A SURVEY ON THE CHRONOLOGICAL EVOLUTION OF TIMESTAMP SCHEDULERS IN PACKET SWITCHING NETWORKS

Yaser Miaji, Suhaidi Hassan

InternetWorks Research Group, UUM College of Arts and Sciences,
Universiti Utara Malaysia, 06010 UUM Sintok, MALAYSIA.
yaser@ieee.org, suhaidi@ieee.org

Abstract

The interest in solving the issue of congestion or flow control in network established from the first discovery and increase popularity of the Internet in 1967 or earlier. As the use of the network deployed and the popularity increase, the issue grows and the demand for an optimal or tentative solution becomes obvious. Since that time there has been an intensive effort from the scholars and researchers to solve the congestion control problem. The problem get worse by the engagement of novel traffic with different characteristics for application called real-time applications such as video and voice applications. Another cause of this demand is the user himself. The attempt in solving the congestion problem in network layer was popular in 90's. This article will demonstrate chronologically how the attempts toward timestamp based scheduling in the packet-switch network have been evolved. Furthermore, the benefit and the drawbacks of using a mechanism will be presented. Also, a brief explanation of the mathematical, conceptual or implementation issue of a mechanism is given. The key success of the scheduler in the market will be highlighted. This paper will stimulate the research thinking to identify the importance and the ability of scheduling in routers to enhance quality of service (QoS) for real time application over other solution in several layers. In addition it will assist the researcher to distinguish the key failure of other proposed mechanisms which have not been implemented in real routers.

Keywords: QoS, real time application, scheduling, queueing, network layer.

1 Introduction

The interest in licking the issue of congestion or flow control in network established from the first discovery and increase popularity of the Internet in 1967 or earlier. The congestion problem was not fully exposed at the presence of ARPANET based network due to the bandwidth informality and switching node identicalness, and the excessiveness of the capacity particularly in the department of defense in early 80's [36-38]. However, as the use of the network deployed and the popularity increase, the issue grows and the demand for an optimal or tentative solution becomes obvious. Since that time there has been an intensive effort from the scholar and researchers to solve the congestion control problem.

The problem get worse by the presence of novel traffic with different characteristics for application called real time application such as video and voice application. Another cause of this demand is the user himself. Users start to complain and request for much effective service. RFC 1193 identifies the user requirements for reliable QoS which depend on the service level of agreement. There are several attempts in different network layers to solve the issues of congestion control.

The attempts in solving the congestion problem in network layer were popular in 90's. Researchers tend to schedule the packet transmission to fulfill the requirements for different network traffic. There was a comprehensive study of queueing algorithms and scheduling disciplines to control the congestion in the network. The argument surrounded the effectiveness in term of bounding delay and packet loss rate, the efficiency in term of users' protection from misbehaved users, and the fairness of a scheduler is quite popular in the network community by the meddle of 90's. This argument result in a discovery of several scheduling mechanisms which based on two main categories namely: timestamp scheduling and round robin scheduling.

However, the effort toward the scheduling solution has been humbled because both round robin and timestamp scheduler have failed to achieve the requirements of all the scheduler's requirements; fairness, bounded delay, loss rate, and low complexity. The attempts have been changed toward another network layers. However, there are a few researchers conducted in late 1990's as well as 2000 in the scheduling area. Nevertheless, almost none of the approaches have been

978-1-4244-4591-2/09/$25.00 ©2009 IEEE

implemented in the market place. For a certain purpose there is a believe that this area should be revived. According to Internet coaching library, the Internet users have grown just over 342.2% since 2000 [28]. Consequently, the usage will suffer from a magnificent growth. Additionally, the emergence of new application such as online gaming and three dimensional (3D) games and application, results in a demand for distinguish effort in the area of scheduling and queuing discipline. Therefore, as been claimed by Floyd [13], there will be a negative impact of best effort traffic and unfairness of the competing traffic.

This article will demonstrate chronologically how the attempts toward timestamp based scheduling have been evolved. Furthermore, the benefit and the drawbacks of using the mechanism will be presented. Also, there will be a brief explanation of the mathematical, conceptual and implementation issue. The key success of the scheduler in the market will be highlighted. This paper will stimulate the research thinking to identify the importance and the ability of scheduling in routers to enhance QoS for real time application. In addition it will assist the researcher to distinguish the key failure of other proposed mechanisms which have not been implemented in real routers. Next section will present the importance of scheduling mechanism in routers for supporting QoS for real time applications over other solution on different layers such as increase the bandwidth and improve the software capability. This will followed by identification of the two research directions and specifies the timestamp category to be illustrated. The following sections have been organized chronologically starting from the discovery of the congestion and end with the current evolution. The final section is summary and conclusion.

2 The Importance of Scheduling Mechanism

There are several reasons to believe in the effectiveness and the efficiency of scheduling and queueing in router to augment QoS for real time application. The use of user datagram protocol (UDP) in transport layer as the primary protocol for transporting real time application has limited the improvement of the enhancement of QoS in this particular layer. Although, the standardization of datagram congestion control protocol (DCCP) to provide sustainability for UDP, the reliability issue still available. Therefore, the scholar effort to provide enhancement for QoS for real time application is less efficient. Furthermore, it has been declared in RFC2309 that congestion avoidance mechanisms in TCP are not sufficient in all circumstances. Hence, there should be another area to enhance the QoS for real time application and provide fairness among all network traffic.

The attempt to overcome the problem by constructing user software is much efficient with the cost as a side effect. Real time protocol (RTP) header compression is one technique to reduce the real time application header which consequently reduces the delay. Increasing the bandwidth to accommodate the large amount of real time application data is not the adequate solution for the problem. With the presence of first-come-first-serve (FCFS) technique the issue will increase. The hanger for bandwidth will increase even with the potential increase of the bandwidth as the real time application files is significantly large, especially, for video streaming and conferencing. Although, video application is mush tolerant to the delay and loss then the voice, the emergence of new application such as IPTV make customer demand constrains this tolerance.

3 Research Direction

Researchers took two different paths in order to achieve the goal of solving the congestion problem in routers using scheduling. The first approach is using timestamp scheduler. The aim of this approach is to approximate the ideal process sharing which called generalized process sharing (GPS). Researchers involved in this class seek for better fairness and bound the delay to enhance the QoS for real time application. The side effect of this research is negligence of complexity issue which makes their scheduler much complicated (with the complexity of (O(n)) than to be implemented in real routers. In addition the complexity problem increase as the network organized hierarchically.

The second approach was much simpler. Researchers involved in this category were aware of the complexity issue. However, they were not conscious of the fairness and bounding delay issues. The result is simpler scheduler with fairness and bounding delay trade off. Moreover, the issue of fairness increase as the network tends to be hierarchy. Therefore, this article will be in favor of scheduling mechanism using timestamp scheduler which supports real time application characteristics. The next section will demonstrate the chronological evolution of timestamp scheduler.

4 The Discovery

The issue of network congestion was first successfully addressed by [36], [37], [38] and published in RFC 896. In Nagle [36], there was a clear explanation of the congestion collapse. The issue of the integrated network has been mentioned in Nagle article by introducing a solution for the problem of small-packet. This means that as different traffic with different characteristics emerge to the network, the network will suffer from providing reliability and quality for serving this new traffic. Consequently, the network should behave differently. The inhibition of sending new packets till the arrival of acknowledgement (ACK) was his basic attempt [38]. In other words, he basically eliminates the time constraint in the vendor machine. His approach was in TCP layer and quite simple and realistic but it would not serve all traffic patterns.

In addition to the small packet solution which introduces the problem of traffic diversity, Nagle proposed the gateway self defense scheme which has been later called as fair queueing (FQ). FQ considered as, in principle, the first successful approach to reduce the impact of misbehaved host and introduces the fairness demand. However, his proposal has not been implemented not even in a simulation environment. Therefore, the attention is not vital.

The famous weighted fair queueing (WFQ) was first proposed by [10]. There are several reasons for the success of WFQ. The actual demand for a technique to solve the congestion and QoS issues in the network leads to the wide acceptance of the discipline in the network community. Another key success which distinguishes this mechanism is comprehensive coverage of the mechanism description which enhanced with the simulation results. The mechanism adopts the fairness definition as a Maximization of the allocation for the most poorly treated session (maximize the minimum). Since the congestion problems dramatically grow, the Internet engineering task force (IETF) in 1996 approved WFQ as a basic building block for future integrated service network. Actually, the practical implementation of WFQ has been presented in [37] and it is primarily one of the reasons for WFQ wide deployment. Eventually, this stage has a happy ending with crucial solution, as it has been assumed, to the fairness and congestion problem.

5 The Revolution Period

As [10] approach has been approved by IETF, researchers start a fairly long discussion and the effort toward the scheduling algorithm has been raised. As the algorithm evolved, some of the side effects of using WFQ have been illuminated and several novel and enhanced algorithms have been proposed. Self clocked fair queuing (SCFQ) proposed by [18] share some similarities with WFQ. It is believed that its first conceptual proposition was by [9]. The principle of SCFQ that every packet is tagged prior of its involvement in the queue and the packet served as its tag increase. Despite the fact that it is simpler than WFQ, it has the sacrifice of the fairness and less end-to-end bounding delay as the number of session grows which makes it less concurrent.

There are some other researchers in favor of WFQ and some other disciplines oppose it. Golestani et al. [19] introduces a new algorithm, which work cooperatively with first-in-first-out (FIFO), to solve the problem of standalone FIFO in the degradation in controlling the delay and the congestion of a network. His conceptual and mathematical approach, which has been presented in several articles [15-20], tended to address the issue in the frame level by introducing the departing and arriving frame. This attempt considered as the basic blocks for timestamp scheduling even before the proposition of [10] as it has been mentioned in his thesis which approved in 1979 [16].

Golestian [16] claims that his approach designed for connection-oriented network. Nevertheless, there are two parts of his strategy; the packet admission and the stop-and-go. Since stop-and-go primarily tends to be implemented in the switches, it could be adopted for connectionless-oriented network. Essentially, his concept is to assign a departing and arriving frame for each packet and divide the time in to frames. The served packet should be departed in the previous time frame in order to be sent. However, the lack of work-conserving discipline and unfairness, despite the fact that his approach is better in regard of bounding delay-variance, led to less popularity of the algorithm in the market environment.

L. Zhang [55] demonstrates his new discipline which called virtual clock (VC). Primarily, he introduces his algorithm in his thesis in 1989 and his first paper in the discipline in 1991. The concept of his mechanism is to emulate the time division multiplexing (TDM) by allocating a virtual transmission time for each packet. However, its insufficient ability to provide fairness, which considered as a primary condition for designing a scheduler, led to its supersession.

There are two disciplines which cognitively correlated to earliest deadline first scheduling (EDF); delay earliest-due-date (Delay-EDD) and jitter earliest-due-date (Jitter-EDD). However, there is a subtle difference EDF, Delay-EDD and Jitter-EDD. EDF does not provide protection to the host from misbehavior host. Delay-EDD, which proposed by [11,12] attempts to overcome this issue by assigning a rate to each flow and compute a deadline based on packet arrival time and allocate separate rate and delay. However, its lack of providing proper bounding for delay variation (jitter), results in its failure to enhance the QoS for real time application. Jitter-EDD, which proposed by [49] achieves a satisfactory level of bounding delay variation and requires less buffer size. Basically, it maintains a head time which stored in the packet header and different from deadline and arriving time. This packet head time delays the packet to allow reconstruction and hence avoidance of jitter. Nevertheless, it utilizes the concept of non-conserving scheduler which is not preferred in term of providing QoS for real time application.

The innovation in the scheduling was not limited in the above mentioned discipline in this period, rather it has been expanded but these are the distinguished mechanisms. It titled as the revolution period because there are several novel ideas has been proposed on this period. There is diversity in implementing each mechanism either in mathematical or simulation approach. To conclude these mechanisms will be counted and grouped in two main categories; work conserving and non-work conserving discipline. The work conserving algorithms are WFQ, SCFQ, Virtual Clock and Delay-EDD. The non-work conserving algorithms are Stop-and-Go and Jitter-EDD. Finally, [54] is a distinguish paper which compares those algorithms in their underlying algorithm, performance guarantee provided, buffer space requirement, associated admission control policy and implementation issues.

6 Argument Period

Upon the magnificent achievement of the WFQ in convincing IETF to accept the mechanism as the basic blocks for routers in 1996, researchers establish a comprehensive discussion regarding the effectiveness, efficiency and fairness of its implementation. Study conducted by [24] concurs with the fairness and the effectiveness of WFQ in its approximation of Generalized Process Sharing. The author verifies his opinion by using a mathematical and simulation model.

However, another study conducted by [2] asserts that WFQ is not good enough in providing fairness. Therefore the author comes up with another model [3]. Worst case fair weighted fair queuing (WF^2Q) is Bennett proposed model. Wf2Q approximate GPS with high probability and difference of no more one packet. In a WF^2Q system, when the server chooses the next packet at time *t*, it chooses only from the packets that have started receiving service in the corresponding GPS at *t,* and chooses the packet among them that would complete service first in the corresponding GPS. Nevertheless, the time complexity of implementing WF^2Q is high because it based on a virtual time function which is defined with respect to the corresponding GPS system. This leads to considerable computational complexity due to the need for simulating events in the GPS system.

Bennett and Zhang [1] have introduced hierarchical packet fair queueing (HPFQ) which is also called enhanced WF^2Q (WF^2Q+). Their approach is similar to WF^2Q with simpler implementation. In WF^2Q +, each flow is associated with a *weight*, such that the sum of the weights of all flows is no larger than a predefined value *W*. A flow's weight specifies how much share of the capacity of the output link a flow is entitled to receive. Note that if *W* is equal to the capacity of the link, then the weights are actual bandwidth given to each flow. By keeping track of eligible times and finishing times of flows, the packets could be scheduled according to WF^2Q +. [6,7] simulates WF^2Q + and compare its performance with SPFQ, which will be described later. In spite of the fairness and less complexity of HPFQ, there is an issue regarding the distribution of the bandwidth in the presence of hierarchical complex network. Moreover, the model has lack of ability to serve the multimedia traffics due to less-consideration of the diversity requirement of the multimedia traffic. It, also, has the inability to accommodate the dynamic flow set and insulating the similar traffic.

Start time fair queueing (SFQ) ([21,23] applies different method with different computational method as well for starting and finishing time for a packet. SFQ has finish number and start number. Start number of a packet arriving into inactive connection is the current round number otherwise it is the finish number of the previous packet. Additionally, round number is set to start number of the current packet. Hence, packets scheduled in the increasing order of start number. SFQ is effective in variable bit rate (VBR) application such as the video application. Its computation method is less complex compare to WFQ and WF^2Q. Nevertheless, packet sorting complexity is an implementation issue which prevents the utilization of SFQ in real routers. Furthermore, the end-to-end delay grows proportionally with the number of session [22].

Leap forward virtual clock (LFVC) introduced in [47] has some properties with a subtle discrepancy with VC. It consists of two queues; high and low priority. Packets from the high priority queue are serviced by lowest tag first. Contrary, packets from the low priority queue must still be serviced before their tag. Furthermore, it applies the punishment policy which means that any flow exceed its throughput will be postponed. There are two obvious limitations for such discipline; the latency will be increased as the number of punished queue increase which leads to the second issue which is the accumulative packets.

Stiliadis [44] presents their mechanism with the complexity of *O(1)* for time stamp and with bounded delay and reasonable fairness which called starting potential based fair queueing (SPFQ). In this algorithm the virtual time is derived from the based potential which defines as the minimum of the starting potential of all backlogged session. Additionally, the system potential is re-calibrated to the minimum of start potential every time this minimum changes. Consequently, in term of its implementation, it requires more state information to be maintained which cause a crucial implementation issue and consequently the system complexity will increase linearly with the time.

Minimum delay self clocked fair queueing (MD-SCFQ) is another algorithm which proposed by [5]. Since MD-SCFQ uses virtual finishing time of the packet as the system virtual time, the complexity of calculating system virtual time is *O(1)*. Likewise, its fairness is optimal beside its bounded delay which is reliable. Nevertheless, the recalibration of the system virtual time which passed on weighted average virtual start time of all backlogged session introduces additional computation which results in more complexity.

To sum up this stage, the complexity issue is the dominant of this stage and therefore, none of the proposed discipline has been successful in term of its implementation in real router. Even though, STFQ, SPFQ and MD-SCFQ has less complexity than WF2Q, WF2Q+, LFVC, and SFQ, the potential increase of the system complexity over time obstacles its implementation.

7 The Indolent Period

During the last two stages, which compose of one decade, there are at least fourteen disciplines have been proposed to combat the issue of scheduling in routers using the time stamp principle. On the other hand, this stage constitutes the same period with much less proposed disciplines. Furthermore, during the last three years the effort toward improving the scheduler has experience an indolence which could be, obviously, seen from the literature.

Simple weighted fair queueing (SWFQ) proposed by [50] has adopted the concept of proportional rate server (PRS). Basically, when the backlog occurs, among the backlogged connections, only the set of connections with the minimum potential at time t is served. Each connection in this set is served with an instantaneous rate proportional to its reservation, so as to increase the potentials of the connections in this set at the same rate. The author claims that his approach has overcome the system potential complexity issue by proposing the execution of the re-calibration every time the transmission completed. However, this attempt could introduce new implementation complexity issue which could critical.

Xiaohui et al. [52] demonstrate a basic proposition of one timestamp per queue (OTPQ) which could be used in the WFQ as well as the one timestamp per packet (OTPP) approach. The article presents some mathematical approaches beside the implementation and analysis. There are some disadvantages regarding their proposition. Firstly, any failure in calculating the OTPQ will result in crucial degradation in the fairness and inaccuracy. Furthermore, the proposition was impractical.

New start potential fair queueing (NSPFQ) is proposed in [31] and proposed as mean starting potential fair queueing in (MSPFQ) in [32] as an enhancement for SPFQ which proposed in [44]. There a slight discrepancy between both algorithms. NSPFQ recalibrates the system virtual time using the maximum timestamp increment (MTI), which defines as a constant value that determine at the system setup and mathematically as the result of the division of maximum packet length and a rate, while it uses the system virtual time for a newly arrived packet as the last calibrated system virtual time added by the elapsed real time between two calibration events. Nevertheless, since the NSPFQ is actually, a PRF, it still has the system complexity of *O log(n)*.

A new method of scheduling has proposed by [41] which called greedy fair queueing (GrFQ). The concept of GrFQ is to seek of the minimization of the maximum difference in the normalized service received by any two flows when the next packet transmission completes. Obviously, this principle has been inspired by SWFQ discipline. The author uses the relative fairness bound (RFB) to prove the fairness of the algorithm. In addition, according to the simulation result, the discipline bounding delay is approximately similar to WFQ. However, its complexity is *O(log N)* with respect to the flow which is still high.

Since there are several issues associated with the implementation of HPFQ [1], Ju [29] introduced new method which augments the mechanism and called novel HPFQ. The principle of the proposition is to divide the scheduling task in to four server; hard-QoS server scheduling, soft-QoS server scheduling, best-effort server scheduling and co-scheduling among the previous mentioned three servers. The rest of the algorithm is typical to the HPFQ. With this sort of division of tasks, it will involve practical complexity.

Lee et al. [33] presents a new scheduling mechanism which adopts the virtual clock principle and called worst-case fair weighted fair queueing with maximum rate control (WF^2Q-M). WF^2Q-M claimed to be consisted of packet shaping and scheduler which enforce the maximum rate constraints with low packet loss. However, the most obvious drawback of such algorithm is potential increase in the complexity which result of the combination of both scheduler and shaper.

Finishing potential fair queueing (FPFQ) is introduced in [25] which based on PRS model. The main contribution of this algorithm is to reduce the system potential complexity which involve in all PRS based scheduler by the combination of system potential function with the function used for the determination of the next served packet. Since this approach is quite version there are, so far, no obvious issue associated with it implementation.

To some up, the PRS scheduler model in this stage constitute most of the researcher concentration. FPFQ, GrFQ, NSPFQ, and SWFQ is the PRS based scheduler and WF2Q-M and novel HPFQ adopted different methods.

8 Conclusions

In this paper the importance of scheduling in router has been demonstrated. Then, the chronological evolution of timestamp scheduler has been presented. Obviously, the scheduling algorithms revolutionary developed in early and late 90’s. However, in the 2000’s, more attention been devoted for enhancing QoS in other layers. Most of current routers utilize the old discipline whether timestamp or round robin. Therefore, WFQ still the dominant of the scheduling mansion.

References

[1] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 143-156, 1996.

[2] J. C. R. Bennett and H. Zhang, "Why WFQ Is Not Good Enough for Integrated Services Networks," 1996.

[3] J. C. R. Bennett, H. Zhang, and F. Syst, "WF 2 Q: worst-case fair weighted fair queueing," 1996.

[4] H. B. Chiou, "Enhancing the Fairness of TCP over Internet Using an Improved Hierarchical Packet Fair Queueing Scheme," 2000.

[5] F. M. Chiussi and A. Francini, "Minimum-delay self-clocked fair queueing algorithm for packet-switched networks," 1998.

[6] N. Ciulli and S. Giordano, "Analysis and simulation of WF2Q+ based schedulers: comparisons, compliance with theoretical bounds and influence on end-to-end delay jitter," Computer Networks, vol. 37, pp. 579-599, 2001.

[7] N. Ciulli and S. Giordano, "Analysis and Simulation of WF^ 2Q+ Based Schedulers: Comparisons and Compliance with Theoretical Bounds," Lecture notes in computer science, pp. 255-272, 2001.

[8] D. Comer and M. Martynov, "Design and Analysis of Hybrid Packet Schedulers," 2008.

[9] J. R. Davin and A. T. Heybey, "A simulation study of fair queueing and policy enforcement," 1990.

[10] Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 1-12, 1989.

[11] D. Ferrari, "Client Requirements for Real-Time Communication Services;RFC-1193," Internet Request for Comments, 1990.

[12] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," Selected Areas in Communications, IEEE Journal on, vol. 8, pp. 368-379, 1990.

[13] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," IEEE/ACM Transactions on Networking (TON), vol. 7, pp. 458-472, 1999.

[14] Francini and F. M. Chiussi, "A weighted fair queueing scheduler with decoupled bandwidth and delay guarantees for the support of voice traffic," 2001.

[15] S. J. Golestaani, I. Massachusetts Inst Of Tech Cambridge Lab For, and S. Decision, "A Unified Theory of Flow Control and Routing in Data Communication Networks," Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1979.

[16] S. J. Golestani, "A stop-and-go queueing framework for congestion management," ACM SIGCOMM Computer Communication Review, vol. 20, pp. 8-18, 1990.

[17] S. J. Golestani and M. Bellcore, "Duration-limited statistical multiplexing of delay-sensitive traffic in packet networks," 1991.

[18] S. J. Golestani and M. Bellcore, "A self-clocked fair queueing scheme for broadband applications," 1994.

[19] S. J. Golestani, B. C. Res, and N. J. Morristown, "Congestion-free transmission of real-time traffic in packet networks," 1990.

[20] S. J. Golestani, B. C. Res, and N. J. Morristown, "Congestion-free communication in high-speed packet networks," IEEE Transactions on Communications, vol. 39, pp. 1802-1812, 1991.

[21] P. Goyal and B. Tech, Packet scheduling algorithms for integrated services networks: University of Texas at Austin, 1997.

[22] P. Goyal and H. M. Vin, "Fair airport scheduling algorithms," 1997.

[23] P. Goyal, H. M. Vin, and H. Chen, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," Applications, Technologies, Architectures, and

Protocols for Computer Communication, pp. 157-168, 1996.
[24] G. Greenberg and N. Madras, "How fair is fair queuing," Journal of the ACM (JACM), vol. 39, pp. 568-598, 1992.
[25] H. Halabian and H. Saidi, "FPFQ: A Low Complexity Fair Queueing Algorithm for Broadband Networks," 2008.
[26] D. Hogan, "Hierarchical Fair Queueing," in Basser Department of computer science, vol. Doctor of Philosophy. Sydney: University of Sydney, 1997, pp. 123.
[27] D. Hogan, "Hierarchical Fair Queuing," Technical Report 506, Basser Department of Computer Science, University of Sidney, May 1996 1997.
[28] Internet World Stats, www.internetworldstats.com-/stats.htm, 19-03-09.
[29] S. Ju and M. Najar, "A novel hierarchical packet fair scheduling model," 2003.
[30] G. B. Kramer, A. Singhal, N. K. Mukherjee, and B. Ye, "Fair queueing with service envelopes (FQSE): a cousin-fair hierarchical scheduler for subscriber access networks," Selected Areas in Communications, IEEE Journal on, vol. 22, pp. 1497-1513, 2004.
[31] D. Y. Kwak, N. S. Ko, B. Kim, and H. S. Park, "Anew Starting Potential Fair Queuing Algorithm with O (1) Virtual Time Computation Complexity," ETRI journal, vol. 25, 2003.
[32] D. Y. Kwak, N. S. Ko, and H. S. Park, "Mean starting potential fair queuing for high-speed packet networks," 2003.
[33] J. F. Lee, M. C. Chen, and Y. Sun, "WF2Q-M: Worst-case fair weighted fair queueing with maximum rate control," Computer Networks, vol. 51, pp. 1403-1420, 2007.
[34] G. Lu, "Issues and technologies for supporting multimedia communications over the Internet," Computer Communications, vol. 23, pp. 1323-1335, 2000.
[35] G. Lv and X. Zhang, "An Efficient Policy-based Packet Scheduler With Flow Cache," 2007.
[36] J. Nagle, "Congestion Control in IP/TCP Internetworks," 1984.
[37] J. Nagle, "RFC-896: Congestion Control in IP/TCP Internetworks," Request For Comments, 1984.
[38] J. Nagle, "On Packet Switches with Infinite Storage," Communications, IEEE Transactions on [legacy, pre-1988], vol. 35, pp. 435-438, 1987.
[39] K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," IEEE/ACM Transactions on Networking (TON), vol. 1, pp. 344-357, 1993.
[40] H. Shi and H. Sethu, "An evaluation of timestamp-based packet schedulers using a novel measure of instantaneous fairness," 2003.
[41] H. Shi and H. Sethu, "Greedy fair queueing: A goal-oriented strategy for fair real-time packet scheduling," 2003.
[42] M. Song, N. Chang, and H. Shin, "A new queue discipline for various delay and jitter requirements inreal-time packet-switched networks," 2000.
[43] M. Song, J. Song, and H. Li, "Implementing a high performance scheduling discipline WF2Q+ in FPGA," 2003.
[44] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," IEEE/ACM Transactions on Networking (TON), vol. 6, pp. 175-185, 1998.
[45] Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks," 1998.
[46] Stoica, H. Zhang, and T. S. E. Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time and priority services," 1997.
[47] [46] S. Suri, G. Varghese, and G. Chandranmenon, "Leap forward virtual clock: a new fair queuing scheme with guaranteed delays and throughput fairness," In Proceedings of INFOCOM&# 039; 97, 1997.
[48] P. Valente, "Exact GPS Simulation with Logarithmic Complexity, and its Application to an Optimally Fair Scheduler," 2004.
[49] Varma and D. Stiliadis, "Hardware implementation of fair queuing algorithms for asynchronous transfer mode networks," IEEE Communications Magazine, vol. 35, pp. 54-68, 1997.
[50] Wang, K. Long, X. Gong, and S. Cheng, "SWFQ: a simple weighted fair queueing scheduling algorithm for high-speed packet switched network," 2001.
[51] E. Wrege and J. Liebeherr, "A Near-Optimal Packet Scheduler for QoS Networks," 1997.
[52] Xiaohui, L. Jiandong, and G. Feng, "Two simple implementation algorithms of WFQ and their performance analysis," 2001.
[53] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," PROCEEDINGS-IEEE, vol. 83, pp. 1374-1374, 1995.
[54] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines," 1991.
[55] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks," ACM Transactions on Computer Systems, vol. 9, pp. 101-124, 1991.