Internetworking 2002

Fair queuing in data networks

Rodrigo Sierra

Abstract

Working on my Master Thesis project I have dealt with different schemes to schedule fairly the resources between contending flows in an intermediate point within a network. I have found interesting how a router chooses among the flows to serve them trying to satisfy their requirements. Therefore, I have written this paper as an assignment for Internetworking, a course given at KTH during the winter 2002.

A brief description of several fair scheduling systems will be presented, showing the changes the wireline schemes have to manage to apply them in a wireless network.

Introduction

In a data communication network, packets belonging to different traffic flows often share links in the way to their destination. When a node cannot send all the packets it receives in a particular moment, packet queues are originated. So flow control procedures are required to manage these queues by regulating the order and amount of data each source can transmit.

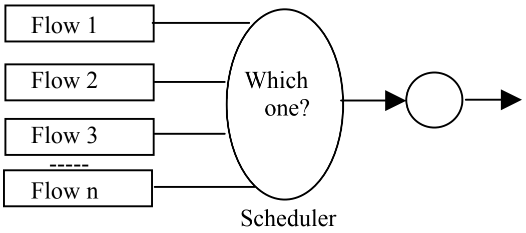

Figure 2. Scheduler

One desirable property for this control system would be to grant each user a minimum rate, sharing the available bandwidth between the different flows as fair as possible and taking into account delay and error restrictions. The responsible for achieving this work is the scheduling agent of the net.

Assuming we can serve only one flow at the same time, static time scheduling, where each flow has a predefined moment to send data, misuses the resources since it does not adapt their assignment to the network necessities. So packet scheduling has been research, serving one packet from the selected flow and deciding which one will be the next selected.

Wireline scheduling

Wireline packet scheduling, also called fluid fair queuing, deal with the time to assign the available resources to the contending flows.

Intuitive schemes

Packets are served according to their arrival time in the most commonly implemented and intuitive discipline, First-Come-First-Served (FCFS). The only parameter that determines the packet allocation in the output link is the order of arrival. Therefore, a burst source can monopolize the network during long periods, leading the system to an unfair situation.

Another intuitive scheduler is the Packet-Based Round Robin (PBRR), where packets are served in a round robin fashion. Although its low complexity, if some flows send packets longer than the rest ones, it will derive in an unfair distribution of the resources.

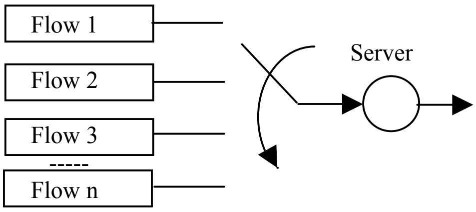

Figure 3. Packet Based Round Robin

Ideal scheme

An ideal scheduler has been proposed in [1], [2]: the Generalized Processor Sharing (GPS). It visits each queue at least once during any finite time interval, processing an infinitesimal amount of data from each one. Although its ideal properties, it is unimplementable since there is no way to serve more than one flow at the same time.

FQ schemes

Trying to emulate the ideal GPS scheduler, Fair Queuing (FQ) algorithms stamp each arriving packet an estimation for the instant the packet should leave the router (virtual finish time) and serve them sorting among the flows according to this number.

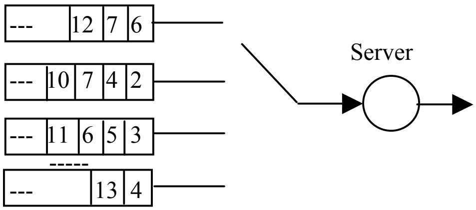

Figure 4. Fair Queuing

Weighted Fair Queuing (WFQ) [3] was the first FQ implementation proposed, followed by several ones that try to simplify its computational complexity or to improve its performance. We can cite Worst-case Weighted Fair Queuing (WF^{2}Q) [4], the most fair packet-by-packet scheduler known; or two more modern versions: Self-Clocked Fair Queuing (SCFQ) [5] and Start-time Fair Queuing (SFQ)[6], that have achieved a simpler implementation, though less fair than WF^{2}Q. Other FQ algorithms like Frame-based Fair Queuing (FFQ) and Starting Potential-Based Fair Queuing (SPFQ) [7] have tried to reduce the complexity of timestamp computations. However, due to the sorting among stamped packets belonging to n different queues, all the algorithms have at least a complexity characterized by O(log n) per packet.

Despite the excellent fairness measure of the FQ algorithms and their delay bounds for the received packets, they are not very suitable for high-speed networks since there is another family of algorithms with lower computational complexity.

This family is based on PBRR. It serves the flows according to a strict round robin order, so the computational complexity can be reduced until O(1) per packet.

To eliminate the unfairness of PBRR, algorithms as Deficit Round Robin (DRR) [8] and Superplus Round Robin (SRR) [9], a modified version of DRR, utilize a *deficit counter* that measures the past unfairness (A deeper definition can be found in the references). However, they need to know the upper bound of the length packets if they want to assure O(1) complexity.

Elastic Round Robin (ERR) [10] suppresses this requirement and assures a computational complexity O(1) per packet without the knowledge of the packets length.

New versions improve their based algorithm; for example, Dynamic Deficit Round Robin (DDRR) [11] uses an adaptive granularity for the deficit counter satisfying both high throughput and small delay requirements for short packets (not achieved by DDR) and Weighted Round Robin (WRR) has different privileges for different flows and share the resources according to these privileges.

Despite their different implementations, their main characteristic is their low complexity (significantly lower than FQ algorithms) at the cost of a worse fairness than the FQ ones.

The fairness of a FQ algorithm and the complexity of a RR one would be a desirable combination. Fair Queuing with Round Robin (FQRR) has tried to achieve this. According to their researchers [12], "It is a round robin algorithm which utilizes some of the concepts developed in connection with the FQ algorithm by Demers *et. al.* in [3]". This scheduler achieves better fairness than the Round Robin algorithms with the same complexity O(1) than them.

A comparison of the different schemes is made in [12] taking one scheduler from each family:

Algorithm	**Fairness Measure**	**Complexity (per packet)**
PBRR	Infinite	O(1)
FQ	MAX	O(log n)
DRR	3MAX	O(1)
FQRR	2MAX	O(1)

The table gives us a general vision of the different algorithm properties. The fairness measure is defined as the difference in throughput at any time, in any queue for any arrival pattern between the measured algorithm and the ideal GPS (In the table MAX is the maximum packet size).

Wireless scheduling

Wireline packet scheduling has been intensely researched, but when we try to apply these algorithms to a wireless system we find new problems to manage:

1. Channel errors are location dependent and burst
2. Channel contention between multiple entities
3. The schedule agent has a limited knowledge of the channel conditions
4. The scheduling must manage both, downlink and uplink flows
5. Mobile terminals are often constrained in terms of power

Flows in wireline networks always transmit in their turn, but if the channel is erroneous, then the retransmission is cursed as a new packet in the queue. This is a waste of bandwidth and a delay increment we cannot afford in a wireless network, where the percentage of retransmissions is much higher than in a wireline one. We need new methodologies to fight these differences, distinguishing between non-backlogged flows (flows that have not any packet to transmit) and backlogged flows that cannot transmit because of the erroneous channel. The critical issues to solve are the failure of the traditional wireline scheduling in a location dependent channel and the compensation for flows that perceive a channel error.

Wireless fair queuing tries to emulate wireline fair packet queuing when the channel is in an error free state, swapping and compensating flows when the channel is erroneous for any of the users. When a flow is in a burst error state, another flow that is in an error free channel state transmits instead (swapping). The difference between the promised rate and the real one is measure (lead-and-lag) and any compensation mechanism is applied later to invert the swapping process.

T. Nandagopal, S. Lu and V. Bharghavan have made in [13] a comparative study of seven recently proposed wireless fair queuing algorithms: IWFQ (Idealized Wireless Fair Queuing) [14], SBFA (Server Based Fairness Approach) [15], CIF-Q (Channel-condition independent fair queuing) [16], WFS (Wireless Fair Server) [17], WPS (Wireless Packet Scheduling) [14], CSDPS (Channel State Dependent Packet Scheduling) [18] and CBQ-CSDPS [19].

- *Error free service*
  The two first have WFQ as their error free reference, CIF-Q and WFS uses different FQ implementations, STFQ and an enhanced version of FQ respectively, while the rest choose WRR (Weighted Round Robin). This selection shows us an approximation for the algorithm complexity.
- *Lead-and-lag and compensation*
  Different models are followed to deal with error channels by the different schemes (see [13]). All of them, except CSDPS, measure the difference between the ideal packet service in an error free channel and the real one (lead-and-lag) to compensate the unbalanced flows when the channel allows them to.
- *Conclusions*
  CSDPS does not provide fairness guarantees since it does not take into account any measurement of erroneous flows. IWFQ and WPS present coarse short-term fairness and throughput bounds. CBQ-CSDPS and SBFA cannot provide short-term fairness. The paper concludes that CIF-Q and WFS achieve short-term and long-term fairness, short-term and long-term throughput bounds, and tight delay bounds for channel access.

We can take a look to WFS, one of the best schemes according to the paper. We will focus our attention in the lead-and-lag model and compensation to know how a wireless scheduler optimizes a wireline one.

Lead and lag model

We name lagging flows to that ones receiving less service than in an error free channel situation. By the contrary, leading flows receive more service than in the error free case. According to the paper: "If a backlogged flow perceiving an error channel is allocated the channel, its lag is increased only if there is another flow that can transmit in its place and increase its lead (or reduce its lag). In effect, the number of slots the flow is entitles to make up in the future, and the lead of a flow reflects the number of slots it must relinquish in the future."

Compensation model

We should put back the stolen slots from the leading flows to the lagging ones, but it must be done in a manner to ensure a minimum service to the leading flows. To attain a graceful degradation for the leading flows, WFS will relinquish a fraction l/l_{max} of the slots allocated to them by error-free service (where l is the lead for a leading flow and l_{max} is its lead bound). An exponential reduction in the number of slots relinquished is achieved. To assign the relinquish slot to a lagging flow, WRR (Weighted Round Robin) is used, being the lag of each flow the weight for the algorithm.

Thanks to this changes, the scheduler does not waste bandwidth because of an error channel and assign fairly the resources among the contending flows.

Conclusions

The different behaviors among different flows in a data communication network lead FCFS and PBRR schedulers to unfair situations. If all the flows have the same packet length, rate, traffic pattern… the bandwidth assignment will be ideal, but this situation is not usual. From that fact, new schemes have been studied to combat this unfair situation.

Wireline schedulers assume an error-free channel or a channel where errors are not very usual. Furthermore, they do not distinguish between flows that need service (backlogged) and flows that are backlogged but cannot transmit due to any error in the channel. They also presuppose the server that schedules the system has a complete knowledge of the channel conditions. According to these assumptions, they guaranteed a minimum throughput for backlogged flows, a bounded delay channel access and fairness among backlogged flows. However, under location dependent and burst errors (wireless networks properties) these schedule models are neither fair nor able to guarantee a minimum throughput.

To assure a fair situation between flows in a wireless environment, the scheduler bases its decision on the channel state. If the selected flow cannot transmit because of an error in the channel, another flow will transmit instead and the first one will recover its turn in a later moment, when it has a clean channel.

References

[1] A.K. Parekh, R.G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks-the single node case”, *IEEE INFOCOM ’92*, 1992, pp. 521-530

[2] A.K. Parekh, R.G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks-the multiple node case”, *IEEE INFOCOM ’93*, 1993, pp. 915-924

[3] A. Demers, S. Keshav and S. Shenkar, "Analysis and Simulation of a Fair Queuing Algorithms," *Proceedings of SIGCOMM ’89*, August 1989, pp. 3-12

[4] J. C. R. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queuing," *Proceedings of INFOCOM ’96,* March 1996, pp. 120-128

[5] S.J. Golestani, “ A Self-Clocked Fair Queuing Scheme for Broadband Applications”, *Proc. IEEE INFOCOM*, June 1994, pp. 636-646

[6] P. Goyal, H. Vin, and H. Cheng, “Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks”, *Proc. IEEE SIGCOMM 1996*, pp 157-168.

[7] D. Stiliadis, A. Varma, “Efficient Fair Queuing Algorithms for Packet-Switched Networks”*, IEEE Trans. Networking*, Vol. 6, No. 2, April 1998, pp. 175-185

[8] M. Shreedhar and G. Varghese, "Efficient Fair Queuing using Deficit Round Robin," *Proceedings of SIGCOMM’95,* August 1995.

[9] H. Adiseshu, G. M. Parulkar, and G. Varghese. “A Reliable and Scalable Striping Protocol*”. Proceedings of the ACM SIGCOMM*, August 1996.

[10] S. Kanhere, H. Sethu and B. Parekh “Fair and Efficient Packet Scheduling Using Elastic Round Robin”. *IEEE Transactions on parallel and distributed systems*, Vol. 13. No. 3, March 2002. pp 324-336.

[11] K. Yamakoshi, K. Nakai, E. Oki and N. Yamanaka “Dynamic deficit round-robin scheduling scheme for variable-length packets”. *IEE Electronics letters,* Vol. 38, No. 3, 31st January 2002, pp 148-149.

[12] Sen, I. Mohammed, R. Samprathi, and S. Bandyopadhyay “Fair Queuing with Round Robin: A New Packet Scheduling Algorithm for Routers” *IEEE ISCC’02*, 2002

[13] T. Nandagopal, S. Lu and V. Bharghavan “A unified architecture for the design and evaluation of wireless fair queuing algorithms” *ACM MOBICOM ’99*, August 1999.

[14] S. Lu, T. Nandagopal and V. Srikant, “Fair Scheduling in Wireless Packet Networks”, *ACM MOBICOM’98*, October 1997.

[15] P. Ramanathan, P. Agrwal, “Adapting Packet Fair Queuing Algorithms to Wireless Networks”, *ACM MOBICOM’98*, October 1998.

[16] T.S. Ng, I. Stoica and H. Zhang, “Packet Fair Queuing Algorithms for Wireless Networks with Location-dependent errors” *IEEE INFOCOM’98*, March 1998.

[17] S. Lu, T. Nandagopal and V. Bharghavan, “Fair Scheduling in Wireless Packet Networks”, *ACM MOBICOM’98*, October 1997.

[18] P. Bhagwat, P. Bhattacharya, A. Krishma and S. Tripathi, “Enhancing throughput over wireless LANs using channel state dependent packet scheduling”, *IEEE INFOCOM’97*, April 1996

[19] M. Srivastava, C. Fragouli, and V. Sivaraman, ”Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling”, *IEEE INFOCOM’98*, March 1998.