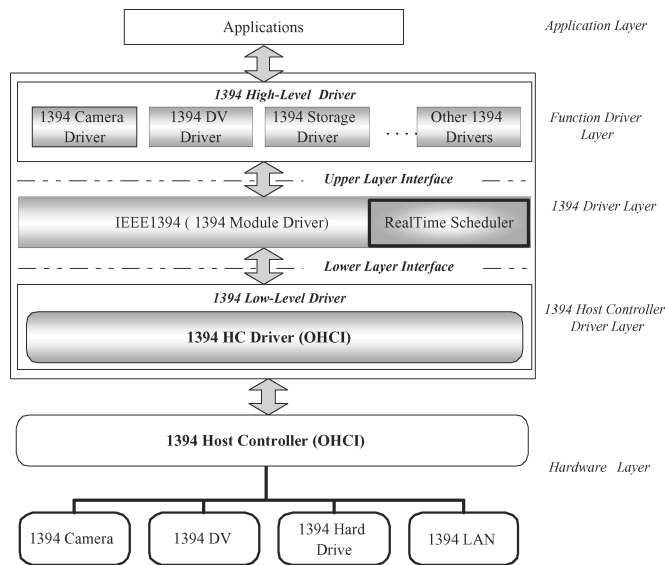


Fig. 5. A real-time 1394 device driver hierarchy.

a daisy-chain, a star, or a tree, where a daisy-chain topology or a star topology could be considered as special cases of a tree topology, as shown in Figure 4. Each IEEE-1394 device in an IEEE-1394 network functions as a repeater by forwarding signals to other devices. The speed specified for a cable is governed by the slowest-speed device attached to the cable. As a result, the maximum transmission speed between any two devices is limited by the cable with the smallest transmission speed in the path.

Figure 5 shows an example of the layered implementation of Linux-based IEEE-1394 device drivers. The driver is partitioned into three layers: the function driver layer, the IEEE-1394 driver layer, and the IEEE-1394 host controller driver layer. Devices, such as camera, digital video, and disk, are connected to the IEEE-1394 bus through the function drivers. The module **IEEE1394**, that is the core of an IEEE-1394 driver subsystem, manages the high-level function drivers and low-level host controller drivers, including *aic5800*, *pcilynx*, and

ohci1394.<sup>1</sup> In this section, a real-time scheduler is proposed in the IEEE-1394 driver layer for request scheduling and admission control of isochronous and asynchronous transfers. We aim at the provisioning of a hard QoS guarantee for isochronous requests (Section 3.2) and a soft QoS guarantee for asynchronous requests (Section 3.3). An evaluation framework is developed to derive the average response time and waiting time for asynchronous requests (Section 3.3). The utilization of IEEE-1394 bus bandwidth is considered under the satisfaction of the QoS requirements. Because the topology of an IEEE-1394 network significantly affects the utilization of the bus bandwidth, we target the configuration problem of IEEE-1394 devices by proposing a max-traffic-flow algorithm to maximize the total traffic flow of IEEE-1394 buses. In the algorithm, the IEEE-1394 bandwidth is better utilized by a proper placement plan of IEEE-1394 devices (Section 3.4).

### 3.2 QoS Guarantees for Isochronous Requests

The purpose of this section is to propose a real-time scheduler in the IEEE-1394 driver layer for the scheduling and admission control of isochronous requests. In this article, requests are referred to the commands issued by the software driver, as shown in Figure 5, where transfers are referred to data transmissions over IEEE-1394 buses. Regardless of the network topology of an IEEE-1394 system, one unique device in the system is picked up as the root. When only one device has a host controller, the device is the root; otherwise, the device with a host controller and the smallest node-id is the root. The root is responsible to the request scheduling of the entire system according to the IEEE-1394 standard.

The network topology for an IEEE-1394 system can be modeled as a graph  $G = (V, E)$ , where each vertex in  $V$  corresponds to one IEEE-1394 device in the system, and each edge (also referred to as a link) in  $E$  corresponds to a link that connects two nodes, that is, two IEEE-1394 devices. Since  $G$  is a tree,  $|V| = n$  and  $|E| = n - 1$ . Let a matrix  $M = \{m_{i,j}\}_{N \times N}$  denote the maximum transmission speed  $m_{i,j}$  between any two devices  $n_i$  and  $n_j$  in the system, where  $N$  is the number of devices in the system. Suppose that devices are renumbered from 1 to  $N$  for simplicity of presentation. (Please see Section 3.4 for configuration determination in the determination of the matrix, where the configuration of the system determines the transfer speeds.)

Consider a set of isochronous requests  $T_{ISO} = \{T_{I_1}, T_{I_2}, \dots, T_{I_M}\}$  for request scheduling and QoS guarantee. Each isochronous request  $T_{I_i}$  to a device of the isochronous transfer type could be modeled by three parameters:  $T_{I_i} = (c_i, p_i, s_i)$  in  $T_{ISO}$ , where  $c_i$  is the payload size of each transfer,  $p_i$  is the period between two consecutive transfers (unit: 125us), and  $s_i$  is the maximum transmission speed for the request. As shown in Figure 3, data transfers over an IEEE-1394 bus are done in cycles, where each cycle is typically of 125 $\mu$ s. For each bus cycle, the total transmission time for isochronous requests shall be limited to 80% of a cycle time (i.e. 100 $\mu$ s). When resource reservation is enforced for the

<sup>1</sup>In this article, the ohci1394 module for OHCI host controller is considered.



set of isochronous requests, the total utilization of the bus cycle for isochronous requests in  $T_{ISO}$  must satisfy

$$\frac{1}{125} \sum_{\forall T_i \in T_{ISO}} \frac{\lceil c_i/p_i \rceil + O}{s_i} \leq 0.8, \quad (1)$$

where  $O$  denotes the protocol overheads of an IEEE-1394 packet. The overheads include packet preambles, packet headers, and some other necessary fields. Note that the preceding formula could be used for the admission control of isochronous requests in QoS guarantees.

### 3.3 Probabilistic QoS Analysis for Asynchronous Transfers

The purpose of this section is to propose a probabilistic QoS analysis for asynchronous transfers. Since asynchronous requests will only be serviced after the satisfaction of isochronous requests in each bus cycle, the total utilization of the bus cycle should not be over 100%. Let  $T_{ASY} = \{T_{A_1}, T_{A_2}, \dots, T_{A_M}\}$  denote a set of (asynchronous) requests to devices of the asynchronous data transfer type. Each asynchronous request  $T_{A_i}$  could be modeled by three parameters:  $T_{A_i} = (\lambda_i, b_i, s_i)$  in  $T_{ASY}$ , where  $\lambda_i$  is the average arrival rate of its transfers,  $b_i$  is the average service time of each transfer, and  $s_i$  is the maximum transmission speed for the request. The following formula regarding the bus utilization should be satisfied; otherwise, it would be hard for asynchronous requests to receive QoS services:

$$\frac{1}{125} \sum_{\forall T_i \in T_{ISO}} \frac{\lceil c_i/p_i \rceil + O}{s_i} + \frac{1}{125} \sum_{\forall T_{A_j} \in T_{ASY}} \frac{\lambda_j \cdot (b_j + O)}{s_j} \leq 1 \quad (2)$$

Note that the previous formula could also be used for the admission control of asynchronous requests.

Once a node has performed the asynchronous transfer, it must not request the bus access until the next fairness interval, that is, the duration in a bus cycle for asynchronous requests, begins. The behavior for asynchronous transfers in the IEEE-1394 subsystem can be modeled as a limited polling system, where polling is a way in serving requests in a cyclic order with a generally nonzero reply interval. (Polling models [Takagi 1986; Takagi 2000] have been used to analyze the performance of a wide variety of interesting and important systems.) Figure 6 provides an illustration of a polling system. In Figure 6, there are  $N$  queues serviced in the system, where each queue is a list of requests with the same upper bound on the workload service time and has its own arrival distribution. The service-switching time from one queue to another is defined as a *reply interval* (also named as a *switch-over time* or a *walking time* in the literature).

An IEEE-1394 system serves the  $N$  queues (i.e.,  $Q_1, Q_2, \dots, Q_N$ ) with infinite buffer capacities in a cyclic manner. Whenever the server visits a queue, it only serves one request (if any is present). All reply intervals are independent, and the reply interval  $S_i$  between the  $i_{th}$  and  $(i+1)st$  queues ( $1 \leq i \leq N-1$ ) has the mean value  $r_i$  and variance  $\sigma_i^2$ . The total reply interval  $r$  for the servicing

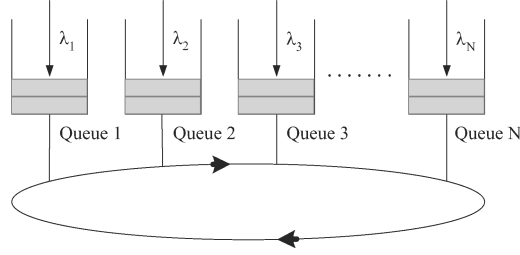


Fig. 6. A polling system for asynchronous transfers.

of all queues during a cycle is given by

$$r = \sum_{i=1}^N r_i.$$

Assume that requests arriving at each queue form a Poisson process, and the arrival rates for  $Q_1, Q_2, \dots, Q_N$  are  $\lambda_1, \lambda_2, \dots, \lambda_N$ , respectively. Provided that the requests arriving at  $Q_i$  are called type- $i$  requests, the service times of type- $i$  requests are independent and identically distributed with distribution function  $B_i(\cdot)$ , where  $B_i(\cdot)$  has the first moment  $b_i$  and second moment  $b_i^{(2)}$ .

Note that in the IEEE-1394 system, the service process is independent of the arrival and reply-interval processes. At  $Q_i$ , the utilization  $\rho_i$  can be defined as

$$\rho_i = \lambda_i b_i, \quad i = 1, 2, \dots, N.$$

The total utilization of the server,  $\rho$ , can be obtained as

$$\rho = \sum_{i=1}^N \rho_i,$$

where  $\rho < 1$  and  $\max(\lambda_1, \lambda_2, \dots, \lambda_N) \cdot r < 1 - \rho$  [Kuehn 1979]. The mean cycle time  $E_c$  for  $Q_i$  is given by

$$E_c = \frac{r}{1 - \rho}. \quad (3)$$

Based on the preceding discussions, the performance measure for our probabilistic analysis, namely the mean waiting time  $Ew_i$  for a type- $i$  request arrival, can be derived as follows. Before the derivations, the following variables are defined.

- $x_i$  denotes the queue length at  $Q_i$  (with the mean value  $Ex_i$ ) upon the arrival of the type- $i$  request.
- $c_i$  denotes a cycle time for  $Q_i$  with the mean value  $Ec_i$ .
- $rc_i$  denotes the residual time (with the mean value  $Eric_i$ ) between the time of the type- $i$  request arrival and the time that  $Q_i$  is served by the server.

The mean waiting time  $Ew_i$  of the type- $i$  request consists of two parts: a mean residual time  $Eric_i$  and the accumulative cycle times for servicing the previous  $Ex_i$  requests of  $Q_i$ . We have

$$Ew_i = Eric_i + Ex_i Ec_i. \quad (4)$$

Based on the fact that Poisson arrivals see time averages,  $Ex_i$  equals the mean value of waiting requests at  $Q_i$  at an arbitrary instant of time. By Little's law, we have

$$Ex_i = \lambda_i Ew_i. \quad (5)$$

From Eqs. (5) and (4),  $Ew_i$  can be derived as

$$Ew_i = \frac{Erc_i}{1 - \lambda_i Ec_i}. \quad (6)$$

Furthermore, from Boxma and Meister [1987],  $Ec_i$  in Eq. (6) can be approximately derived as

$$Ec_i = \frac{b_i + r}{1 - \rho + \rho_i}, \quad i = 1, 2, \dots, N, \quad (7)$$

and  $Erc_i$  can be approximately derived as

$$Erc_i = \frac{1 - \rho}{(1 - \rho)\rho + \sum_{j=1}^N \rho_j^2} \cdot \left[ \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j b_j^{(2)} + \frac{\rho}{2r} \sum_{j=1}^N \sigma_j^2 + \frac{r}{2(1 - \rho)} \sum_{j=1}^N \rho_j (1 + \rho_j) \right]. \quad (8)$$

Then based on (6), (7), and (8), we have

$$Ew_i \approx \frac{1 - \rho + \rho_i}{1 - \rho - \lambda_i r} \cdot \frac{1 - \rho}{(1 - \rho)\rho + \sum_{j=1}^N \rho_j^2} \cdot \left[ \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j b_j^{(2)} + \frac{\rho}{2r} \sum_{j=1}^N \sigma_j^2 + \frac{r}{2(1 - \rho)} \sum_{j=1}^N \rho_j (1 + \rho_j) \right]. \quad (9)$$

For the case when  $r = 0$  (i.e., the reply interval is not considered in the analysis), (9) can be simplified as [Boxma and Meister 1987]

$$Ew_i \approx \frac{1 - \rho + \rho_i}{(1 - \rho)\rho + \sum_{j=1}^N \rho_j^2} \cdot \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j b_j^{(2)}. \quad (10)$$

Finally, the mean response time  $E(T_i)$  for a type- $i$  request can be derived as follows.

$$E(T_i) \approx \frac{1 - \rho + \rho_i}{(1 - \rho)\rho + \sum_{j=1}^N \rho_j^2} \cdot \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j b_j^{(2)} + b_i \quad (11)$$

The average waiting time of each device can be evaluated by formula (11). The average waiting time can help to ascertain whether the bandwidth reservation could meet the QoS requirements of applications.

### 3.4 Topology Reconfiguration for Bandwidth Improvement

In the IEEE-1394 specification, data can be transmitted at one of three speeds: 100Mb/s, 200Mb/s, and 400Mb/s. The maximal speed in data transmission is

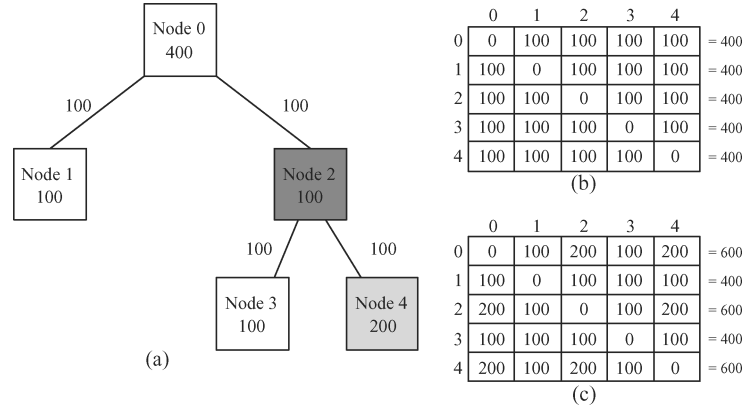


Fig. 7. An IEEE-1394 network topology.

limited to the minimal transmission speed among the devices along the transmission path. Consider an existing topology of IEEE-1394 devices. Let the topology be modeled by an acyclic undirected graph  $G = (V, E)$ , where each vertex in  $V$  denotes a device location in the topology, and each edge in  $E$  denotes a link between two connected vertices. The maximum speed of a path between  $v_i$  and  $v_j$ , denoted as  $p_{ij}$ , is defined as

$$\text{speed}(p_{ij}) = \min \{ \text{speed}(v_k) \mid \forall v_k \text{ in the path } p_{ij} \}, \quad (12)$$

where  $\text{speed}(v_k)$  is the transmission speed of  $v_k$ . The objective of this section is to propose an optimal algorithm to assign IEEE-1394 devices to locations (i.e., vertices of a topology) to maximize the transmission speed between any two devices (i.e.,  $\text{totTT}_G = \sum_{\forall p_{ij} \in S} \frac{1}{s(p_{ij})}$ ). The aforesaid optimization problem is referred to as the *location assignment problem* for the rest of this article.

Figure 7(a) shows an example location assignment of a tree-like IEEE-1394 network topology, where the transmission speed of each device is shown in its location box. In this example, there are three vertices (and devices) and  $\binom{3}{2} = 3$  transmission paths. The transmission speed between any two devices is 100. However, if we have a placement plan of the devices, as shown in Figure 7(b), then the transmission speed between devices  $d_0$  and  $d_1$  is increased to 200. Figure 7(c) shows another plan in which the device with the highest speed, namely  $d_1$ , is in the root. In this plan, we have a plan as good as that in Figure 7(b). One observation in this example is that we might have a good plan if we start the location assignment with devices having higher transmission speeds and then gradually grow outward by assigning other devices to the surrounding vertices.

Algorithm Max-Traffic-Flow assigns devices to vertices in a decreasing order of their transmission speeds. The idea is to gradually enlarge the group of devices assigned to the topology one at a time, provided that the newly included device would not slow down the transmission speed between any two devices already assigned in the group. The devices in  $D$  are first sorted and renumbered

**Algorithm 1.** Topology Reconfiguration

---

```

1: Max-Traffic-Flow(G,D)
2:  $G$  : an acyclic undirected graph  $(V, E)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  and  $n = |V|$ ;
3:  $D$  : a set of devices;
4: {
5: Sort and renumber all devices in  $D = \{d_1, d_2, \dots, d_n\}$  in a decreasing
   order by their transmission speeds;
6: Assign  $d_1$  to a vertex  $v_1$  in  $V$ ;          /*  $d_1$  is with the largest transfer speed
   */
7:  $S_1 \leftarrow \{v_1\}$ ;
8:  $S_2 \leftarrow (V - S_1)$ ;
9: for ( $i = 2; i \leq n; i++$ ) do
10:  Pick up any vertex  $v_x \in S_1$  that has a neighbor  $v_y$  in  $G$ , where  $v_y$  is any one
     in  $S_2$ ;
11:  Assign  $d_i$  to vertex  $v_y$ .
12:   $S_1 \leftarrow S_1 \cup \{v_y\}$ ;
13:  Remove  $v_y$  from  $S_2$ ;
14: end for
15: return  $S_1$ ;
16: }
```

---

in a decreasing order of their transmission speeds (step 5). Initially, only the device with the highest transmission speed is in the group, namely  $S_1$  (step 7). Each iteration in the loop between step 9s and 14 moves one device with the highest transmission speed in the remaining devices to  $S_1$ , where the device is assigned to a vertex outreached by a vertex assigned to a device in  $S_1$ . In this way, the transmission speed between any two devices in  $S_1$  would not be affected, and the maximum transmission speed between the newly included device and any device in  $S_1$  is only constrained by the transmission speed of the newly included device.

**THEOREM 3.1.** *Algorithm Max-Traffic-Flow always derives an optimal solution for the location assignment problem.*

**PROOF.** The proof could be done by induction: Consider a given topology  $G = (V, E)$  and a set of devices  $D = \{d_1, d_2, \dots, d_n\}$ , where devices in  $D$  are sorted and renumbered in decreasing order of their transmission speeds. For the induction base, devices  $d_1$  and  $d_2$  are assigned to two neighboring vertices. The transmission speed between these two devices is optimized because there are only two devices, and these are the devices with the top two highest speeds. In the induction hypothesis, suppose that the algorithm still derives an optimal solution for the location assignment problem for the first  $K$  devices in  $D$ , where  $2 < K \leq (|V| - 1)$ . For the induction step, let the next device in  $D$  (i.e.,  $d_{K+1}$ ) be included by the algorithm. Since the device has the smallest transmission speed among the first  $(K + 1)$  devices in  $D$ , the vertex assignment would not change the maximum transmission speed between any two devices among the first  $K$  devices in  $D$ . The inclusion of the new device would still be an optimal solution for the location assignment problem for the first  $(K + 1)$  devices in  $D$ .  $\square$

Table I. Experimental Workloads (Excluding Protocol Overheads)

	Isochronous Requests			Asynchronous Requests	
	Digital Video	Web Camera1	Web Camera2	Hard Drive1	Hard Drive2
Payload Size (bytes)	2304	1152	576	2046	2046
Polling Period ( $\mu s$ )	125	125	125	sporadic	sporadic
Bandwidth Utilization or Reservation	37.5% (Reservation)	18.75% (Reservation)	9.375% (Reservation)	34.38% (Maximum)	34.38% (Maximum)

#### 4. PERFORMANCE EVALUATION

This section is meant to access the capability of the proposed methodology in QoS guarantees for isochronous and asynchronous requests. The experiments consisted of two parts: In the first part, we evaluated the effectiveness of the proposed methodology in QoS guarantees of isochronous requests when realistic workloads were considered. In the second part, Eq. (10) was evaluated in terms of its capability in providing probabilistic QoS analysis for asynchronous requests.

##### 4.1 Bandwidth Reservation: Isochronous Requests

*4.1.1 Experiment Setup and Metrics.* The effectiveness of the proposed methodology in the QoS guarantees was evaluated over a realistic system prototype implemented over Linux (with kernel 2.6.10) and IEEE-1394 OHCI host controllers. The experimental platforms were personal computers equipped with Celeron 2.4GHz and IEEE-1394 devices. The primary performance metric of the experiments was the satisfaction of the bus bandwidth reservation of the isochronous requests of each device. This is the difference between the reserved bus bandwidth of a device and the bus bandwidth received by the device in the experiments.

The experimental workloads were generated by devices (available in the market), and the maximum bus utilization of the requests in the experimental workloads was over 100%: One IEEE-1394 digital video camera (referred to as DV) and two IEEE-1394 Web cameras (referred to as WebCam1 and WebCam2) were used to generate periodic requests of isochronous transfers to evaluate the capability of the proposed methodology in QoS reservations. Asynchronous transfers were generated by two IEEE-1394 hard drives (referred to as HD1 and HD2) to create interference in the service of isochronous requests. The characteristics of the devices are summarized in Table I.

Let the IEEE-1394 digital video camera reserve 37.5% of the total bus bandwidth, and the first and second IEEE-1394 Web cameras reserve 18.75% and 9.37% of the total bus bandwidth, respectively. The percentages of bandwidth reservations were derived based on the payload sizes and polling periods of devices. Suppose that the maximum bus utilization of the two hard disks could be up to 34.38% of the total bus bandwidth. Although the maximum bus utilization of the aforesaid devices was only about 100%, the real bus utilization could be over 100% in reality because the preceding figures did not include protocol overheads. Note that the bandwidth reservations of the IEEE-1394 digital video

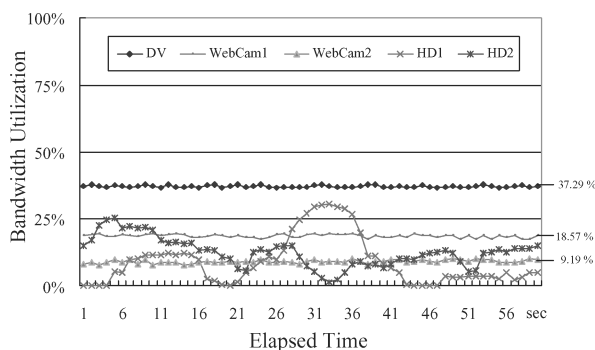


Fig. 8. The bandwidth utilization of one digital video camera, two Web cameras, and two hard drives.

camera and two IEEE-1394 Web cameras would be used up by real requests. The workloads of the two hard disks were based on a real trace collected in file access and music playing. Each experiment was run for 600 s.

**4.1.2 Experimental Results.** Figure 8 shows the bandwidth utilization of one digital video camera, two Web cameras, and two hard drives for the first 60 s in the experiment. The experimental results for the remaining 540 s were not included because they were similar to those shown in Figure 8. The x-axis denotes the time elapsed so far in s, and the y-axis denotes the percentage of IEEE-1394 bus bandwidth utilization for each IEEE-1394 device. It is shown that the bandwidth reservations of the digital video camera and two Web cameras were met without any problem, even though we did observe a significant amount of workload fluctuation of the two hard drives in the experiments. Note that the QoS guarantees of the proposed methodology were fulfilled, even though the total bus utilization of requests sometimes went over 100%.

#### 4.2 Probabilistic QoS Analysis: Asynchronous Requests

The purpose of this section is to evaluate the analytic framework, namely that based on Eq. (10), in providing probabilistic QoS analysis of asynchronous requests. There were 10 queues for asynchronous requests. Each transfer of an asynchronous request was of a size randomly selected between 1B and 2500B. Two types of arrival-rate patterns were adopted for the evaluation: (1) Each queue has the same arrival rate; and (2) each queue has a different arrival rate.

When every queue has the same arrival rate, this rate ranges from 1 transfer per bus cycle (i.e.,  $125\mu\text{s}$ ) to 9 transfers per bus cycle. Since there were 10 queues, the maximum workload of asynchronous requests is on average, 1125B of asynchronous transfers per bus cycle. Figure 9 shows the average waiting time of a request based on Eq. (10) (referred to in the figure as the Analytic Results) and those observed in the experiments (referred to as the Experimental Results). The x-axis denotes the arrival rate in a unit equal to 0.01 transfer per bus cycle. The y-axis denotes the average waiting time in bus cycles. As shown in Figure 9, the analytic results were very close to those observed in the experiments.

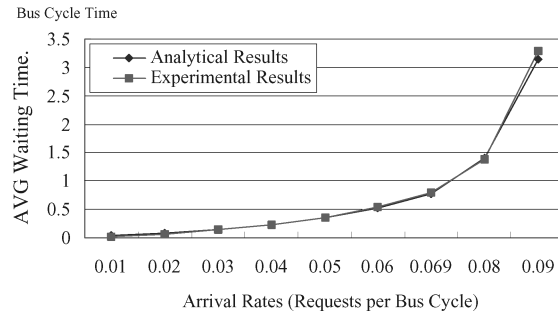


Fig. 9. All of the queues have the same arrival rate.

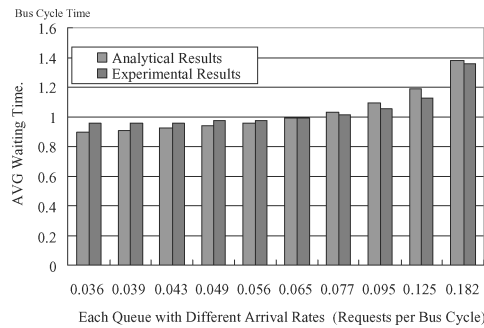


Fig. 10. Different queues have different arrival rates.

When every queue has a different arrival rate, the interval arrival time of transfers to each queue ranges from 5 bus cycles (i.e.,  $625\mu s$ ) to 27 bus cycles (i.e.,  $3375\mu s$ ). In other words, the average arrival rate ranges from 0.037 transfers per bus cycle to 0.2 transfers per bus cycle. The total arrival rate of 10 queues consists of about 79.5 transfers per bus cycle. The maximum workload for asynchronous requests is about 959B of asynchronous transfers, on average, per cycle. As shown in Figure 10, the analytic results are also very close to those observed in the experiments, where each mark in the x-axis denotes a queue, and the y-axis denotes the average waiting time in bus cycles.

## 5. CONCLUSION

Delivering QoS guarantees to applications is the responsibility of not only the system kernel, but also all of the subsystems involved in the servicing of the applications. The objective of this research has been to explore the design and implementation of QoS supports for IEEE-1394 device management. Different from past related work, we aimed at resource reservation for requests and their analytic framework for QoS guarantees. In this article, a bandwidth reservation algorithm has been proposed to reserve available IEEE-1394 bandwidth for devices of the isochronous transfer type which need different attention from the system. An integrated real-time driver architecture was presented that complies with IEEE-1394 specifications. We then considered the mixture of isochronous and asynchronous requests in admission control for QoS services.



An analytic framework for probability-based (soft) QoS guarantees was then proposed for asynchronous requests, since asynchronous requests are serviced in a best-effort fashion over IEEE-1394. An optimal topology configuration algorithm has been proposed for IEEE-1394 devices to better utilize the bandwidth of an IEEE-1394 bus. The capability of the proposed methodology and the analytic framework were evaluated and demonstrated by a series of experiments over a Linux-based system prototype.

For future research, we shall further extend the work to QoS supports over converged heterogeneous networks/buses, such as Ethernet, wireless LAN, IEEE-1394, etc. The problem shall become even more challenging when different technologies have different characteristics and constraints in data transmission.

#### ACKNOWLEDGMENTS

This work is supported in part by research grants from the RoC National Science Council under Grants NSC 93-2752-E-002-008-PAE, 94-2213-E-002-007, and 96-2219-E-002-004.

#### REFERENCES

- BOXMA, O. AND MEISTER, B. 1987. Waiting-Time approximations in multi-queue systems with cyclic service. In *Perform. Eval.* 7, 1, 59–70.
- HUANG, C., CHANG, L., AND KUO, T. 2003. A cyclic-executive-based QoS guarantee over USB. In *Proceedings of the IEEE 9th Real-Time and Embedded Technology and Applications Symposium (RTAS'03)*, 88–95.
- HUANG, C., KUO, T., AND PANG, A. 2004. QoS support for USB 2.0 periodic and sporadic device requests. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS'04)*, 395–404.
- ISO/IEC. 2009. ISO/IEC 13213 ANSI/IEEE std 1212 control and status registers (CSR) architecture for microcomputer busses .
- IEEE. 1996. IEEE std. 1394-1995, IEEE std 1394a-2000, IEEE standard for a high performance serial bus. IEEE Computer Society.
- IGARASHI, T., HAYAKAWA, K., NISHIMURA, T., OZAWA, T., AND TAKIZUKA, H. 1999. Home network file system for home network based on IEEE-1394 technology. *IEEE Trans. Consumer Electron.* 45, 1000–1003.
- KU, T. Y., PARK, D. H., AND MOON, K. D. 2002. A Java-based home network middleware architecture supporting IEEE1394 and TCP/IP. *IEEE Trans. Consumer Electron.* 48, 496–504.
- KUEHN, P. J. 1979. LMultiqueue systems with nonexhaustive cyclic service. *The Bell Syst. Tech. J.* 58, 3, 671–698.
- LIM, H., PARK, D., KANG, S., AND OH, B. 2000. Priority queue-based IEEE1394 device driver supporting real-time characteristics. *IEEE Trans. Consumer Electron.* 46, 825–833.
- PARK, D. AND KANG, S. 2000. IEEE1394 OHCI device driver architecture for guarantee real-time requirement. In *Proceedings of 7th International Conference on Real-Time Computing Systems and Applications*, 389–393.
- SAITO, T., TOMODA, I., TAKABATAKE, Y., ARNI, J., AND TERAMOTO, K. 2000. Home gateway architecture and its implementation. *IEEE Trans. Consumer Electron.* 46, 1161–1166.
- TAKAGI, H. 1986. *Analysis of Polling System*. MIT Press series.
- TAKAGI, H. 2000. Queuing analysis of polling models. *ACM Comput. Surv.* 20,
- ZHANG, H., UDAGAWA, T., ARITA, T., TSUJI, J., OKADA, K., SASASE, I., AND NAKAGAWA, M. 2001. Wireless 1394: A new standard for integrated wireless broadband home networking. In *Proceedings of the IEEE VTS 53rd Vehicular Technology Conference (VTC) 2*, 1124–1128.

Received July 2006; revised November 2008; accepted November 2008