ATM Input-Buffered Switches with the Guaranteed-Rate Property

A. Hung
Newbridge Networks Corp.
600 March Road
Kanata, ON, Canada K2K 2E6
anthonyh@newbridge.com

G. Kesidis
E & CE Dept
University of Waterloo
Waterloo, ON, Canada N2L 3G1
g.kesidis@eandce.uwaterloo.ca

N. McKeown
Elec. Eng. Dept
Stanford University
Stanford, CA 94305
nickm@ee.stanford.edu

Abstract— There is considerable interest in the provision of guaranteed-rate services for IP and ATM networks. Simultaneously, bandwidth demands make input-buffered architectures attractive, and in some cases, necessary. In this paper, we consider the problem of how to support guaranteed-rate services in a single-stage, input-buffered switch suitable for a LAN switch, an ATM switch or an IP router. Such a switch must be feasible at high transmission speeds, offering both guaranteed-rate performance for CBR channels (e.g. for real-time connections) and best-effort services for traditional data traffic. We consider a switch scheduling mechanism that employs idling hierarchical round-robin (HRR) scheduling and fabric arbitration at the connection-level for guaranteed-rate service using the Slepian-Duguid algorithm. The switch uses cell level arbitration for best-effort service. This overall switch scheduling mechanism is a variation of DEC's AN2 design [2].

I. Introduction

There is a strong desire to support a guaranteed-rate service for both IP and ATM networks, in particular for real-time traffic. For the Internet, the IETF is pursuing a guaranteed-rate service [23] based on generalized processor sharing [21]. For ATM networks, the ATM Forum has proposed a CBR service described in [4].

In this paper, we consider the provision of a guaranteed-rate service on a network switch or router. In particular, we consider the support of this service over a switching fabric that employs *input-buffering.* Whereas many researchers have studied provisioning of guaranteed-rate services over output-buffered switches, there has been little work reported for input-buffered switches. Interest is growing in input-buffering: today's memory bandwidths cannot keep up with the demand for faster line-rates and greater switching capacity. And the problem is getting worse: memory access times are barely improving, whereas the demand for switching capacity continues to grow exponentially.

As a result, many of the fastest commercial [2], [3] and research [19], [22] switches and routers today are based on an input-buffered crossbar switch.[1] Each of these systems internally uses a small fixed-size packet, similar or equal in length to a 53-byte ATM cell. Each system contains line cards that accept variable length packets from the outside world. When all of the cells have been switched, they are reassembled into variable length packets before being sent on their way. Because of their widespread use, we focus our attention on switches that use fixed-size packets. For obvious reasons, we refer to them as "cells" but make no assumptions about their (fixed) length.

In this paper, we address the problem of scheduling bandwidth for input-buffered switches. We assume that the network provides (perhaps renegotiable) CBR virtual channels for real-time services. We focus on how an input-buffered switch can provide a guaranteed-rate service. In addition, the switch must support a "best-effort" service for traditional data traffic. Other services can be supported over these two "basic" services.

A 2×2, single-stage, input-buffered switch is illustrated in Figure 1. By "$N \times N$" we mean that the switch has N input ports[2] and N output ports. The switch operates on a "cell-time" clock where one cell-time is the (common) transmission time of a cell on the links connected to the switch, e.g., at 155 Mbps, one cell-time is approximately $2.8\mu s$. At most one cell arrives at each input port every cell-time and, similarly, at most one cell departs from each output port every cell-time.

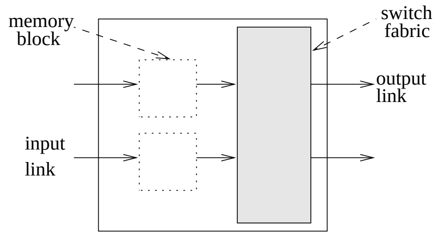

Fig. 1. A 2×2, Single-Stage, Input-Buffered Switch

In general for single-stage switches, there is an input-side switch fabric between the input ports and the single

G. Kesidis is supported by the NSERC of Canada.

[1] The main alternative architectures to single-stage input-queued switches are multi-stage switches, e.g., [7].

[2] For simplicity, we will associate a single link to each port.

queueing stage, and there is an output-side switch fabric between the queueing stage and the output ports. The queueing stage is simply a bank of logically separate queues. The queues are distributed among "blocks" of memory where each memory block has a separate input/output bus and, therefore, can operate independently from the other blocks of memory. We also assume each memory block has a single address decoder allowing only one read or write operation at a time. Each logical queue is served in a first-in-first-out (FIFO) fashion. Both the number of queues and the number of blocks of memory can be different from N.

Each memory block of a single-stage, input-buffered switch has a single associated input port. So, each memory block will experience at most one cell write operation per cell-time. In the scope of this paper, the input-side fabric merely determines where in buffer memory each arriving cell is written and is not illustrated in Figure 1. The output-side fabric has an associated "arbiter". At each cell-time, the arbiter decides which cells (at most N in total) from the memory blocks traverse the fabric and are transmitted onto the output links.

Every cell-time, the input-side switch fabric simultaneously removes (at most N) cells from the input ports and places them in the queueing stage. Similarly, every cell-time the output-side switch fabric simultaneously removes (at most N) cells from the queueing stage and places them in output ports for transmission onto the output links. We assume that the switch fabrics are nonblocking, i.e., cells are never dropped while passing "through" a fabric. On the other hand, a cell may be dropped by the queueing stage if, for example, it arrives to a full queue.

The particular queue visited by a cell is determined by its input port and the address field in its header. For example, an ATM switch has a look-up table mapping (input-port, VPI/VCI) to the appropriate output port for each cell. This look-up table is modified at call set-up and termination. An IP router supporting RSVP [24] has a routing table mapping (destination IP address) to the appropriate output port, and a table mapping an IP flow to the appropriate queue. A 2×2 input-buffered switch that uses a separate queue for *all* traffic with the same (input link, output link) combination is depicted Figure 2.

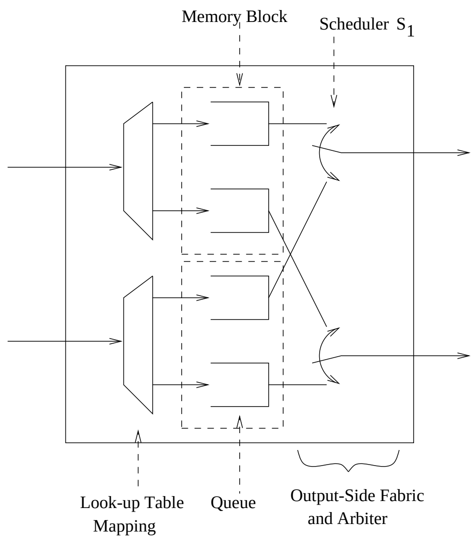

Fig. 2. A "Simple" 2×2 Single-Stage, Input-Buffered Switch

In this paper, we will observe the following general design goals:

- O(N) cell memory blocks, i.e., scalability in the number of memory blocks.
- Minimal cell memory bandwidth requirement.
- Minimal amount of scheduling computation per cell.
- Guaranteed-rate performance.

Without the O(N) condition on the number of memory blocks we could choose to have N^2 memory blocks: one for each (input port, output port) pair. This would create a switch that is *both* input-buffered and output-buffered: output-buffered switches have a simple guaranteed-rate property, see [14] for example. Under the condition of O(N) memory blocks, however, the speed of operation of output-buffered switches is severely limited by memory bandwidth.

Practical prioritized input-buffered switches are considered in, e.g., [16], [18], [15]. DEC's AN2 switch [2] goes further and employs "connection-level[3] arbitration" for CBR flows. The connection-level arbitration uses an idling weighted round-robin (WRR) scheduling mechanism with a common frame size. The precomputed schedule is based on the Slepian-Duguid algorithm, see Chapter 3 of [10]. In DEC's AN2 switch, best-effort traffic is supported by a separate "cell-level arbiter".

In this paper, we revisit DEC's switch design. We describe a per-connection version of this switch that uses idling hierarchical round-robin (HRR) schedulers [13], [12]. We believe that idling round-robin schedulers are an efficient way to support guaranteed-rate service with minimal per-cell scheduling computation. Moreover, they allow for a controllable distribution of excess (unused or unreserved) capacity to achieve given "fairness" criteria.

In Section 2, the traffic and queueing variables are defined for the input-buffered switch. In Section 3, connection-level arbitration for CBR flows is described. The handling of best-effort traffic is described in Section 4. A summary is given in Section 5.

[3] In [2], the CBR arbitration is said to be "precomputed".

II. Scheduling and Memory Management

Memory operations may limit the speed of operation of a switch. So, in input-buffered switches, all memory blocks are restricted to one cell read operation per cell-time. Consistent with the switch design goals stated above, we assume that there is just one memory block per input port processor (IPP). In this case, we may have *contention* at each input port among flows that wish to connect to different output ports. This contention is resolved by the bandwidth schedulers situated at the IPPs as we will see below.

Consider an $N \times N$, single-stage, input-buffered switch handling traffic that is classified into several *priorities.* In the first priority (priority-1) are connections that require bandwidth guarantees from the switch. Connections that have best-effort varieties of service belong to subsequent priorities. In the following, we will focus on the handling of priority-1 connections. Best-effort flows are considered in Section 5.

Let $\rho^k_{i,j}$ be the bandwidth allotment of the k^{th} priority-1 connection which flows from the i^{th} input link to the j^{th} output link where $1 \leq k \leq K_{i,j}$; let the cell arrival-times process of this connection be $\mathbf{a}^k_{i,j}$. We assume that there was *no overbooking* on the input links, i.e.,

$$\sum_{j=1}^{N} \sum_{k=1}^{K_{i,j}} \rho^k_{i,j} \quad \leq \quad 1 \quad \text{for all } i, \tag{1}$$

and no overbooking on the output links,

$$\sum_{i=1}^{N} \sum_{k=1}^{K_{i,j}} \rho^k_{i,j} \quad \leq \quad 1 \quad \text{for all } j. \tag{2}$$

In the proposed, there is a scheduler situated at each IPP. We denote the scheduler at the i^{th} IPP by S_i. Each cell-time, every scheduler S_i chooses a cell from its associated memory block to transmit through the (output-side) switch fabric.

A. Virtual Output Queueing (VOQ) versus Per-VC Queueing

First suppose that, for all i, j, all the priority-1 connections from input port i to output port j use a single FIFO queue, $\hat{Q}_{i,j}$, with aggregate bandwidth allotment

$$\hat{\rho}_{i,j} \quad = \quad \sum_{k=1}^{K_{i,j}} \rho^k_{i,j}.$$

That is, the cell arrival-times process to $\hat{Q}_{i,j}$ is

$$\hat{\mathbf{a}}_{i,j} \quad = \quad \bigcup_{k=1}^{K_{i,j}} \mathbf{a}^k_{i,j}.$$

In an input-buffered switch, aggregating the flows in this manner is called "virtual output queueing" (VOQ) or "per-output-port queueing" [1]. Under VOQ, the S_i's are idling weighted round robin (WRR, i.e., single-frame HRR) schedulers with *common* frame size. The slot assignments (output-port indexes) of S_i are based on the aggregate bandwidth allotments $\{\hat{\rho}_{i,j} \ : \ 1 \leq j \leq N\}$, c.f., Section 3. VOQ is the "CBR" scheme used in DEC's AN2 switch [2].

With VOQ, cell "head-of-line blocking" can be eliminated entirely [17]. We will see that under the proposed architecture, the term "VOQ" is especially appropriate because the switch behaves like an output-buffered switch.

In per-connection (a.k.a. per-virtual-channel or "per-VC") memory management, there is a separate FIFO queue, $Q^k_{i,j}$, handling each flow $\mathbf{a}^k_{i,j}$. In this case, for all i, S_i is an idling, multiple-branch HRR scheduler [13], [12] handling queues

$$\{Q^k_{i,j} \ : \ 1 \leq j \leq N, \ 1 \leq k \leq K_{i,j}\}.$$

The level-one frame of S_i under per-VC management is *identical* to the that of the idling WRR S_i under VOQ. The j^{th} branch of per-VC S_i's structure resolves the bandwidth $\hat{\rho}_{i,j}$ into $\{\rho^k_{i,j} \ : 1 \leq k \leq K_{i,j}\}$ for $\{Q^k_{i,j} \ : \ 1 \leq k \leq K_{i,j}\}$. An example frame structure is given in Figure 3 below.

Best-effort flows can be separated into FIFO queues according to cell (input port, output port) pair under per-VC Queueing or VOQ. So, under VOQ, each input port has $2N$ associated FIFO queues: N for priority-1 flows and N for best-effort flows. Under per-VC Queueing, each input port has N associated FIFO queues for best-effort flows (c.f., Section 4) and a potentially large number ($\sum_{j=1}^{N} K_{i,j}$) of logical FIFO queues for priority-1 flows.

III. Fabric Arbitration for Priority-1 Service

Recall that the idling WRR schedulers of VOQ and the level-one frames of the idling HRR schedulers of per-VC Queueing partition bandwidth according to output-port indexes. The slot assignments of the S_i's must be coordinated so that no two of them choose the same output port in any given cell-time. This coordination is called "contention resolution" or "fabric arbitration".

For simplicity, consider VOQ. At any given time, all the S_i have a common frame size of f slots (cells). So, there will be $r_{i,j} := \lceil \hat{\rho}_{i,j} f \rceil$ slots reserved for the priority-1 flows to j^{th} output port in each frame of S_i. Thus, S_i has $f - \sum_{j=1}^{N} r_{i,j}$ slots that are unreserved and intended for best-effort flows, c.f., Section 4. So, the stronger "no overbooking" conditions are

$$\sum_{j=1}^{N} r_{i,j} \le f \text{ for all } i \quad \text{and} \quad \sum_{i=1}^{N} r_{i,j} \le f \text{ for all } j. \quad (3)$$

An $N \times f$ "slot assignment matrix" for the level-one frames of all of the S_i schedulers can now be defined. No column of this matrix contains the same numeral more than once; as these numerals correspond to output ports, cell "collisions" at the output ports will consequently not occur. Also, the number of slots assigned to output port j in row i (i.e., in the level-one frame of S_i) is $r_{i,j}$. Let R be the $N \times N$ matrix whose $(i,j)^{\text{th}}$ entry is $r_{i,j}$. Under the "no overbooking" conditions (3), determining such an $N \times f$ slot assignment matrix given R and f is the *priority-1 fabric arbitration problem.*

For example, consider the case of a 3×3 switch which, at some given time, has $f = 6$ and

$$R = \begin{pmatrix} 2 & 3 & 0 \\ 1 & 0 & 3 \\ 2 & 1 & 2 \end{pmatrix}$$

A 3×6 slot assignment matrix is:

	1	1	2	2	2	S_1
1			3	3	3	S_2
2	3	3	1	1		S_3

Note that the "blanks" in the slot assignment matrix represent unreserved slots that may potentially be used for best-effort cells, c.f., Section 5.

The priority-1 fabric arbitration problem can generally be solved by applying the Slepian-Duguid approach for a circuit-switched Clos network, see Chapter 3 of [10]. Using this approach, the entire slot assignment matrix takes $O(N^2 f)$ time to calculate, which can be a significant computational expense at high ATM speeds. Thus, this computation cannot occur at the cell level. Modifications of bandwidth allotments, priority-1 slot assignments or frame structures would occur at the connection level. In response to changing priority-1 traffic demands, the slot assignment matrix would be only *periodically* modified where the period between modifications clearly depends on the speed of the implemented Slepian-Duguid algorithm.

A. An Example Per-VC Frame Structure

Consider a 2×2 switch handling two priority-1 connections for each (input port, output port) pair. The level-one frame size is $f = 5$ and the bandwidth allotments in this example are:

$$(\rho_{1,1}^1, \rho_{1,1}^2) = (0.1, 0.1), \qquad (\rho_{1,2}^1, \rho_{1,2}^2) = (0.2, 0.2),$$
$$(\rho_{2,1}^1, \rho_{2,1}^2) = (0.1, 0.3), \quad \text{and} \quad (\rho_{2,2}^1, \rho_{2,2}^2) = (0.1, 0.1).$$

A set of corresponding multiple-branch HRR frame structures are given in Figure 3.

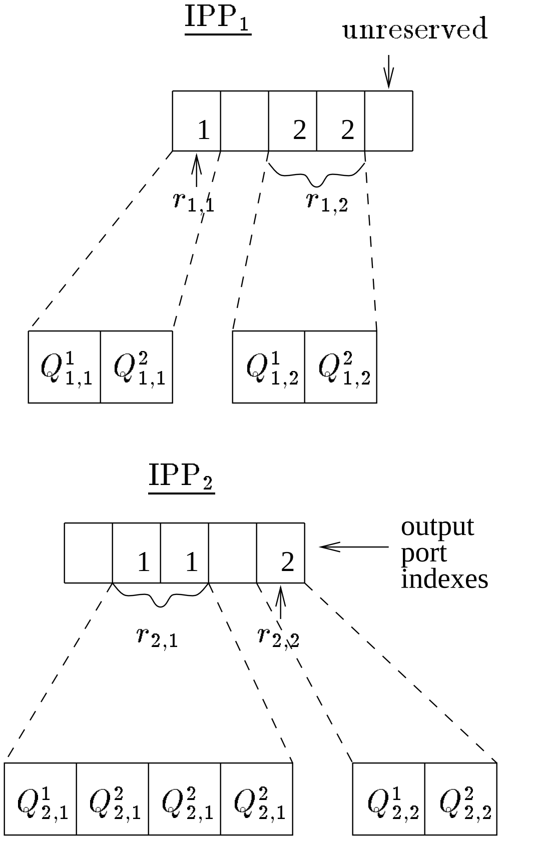

Fig. 3. Example HRR Frame Structures for a 2×2 Switch

B. Guaranteed-Rate Performance and Buffer Sizing

The guaranteed-rate performance of multilevel-assignment HRR is given in [12], [11], [14]. Buffer sizing rules are given in Section 4.2 of [14] (see also [9]). Results for queues in either idling or nonidling mode are available. For a VOQ switch, end-to-end buffer sizing can be obtained from the results in Section 4.5 of [14].

When the HRR frame structures or level-1 slot assignment positions are modified to accommodate a different traffic load, the cells of active connections may experience some additional delay jitter [10], [13]. The guaranteed-rate parameters roughly double to account for this [11], [14].

IV. Supporting Best-Effort Traffic

Recall that S_i has $f - \sum_{j=1}^{N} r_{i,j}$ slots that are unreserved; these slots are represented as blanks in the slot assignment matrix. Best-effort cells can be served in unreserved or reserved-but-unused slots. A priority-1 queue's reserved slot is unused when that queue is idle. So, a certain number of input ports and output ports may be unmatched by the priority-1 arbiter. A cell-level arbiter may be used to match these "left over" input and output ports for best-effort cells. In [2], a (randomized) parallel iterative match (PIM) is suggested; however, *i*SLIP [16] may also be used and has certain performance advantages. See Section 3.4 of [5] or [18] for discussions of cell-level arbiters for input-buffered switches. The best-effort cell-level arbitration may be governed by a "flow control" entity and related "fairness" considerations, as mandated in Section 5.2 of [4].

A concern of cell-level arbitration is that the required signaling each cell-time among OPPs and IPPs is costly. An alternative would be to divide the unreserved slots among best-effort flows just prior to the connection-level priority-1 arbitration process (without violating the "no overbooking" conditions). Clearly, this would result in smaller aggregate throughput of best-effort traffic compared to the "fully-shared" approach based on cell-level arbitration.

In general, best-effort service can be accommodated by adding a priority indication to the queues involved, see Section 2.2.2 of [6]. That is, best-effort FIFO queues would be assigned priorities from the set $\{2, 3, 4...\}$ with priority-1 indicating queues with bandwidth guarantees.

For example, we can consider a two-priority switch [8] handling:

- connections requiring bandwidth guarantees with priority 1
- IP data traffic with priority 2

For both VOQ and per-VC Queueing, we can arrange each IPP to have N priority-2 best-effort FIFO queues: one for each switch output port. Note how the use of *idling* round-robin scheduling allows the switch to *control* how excess capacity (unused or unreserved slots) is distributed among the queues of an IPP to achieve given fairness criteria.

V. Summary

We have presented a method for supporting guaranteed-rate service in a single-stage, input-buffered switch. Idling multiple-branch HRR schedulers were employed for per-connection, guaranteed-rate management or idling WRR schedulers were employed for per-output-port queueing (VOQ) guaranteed-rate management. The problem of connection-level guaranteed-rate fabric arbitration can be readily solved using the Slepian-Duguid method. Best-effort traffic is "squeezed into" unreserved or unused time slots, under the control of a centralized cell-arbiter. VOQ is basically DEC's AN2 switch design [2]. The guaranteed-rate performance and buffer sizing rules are available, [12], [11], [14]. The described switch is the only *input*-buffered, single-stage switch with a guaranteed-rate property. The guaranteed-rate property enables CBR service for real-time connections in particular.

References

[1] M.K.M. Ali and M. Youssefi. The performance of an input access scheme in a high-speed packet switch. In *Proc. IEEE INFOCOM*, Miami, FL, pages 454-461, Apr. 1991.

[2] T. Anderson, S. Owicki, J. Saxe and C. Thacker. High speed switch scheduling for local area networks. *ACM Trans. on Computer Systems*, pages 319-352, Nov. 1993.

[3] Ascend GRF 400 Overview and Features http://www.ascend.com.au/products/grf400

[4] ATM Forum Technical Committee. Traffic Management Specification Version 4.0. Technical Report No. 95-0013R2, Draft version 3.0, April 1995.

[5] R.Y. Awdeh and H.T. Mouftah. Survey of ATM switch architectures. *Computer Networks and ISDN Systems*, Vol. 27, pages 1567-1613, 1995.

[6] E. Basturk, A. Birman, G. Delp, R. Guerin, R. Haas, S. Kamat, D. Kandlur, P. Pan, D. Pendarakis, R. Rajan, D. Saha and D. Williams, Design and Implementation of a QoS Capable Switch-Router. Technical Report no. RC 20848, IBM Research, Jan. 31 1997.

[7] T. Chaney, J.A. Fingerhut, M. Flucke and J. Turner, Design of a Gigabit ATM Switch. In *Proc. IEEE INFOCOM*, Kobe, April 1997.

[8] J.S.-C. Chen and R. Guerin, Input queueing of internally nonblocking switch with two priority classes. In *Proc. IEEE INFOCOM*, pages 529-537, 1989.

[9] R. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE JSAC*, Vol. 13, No. 6:pages 1048–1056, Aug. 1995.

[10] J. Hui. *Switching and Traffic Theory for Integrated Broadband Networks.* Kluwer Academic Publishers, Boston, MA, 1990.

[11] A. Hung, Bandwidth Scheduling and its Application in ATM Networks. Ph.D. Thesis, E&CE Dept, University of Waterloo, July 1997.

[12] A. Hung and G. Kesidis. Performance evaluation of hierarchical round-robin bandwidth scheduling for ATM. In *Proc. ITC-15, Washington, DC*, pp. 1247-1256, June 1997.

[13] C.R. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks. In *Proc. IEEE Globecom*, 1990.

[14] G. Kesidis. *ATM Network Performance.* Second Edition, Kluwer Academic Publishers, Boston, MA, 1998.

[15] C. Lund, S. Phillips, and N. Reingold. Fair prioritized scheduling in an input-buffered switch. In *Proc. Broadband Communications, Montreal*, 1996.

[16] N. McKeown. Scheduling Cells in an Input-Queued Cell Switch. Ph.D. Thesis, University of California, Berkeley. May 1995.

[17] N. McKeown, V. Anantharam and J. Walrand. Achieving 100% Throughput in an Input-Queued Switch. *Proc. of IEEE Infocom, p. 296-302 vol.1, March 1996.*

[18] N. McKeown. The SLIP Scheduling Algorithm for Input-Queued Switches. *preprint*, 1997.

[19] N. McKeown, M. Izzard et al. Tiny Tera: A Packet Switch Core. IEEE Micro Magazine, Jan-Feb 1997, pp.26-33.

[20] P. Newman, G. Minshall, T. Lyon, and L. Huston. IP switching and gigabit routers. *IEEE Comm. Mag.*, Vol. 35, No.1:pp. 64–69, Jan. 1997.

[21] A.K. Parekh and R.G. Gallager, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case, *IEEE/ACM Trans. Networking,* Vol. 1, No. 3, June 1993, pages 344-357.

[22] C. Partridge et al. A Fifty Gigabit per Second IP Router. *Submitted for publication.*, 1997.

[23] S. Shenker, C. Partridge and R. Guerin. Specification of Guaranteed Quality of Service. Internet Draft draft-ietf-intserv-guaranteed-svc-07.txt, Feb 1997.

[24] L. Zhang, S. Deering, D. Estrin, S. Shenker et al. RSVP: a new resource ReSerVation Protocol. IEEE Network Mag., Sept. 1993, vol.7, (no.5): p. 8-18.