AN APPROXIMATION TO RATE-EQUALIZATION FAIRNESS WITH LOGARITHMIC COMPLEXITY

Suparn Gupta, Masters

The University of Texas at Dallas, 2014

Supervising Professor: Jorge Arturo Cobb

Scheduling protocols that provide both rate and fairness guarantees, such as Weighted Fair Queuing (WFQ) distributes the unused bandwith among flows in proportion to their reserved rate. Flows with higher reservation rate thus receive a larger share of bandwith than those with lesser reserved rates.

In an earlier work [30], a new approach to allocate unused bandwidth known as rate-equalization (REQ) was proposed, in which the unused bandwidth is first given to the flows with least reserved rates. However, this algorithm has a per-packet complexity of $O(n)$ where n is the number of flows. The present study proposes a new algorithm, which is an approximation of REQ fairness with logarithmic per-packet complexity. The simulation results endorse than in different practical scenarios, the behavior of the new algorithm is quite similar to that of pure REQ fairness.

CHAPTER 1

INTRODUCTION

In a network of various interconnected nodes which provides QoS, a certain amount of bandwidth is reserved for each flow. A scheduling protocol is implemented to make decisions about the packet transmission from each backlogged flow. Selection of such protocol is a crucial decision because of the complexity and execution cost. Any such scheduling protocol can be analyzed on the basis of different kinds of guarantees it provides. Two such metrics of analysis are rate and fairness guarantees.

Rate Guarantee: A flow requests for a certain amount of bandwidth (rate) in the network before it begins transmitting packets. If the network has enough bandwidth available, it grants the reservation request. Otherwise, the request is denied. A scheduling protocol which provides rate guarantee assures that a flow gets atleast its reserved bandwidth at all times while it is active. A flow may receive more bandwidth than what it originally requested for, if bandwidth gets available. Such a guarantee thus ensures that any flow gets atleast a certain amount of service in the scheduler.

Fairness: If the network is not congested, it might be possible that a part of bandwidth is unused. It may happen for two reasons:

(1) There is some unallocated bandwidth available in the network.

(2) A set of flows are not fully utilizing their share of allocated bandwidth by transmitting packets at a slower rate than their service rate.

To use the resources efficiently, a scheduler thus distributes the unused bandwidth among the flows fairly while maintaining firewalls between them so that the allocation is followed strictly by all the flows. Fairness means a flow should not be "punished" (temporarily denied service) if it exceeds its reserved rate to take advantage of unused bandwidth in the channel. The way in which each protocol distributes the unused bandwidth is different.

If a flow gets empty, the scheduler starts serving the packets from the next non empty flow. Such a scheduler will be work-conserving since it doesn't let resources go to waste. In certain scenarios, a scheduler may decide not to distribute the unused bandwidth. Such a scheduler is non-work conserving. In the next few sections, a number of well known scheduling protocols are described.

1.1 SCHEDULING PROTOCOLS

Real-time applications, such as interactive audio and video, require from the network an assurance that their packets will be forwarded with a lower bound on the forwarding rate and also with a bounded end-to-end delay. Rate-guaranteed schedulers [22], [37], [43] is a family of protocols that is able to provide this assurance. Iconic examples of this protocol family include Virtual Clock (VC) [79], [84] and Weighted Fair Queuing (WFQ) [65], plus their many variations. One desirable property of a rate-guaranteed scheduler is fairness. Some protocols, such as Virtual Clock, are unfair, while others, like WFQ, are fair. Being able to provide fairness is desirable for some applications whose can adapt to changes in the bandwidth provided by the network. Examples of such flows are file transfers and multi-resolution video [61]. The sources of these flows will likely reserve from the network the smallest packet rate necessary to receive a minimum quality of service. The source is then able to detect that additional bandwidth is available due to side-effects observed in the network, such as a smaller end-to-end delay, and increase its packet-generation rate accordingly. Speaking generally of scheduling protocols, rate guarantee and fairness can be considered as two extreme ends between which various rate guaranteed scheduling algorithms fit in.

Prior to begining the discussion about scheduling protocols, it is important to establish a formal notion of a flow. A flow :

(1) is a stream of packets where each packet follows the same path in the network.

(2) avails a certain minimum amount of service in the network.

Each packet is uniquely bound to a particular flow which can be identified by analysing the headers in the packet. At any given time t, if a flow has packets in its queue, it is known as a *backlogged* flow. Otherwise, a flow with empty queue at any given time t is *idle*. Consider a set of all the flows represented by W. Then at any given time t, the set of backlogged flows given by B satisfies $B \subseteq W$.

In the following sections, consider an output channel with a capacity C and a set of flows W in the fluid server such that each flow is represented by $f_\mathrm{i}\ \varepsilon\ W,\ where\ 1 \leq i \leq n.$ Let B represent the set of backlogged flows in the fluid server.

1.1.1 Weighted Fair Queuing (WFQ)

A Generalized Processor Sharing (GPS) discipline provides rate and fairness guarantees. However, since it models a fluid server, it assumes that the scheduler can serve all the flows. However, a real scehduler has to transmit a packet from one flow at any given time and thus, it is impossible to practically realize a GPS system. Several packetized approximations which follow the footsteps of GPS have been proposed in past many years. Out of such known algorithms, Weighted Fair Queuing is well known to provide both rate and fairness guarantees. It assures that *(a)* each flow gets atleast the amount of bandwidth it reserved and *(b)* the unused bandwith is allocated among the flows in proportion of their reserved rates. Thus the flows with higher reserved rates get the larger share of unused bandwidth than those with smaller reserved rates.

Each flow $f_i\ \varepsilon\ B$ is assigned a weight w_i and gets the bandwidth c_i allocated as per the following rule:

$$c_i = \frac{w_i}{\sum_x w_x}, \quad f_x\ \varepsilon\ B \tag{1.1.1}$$

There are two schedulers (servers) in the system:

(1) Fake or bit-by-bit scheduler, which forwards a few bits of each flow at a time (i.e. fractions of a packet)
(2) Real scheduler, which
- assigns timestamps to packets.
- sends out packets in non-decreasing order of timestamp.

The timestamp is the "virtual" finishing time of the packet at the fake server. The virtual time *V(t)* at real time t is the "bit number" or "round-number" in the fake bit-by-bit server at real time *t*. In case of fair queuing, *V(t)* is increased by 1 every time one bit is forwarded from all the flows in the fake server. In case of WFQ, *V(t)* is increased by 1 every time when for each flow $f_i \varepsilon B$, w_i bits are forwarded from all the flows in the fake server. If there are fewer number of flows in the server, the *virtual time* will run faster. Also, for each tick in *virtual* time, the number of bits forwarded for flows with higher weights is more than those with lower weights.

Let $F_{f,i}$ be the virtual time when the *i-th* packet of flow *f* exits the bit-by-bit server, $A_{f,i}$ be the virtual time when the *i-th* packet of flow *f* exits the bit-by-bit server. Then following relation can be established.

(1.1.2)
$$F_{f,i} = max(V(A_{f,i}),\ F_{f,i-1}) + \frac{L_{f,i}}{w_f}$$

where $L_{f,i}$ is the length of *i-th* packet of flow f

In [65], it has been proven that if the rate admission control is maintained using the leaky bucket, end to end delay guarantee can be achieved. A packet gets out of the real server by the time it gets out of the fake server plus a constant. Thus,

(1.1.3)
$$Finishing\ time = F_{f,i} + \frac{L_{max}}{C}$$

where L_{max} is the maximum length of the packet.

It was long known that the WFQ can be implemented with a per-packet complexity of *O(n)*, until Valente [77] proposed a way to implement WFQ in *O(log(n))* per-packet complexity. He uses a complex scheme to organize the breaking points in a search tree to optimize searches.

1.1.2 Worst-case Fair Weighted Fair Queueing (WF^{2}Q)

In [65], Parekh stated a number of relationships between a GPS and a packetized approximation of GPS, which he called as PGPS (identical to WFQ). Two main claims were as follows:

(1) A packet gets out of a PGPS scheduler no later than it gets out of GPS scheduler with a delay of atmost one packet transmission time.

(2) PGPS cannot lag behind the GPS scheduler by more than one packet transmission.

From above two claims, one may deduce that a PGPS scheduler offers a service which is almost identical to a GPS scheduler with a variation of atmost one packet transmission time, which was once a popular belief until Bennet et al.[8] proved that there can be significant discrepances between a PGPS and a GPS scheduler. They claim that the PGPS scheduler can be far ahead of GPS in terms of service given to a session leading to an unfair behavior of the protocol towards other sessions.

In PGPS or WFQ approximation of GPS, packets are transmitted according to the order of their timestamps. It might happen that a packet may get transmitted in the PGPS before it exits the GPS scheduler. In this case, the next packet of the session might start receiving service even before the first packet finishes in GPS. This may lead WFQ to run ahead of GPS thus favoring a particular flow. In [8], a modified policy for selecting packets for transmission has been proposed which is called WF^{2}Q. According to this policy, when a WFQ scheduler selects the next packet for transmission, it choses a packet among those, which have already started receiving service in corresponding GPS system. Also, it has been established that with WF^{2}Q policy, the system offers service almost identical to GPS with a maximum difference of one packet transmission. Also, WF^{2}Q scheduler is proven to be work conserving.

1.1.3 Virtual Clock (VC)

Another packet service discipline has been proposed in by Zhang [84] which is known as Virtual Clock algorithm. The basic concept of this algorithm is somewhat similar to Time Division Multiplexing (TDM), which maintains a perfect firewall among various sessions. In TDM, each session can transmit data only in a particular time slot. However, this turns out to be inefficient on resource utilization as a time slot may get wasted if the session doesn't have any packet to transmit. Virtual Clock preserves the firewall property of a TDM and ensures work conserving nature of the scheduler.

In VC algorithm, two clocks are implemented, a real clock and a virtual clock. Whenever, a packet is received at the queuing node, it is timestamped with a virtual clock value. The packets are served in non-decreasing order of virtual clock timestamps of each packet. Suppose that a session f has a reserved bandwidth of R_f. Then for each such flow a value known as $Vtick_f$ [84]is calculated according to the following relationship:

$$Vtick_f = \frac{1}{R_f} \tag{1.1.4}$$

Now consider the *i-th* packet of flow f represented by $p_{f,i}$. Using 1.1.4, the Virtual Clock value $VC_{f,i}$ for $p_{f,i}$ can be calculated using the following equation:

$$VC_{f,i} = VC_{f,i-1} + Vtick_f \tag{1.1.5}$$

One important point to note here is that, flows which generate packets at a slower rate may accumulate credits over the time and thus create unfairness in the system towards other flows. This is taken care of by periodically synchronizing the Virtual Clock values with the real time so that VC values don't fall far behind of real time. In[79], it has been proven that Virtual Clock Algorithm has delay bound which is identical to that of WFQ. However, VC is not a fair scheduling policy.

Flows which transmit data at a speed higher than their reserved rates may get extra bandwidth for sometime if its available. But in later time, since their VC values will be far ahead of slower flows, they will get punished. This behavior has been modified in a scheme proposed in [76]. This scheme gives an end-to-end delay guarantee similar to WFQ and also exhibits throughput fairness.

CHAPTER 2

RATE EQUALIZATION FAIRNESS

In all the previously discussed algorithms, the rate guarantee is always ensured by providing atleast the reserved rate of service to a session. The way fairness is ensured varies with different protocols. Algorithms, which follow the GPS discipline exhibit fairness by distributing the unused bandwidth in proportion to the reserved rates of flows. Algorithms like Leap Forward Virtual Clock [76] maintain fairness by moving misbehaved sessions to a separate queue while keeping the well-behaved flows together.

BIBLIOGRAPHY

[1] W. Almesberge, T. Ferrari, and J. L. Boudec. SRP: A scalable resource reservation protocol for the internet. In *Proc. of The International Workshop on Quality of Service (IWQOS)*, 1998.

[2] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention resolution protocols. In *37th Annu. IEEE Symp. Foundations Computer Science (FOCS)*, pages 380–389, October 1996.

[3] Sanjoy K. Baruah. Pfair scheduling of generalized pinwheel task systems. *IEEE Transactions on Computers*, 47(7), July 1998.

[4] J.C.R. Bennet, C. Partridge, and N. Shectman. Packet reordering is not pathological network behavior. *IEEE/ACM Transactions on Networking*, 7(6), December 1999.

[5] J. C. R. Bennett and et al. Delay jitter bounds and packet scale rate guarantee for Expedited Forwarding. *IEEE/ACM Transactions on Networking*, 10(4):529–540, August 2002.

[6] J. C.R. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689, October 1997.

[7] Jon C. R. Bennett, Kent Benson, Anna Charny, William F. Courtney, and Jean-Yves Le Boudec. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. *IEEE/ACM Transactions on Networking*, 10(4), August 2002.

[8] Jon C.R. Bennett and Hui Zhang. WF2Q: worst-case fair weighted fair queueing. In *IEEE INFOCOM Conference*, 1996.

[9] J.M. Blanquer and B. Ozden. Fair queuing for aggregated multiple links. In *Proc. of the ACM SIGCOMM Conference*, 2001.

[10] Jean-Yves Le Boudec and Gerard Hebuterne. Comments on 'A deterministic approach to the end-to-end analysis of packet flows in connection oriented networks'. *IEEE/ACM Transactions on Networking*, 8(1):121–124, February 2000.

[11] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture. Internet RFC 1633.

[12] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan. A framework for multiprotocol label switching. 1997. Internet draft draft-ietf-mpls-framework-02.txt.

[13] B. Caprita, J. Nieh, and W. C. Chan. Group round robin: Improving the fairness and complexity of packet scheduling. In *Proc. Symp. on Architecture for Netw. Commun. Syst.*, pages 29–40, October 2005.

[14] C. S. Chang. On deterministic traffic regulation and service guarantee: A systematic approach by filtering. *IEEE Transactions on Information Theory*, 44:1096–1107, August 1998.

[15] Imrich Chlamtac, Andras Farago, and Hongbiao Zhang. A new criterion for router selection in communication networks with delay-sensitive traffic. In *Proc. Conf. on Computer Communication and Networks (ICCCN)*, pages 317–320, Rockvilee, MD, October 1996.

[16] Imrich Chlamtac, Andras Farago, Hongbiao Zhang, and Andrea Fumagalli. A deterministic approach to the end-to-end analysis of packet flows in connection-oriented networks. *IEEE/ACM Transactions on Networking*, 6(4):422–431, August 1998. mynote.

[17] Jorge Cobb. Universal timestamp scheduling for real-time networks. *Computer Networks*, 31:2341–2360, 1999. Elsevier.

[18] Jorge Cobb. Preserving quality of service without per-flow state. In *Proc. IEEE International Conference on Network Protocols (ICNP)*, November 2001.

[19] Jorge Cobb. Preserving quality of service guarantees in-spite of flow aggregation. *IEEE/ACM Transactions on Networking*, 10(1):43–53, February 2002.

[20] Jorge Cobb. Scalable quality of service across multiple domains. *Computer Communications*, 28(18):1997–2008, November 2005. Elsevier.

[21] Jorge Cobb. Work conserving fair-aggregation with rate-independent delay. In *Proc. IEEE International Conference on Computer Communication and Networks (ICCCN)*, 2008.

[22] Jorge Cobb and Mohamed Gouda. Flow theory. *IEEE/ACM Transactions on Networking*, 5(5):661–674, October 1997.

[23] Jorge Cobb, Mohamed Gouda, and Amal-El Nahas. Time-shift scheduling: Fair scheduling of flows in high-speed networks. *IEEE/ACM Transactions on Networking*, 6(3):274–285, June 1998.

[24] Jorge Cobb and Miaohua Lin. End-to-end delay guarantees for multiple-channel schedulers. In *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, Miami, May 2002.

[25] Jorge Cobb and Miaohua Lin. A theory of multi-channel schedulers for quality of service. *Journal of High-Speed Networks*, 12(1 and 2):61–86, November 2002. IOS Press.

[26] Jorge Cobb and Zhe Xu. Maintaining flow isolation in work-conserving flow aggregation. In *Proc. IEEE GLOBECOM Conference*, 2005.

[27] Jorge Cobb and Zhe Xu. Guaranteed throughput in work-conserving flow aggregation through deadline reuse. In *Proceedings of the ICCCN Conference*, October 2006.

[28] Jorge Cobb and Zhe Xu. Fair-aggregation across multiple core networks. In *Proc. IEEE International Conference on Computer Communication and Networks (ICCCN) Workshop on Advanced Networking and Communications*, 2007.

[29] Jorge A. Cobb. Rate-independent delay across state-reduced networks. *The University of Texas at Dallas Technical Report*, May 2009.

[30] Jorge A. Cobb. Rate equalization: A new approach to fairness in deterministic quality of service. *37th Annual IEEE Conference on Local Computer Networks*, 0:50–57, 2011.

[31] R. L. Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1), January 1991.

[32] R. L. Cruz. A calculus for network delay, part ii: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1), January 1991.

[33] B. Davie and et al. An expedited forwarding PHB. March 2002. IETF RFC 3246.

[34] S. K. Dhall and C. L. Liu. On a real-time scheduling problem. *Operations Research*, 26(1):127–140, 1978.

[35] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3), April 1990.

[36] D. Ferrari and H. Zhang. Rate controlled static priority queuing. In *Proc. of the IEEE INFOCOM Conference*, 1993.

[37] N. Figueira and J. Pasquale. Leave-in-time: A new service discipline for real-time communications in a packet-switching data network. In *Proc. of the ACM SIGCOMM Conference*, 1995.

[38] Norival R. Figueira and Joseph Pasquale. A schedulability condition for deadline-ordered service disciplines. *IEEE/ACM Trans. Netw.*, 5(2):232–244, 1997.

[39] Huirong Fu and Edward W. Knightly. A simple model of real-time flow aggregation. *IEEE/ACM Transactions on Networking*, 11(3), June 2003.

[40] S. J. Golestani. A stop-and-go queuing framework for congestion management. In *Proc. of the ACM SIGCOMM Conference*, 1990.

[41] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *IEEE INFOCOM Conference*, 1994.

[42] Mohamed Gouda. *Elements of Network Protocol Design*. John Wiley and Sons, 1998.

[43] P. Goyal, S. Lam, and H. Vin. Determining end-to-end delay bounds in heterogeneous networks. In *Proc.of the NOSSDAV Workshop*, 1995.

[44] Pawan Goyal, Harrick M. Vin, and Haichen Cheng. Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Trans. Netw.*, 5:690–704, October 1997.

[45] C. Guo. Srr: an o(1) time-complexity packet scheduler for flows in multiservice packet networks. *IEEE/ACM Transactions on Networking*, 12(6), December 2004.

[46] C. Guo. G-3: An o(1) time complexity packet scheduler that provides bounded end-to-end delay. In *Proc. of the IEEE INFOCOM Conf.*, 2007.

[47] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding phb group. Internet RFC 2597.

[48] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel. The pinwheel: A real-time scheduling problem. In *Proc. 22nd Hawaii Int'll Conf. System Science*, 1989.

[49] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding phb. Internet RFC 2598.

[50] Yuming Jiang. Delay bounds for a network of guaranteed rate servers with FIFO aggregation. *Computer Networks*, 40:683–694, 2002. Elsevier.

[51] Yuming Jiang. Per-domain packet scale rate guarantee for expedited forwarding. In *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, 2003. LNCS 2707.

[52] Yuming Jiang. Relationship between guaranteed rate server and latency rate server. *Computer Networks*, 43:307–315, 2003. Elsevier.

[53] Yuming Jiang. Link-based fair aggregation: A simple approach to scalable support of per-flow service guarantees. In *Proc. of IFIP Networking Conference*, 2004. Springer LNCS 3042.

[54] Jasleen Kaur and Harrick M. Vin. Core-stateless guaranteed rate scheduling algorithms. In *Proc. of the IEEE INFOCOM Conf.*, 2001.

[55] Jasleen Kaur and Harrick M. Vin. Core stateless guaranteed throughput networks. In *Proc. of the IEEE INFOCOM Conf.*, 2003.

[56] Jasleen Kaur and Harrick M. Vin. Providing deterministic end-to-end fairness guarantees in corestateless networks. In *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, 2003.

[57] Srinivasan Keshav. A control theoretic approach to flow control. In *Proc. ACM SIGCOMM Conference*, 1991.

[58] J. Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory*, 44:1087–1096, May 1998.

[59] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 20, January 1973.

[60] S. Machiraju, M. Seshadri, and I. Stoica. A scalable and robust solution for bandwidth allocation. In *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, Miami, May 2002.

[61] Partho P. Mishra and Hemant Kanakia. A hop by hop rate-based congestion control scheme. *SIGCOMM Comput. Commun. Rev.*, 22:112–123, October 1992.

[62] M. Moir and S. Ramamurthy. Pfair scheduling of fixed and migrating periodic tasks on multiple resources. In *IEEE Real-Time Systems Symposium*, 1999.

[63] Florian-Daniel Otel. Deterministic qos guarantees using route interference. In *Proc. of The IEEE International Symposium on Computers and Communication (ISCC)*, 2003.

[64] Florian-Daniel Otel and Jean-Yves Le Boudec. Deterministic end-to-end delay guarantees in a heterogeneous route interference environment. In *Proc. of The International Workshop on Quality of Future Internet Services*, 2003. Springer LNCS 2811.

[65] A. K. J. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Transactions on Networking*, 1(3), June 1993.

[66] C. Partridge. *Gigabit Networking*. Addison-Wesley, 1993.

[67] Jingyu Qiu and Edward W. Knightly. Measurement-based admission control with aggregate traffic envelopes. *IEEE/ACM Transactions on Networking*, 9(2), April 2001.

[68] S. Ramabhadran and J. Pasquale. Stratified round robin: A low com- plexity packet scheduler with bandwidth fairness and bounded delay. *ACM Comput. Commun. Review*, 33(4):239–250, October 2003.

[69] K. K. Ramakrishnan and Raj Jain. A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer. *SIGCOMM Comput. Commun. Rev.*, 18:303–313, August 1988.

[70] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. Internet RFC 2212.

[71] D. Stidialis and A. Varma. Rate proportional servers: A design methodology for fair queuing algorithms. *IEEE/ACM Transactions on Networking*, April 1998.

[72] D. Stiliadis and A. Varma. Latency rate servers: a general model for analysis of traffic scheduling algorithms. *IEEE/ACM Transactions on Networking*, 6(5):611–624, 1998.

[73] Dimitrios Stiliadis and Anujan Varma. A general methodology for designing efficient traffic scheduling and shaping algorithms. In *IEEE INFOCOM Conference*, 1997.

[74] Ian Stoica and Hui Zhang. Providing guaranteed services without per-flow management. In *Proc. of the ACM SIGCOMM Conference*, 1999.

[75] Wei Sun and Kang G. Shin. Coordinated aggregate scheduling for improving end-to-end delay performance. In *Proc. of the IEEE Workshop on Quality of Service (IWQoS)*, 2004.

[76] Subhash Suri, George Varghese, and Girish Chandranmenon. Leap forward virtual clock: A new fair queuing scheme with guaranteed delays and throughput fairness. In *In Proceedings of INFOCOM'97*, 1997.

[77] Paolo Valente. Exact gps simulation with logarithmic complexity, and its application to an optimally fair scheduler. *SIGCOMM Comput. Commun. Rev.*, 34:269–280, August 2004.

[78] J. Wroclawski. Specification of controlled-load network element service. 1997. Internet RFC 2211.

[79] G. Xie and S. Lam. Delay guarantee of the virtual clock server. *IEEE/ACM Transactions on Networking*, pages 683–689, December 1995.

[80] Zhe Xu. *Scalable Scheduling for Quality of Service guarantees*. PhD thesis, The University of Texas at Dallas, Aug. 2007.

[81] X. Yuan and Z. Duan. Frr: A proportional and worst-case fair round robin scheduler. In *Proc. IEEE INFOCOM*, pages 831–842, March 2005.

[82] Hongbiao Zhang. A note on deterministic end-to-end delay in connection-oriented networks. In *Proc. of the IEEE International Conference on Communications (ICC)*, pages 1223–1227, Vancouver, 1999.

[83] Hui Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 93(10), October 1995.

[84] Lixia Zhang. Virtual clock: A new traffic control algorithm for packet-switched networks. *ACM Transactions on Computer Systems*, 9(2):101–124, May 1991.

[85] Z. Zhang, Z. Duan, L. Gao, and Y. T. Hou. Decoupling QoS control from core routers: A novel bandwidth architecture for scalable support for guaranteed services. In *Proc. ACM SIGCOMM Conference*, 2000.

[86] Q. Zheng and K. G. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Transaction on Communications*, 42(3), March 1994.

[87] Shin K.G. Zheng Q. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Transactions on Communications*, 42(2-4), 1994.

[88] L. Zhong, J. Xu, and X. Wang. Vwqgrr: A novel packet scheduler. In *Proc. of the IEEE ICN*, pages 22–28, April 2007.