Performance Evaluation of Back-Pressure Fairness in RPR

Stein Gjessing and Fredrik Davik
Simula Research Laboratory
Fornebu, Norway

Email: steing@simula.no, bjornfd@simula.no

1. Introduction

A ring network needs a fairness algorithm that regulates each station's access to the ring. In this paper we explore some of the properties of back-pressure flow control to achieve fair allocation of ring bandwidth between stations. We investigate the stability of different variants of back-pressure flow control, and suggest improvements to the fairness algorithm proposed by the RPR working group. We have developed a discrete event simulator in the programming language Java and use this simulator to evaluate the performance of the discussed back-pressure algorithms using different traffic scenarios.

Ring networks have been extensively used and studied in the literature, notably the Cambridge Ring, IEEE Token Ring and FDDI [18, 10, 21]. Around 1990, destination stripping with spatial reuse was exploited in systems like SCI, MetaRing, DQDB, ATMR, and CRMA-II [12, 6,11,13,15]. Access to the ring must be controlled by a fairness- or media access control (MAC) algorithm, and several have been proposed and evaluated [2,5,9,16,17,20]. Cisco developed Dynamic Packet Transport with a fairness algorithm called Spatial Reuse Protocol (SRP) [23]. Parts of the fairness algorithm currently under consideration by the IEEE P802.17 RPR working group is heavily based on SRP.

In this paper we start with a brief fairness discussion. Then we discuss back-pressure in a dual ring topology and suggest some improvements to the back-pressure fairness algorithm proposed by the RPR working group. In section 3 we describe our experimental platform (an RPR discrete event simulator written in Java), and in section 4 we discuss and evaluate our proposals for improvements using this platform. In section 5 we conclude and point out directions for further work.

2. Fairness and Back-Pressure

If we define a flow to be the traffic between a given pair of source and destination stations, we could define a fairness concept based on the goal that all flows get the same capacity (when there is a shortage).

The RPR working group has tacitly assumed another fairness model based on the assumption that if there is a capacity shortage, all stations should be able to send the same amount of data, regardless of destination address. Ed Knightly has made a formal definition of this in his RIAS (Ring Ingress-Aggregated with Spatial Reuse) fairness concept [25]. In this paper we base our notion of fairness on his definition.

The basic idea behind a back-pressure-based fairness protocol in a RPR-like ring is that stations unable to send due to congestion send a flow control packet upstream to ask its neighbors to reduce their send-rate. There are at least two strategies that can be used:

1. Ask upstream neighbors to reduce their send-rate down to or below a level where congestion is certainly avoided. If this results in underutilization of the link capacity, stations may increase their send-rate later.
2. Ask upstream neighbors to reduce their send-rate somewhat. If the reduction was insufficient, a new reduction request is issued.

Strategy 1 above is used by the RPR working group. In this paper we investigate properties of this strategy as well as a variant of strategy 2.

RIAS fairness defines that when link capacity is a scarce resource, stations must reduce their send-rate so that they send the same amount of data over the congested link. To achieve this, individual stations require knowledge of the amount of data transmitted by other stations. One method to obtain this knowledge is to include the station's send-rate in the flow control packets sent to upstream neighbors.

When upstream stations react to flow control packets and reduce their send-rate, the downstream stations' transit buffers are emptied and congested stations can increase their send-rate. As long as an upstream station receives flow control packets, it adjusts to the send-rate advertised in these packets. When a station does not receive flow control packets, it gradually increases its send-rate again.

The RPR fairness algorithm and its associated parameter values as suggested by the RPR working group is referred to as the Original algorithm. For details, refer to Equation 1 below.

We believe that the fairness algorithm proposed by the RPR working group is overly cautious and that it adapts too slowly to changes in network traffic. E.g. if a station that has not sent traffic for a long time starts to send and get congested, its historic send-rate will be close to zero. This influences the Original fairness algorithm in a negative way, because an upstream station must decrease its send-rate to the value received from its downstream neighbor (which is close to zero in this case). This results in parts of the ring's bandwidth resources being underutilized (while waiting for the send-rate to be increased to the bandwidth available).

In this paper we propose two methods (one for each of the two strategies above) to make back-pressure work better. Details of the methods and their associated parameter values are shown in Equation 1 below.

1. **Aggressive algorithm**: Let the newer values count much more when the low pass filtered version of a station's usage value is computed, and also use a much larger increment when the increase in traffic is computed.
2. **Reduce algorithm**: Use a back-pressure fairness algorithm where upstream stations reduce their send-rate in smaller decrements. If the reduction was insufficient, the congested state of downstream stations will prevail, and new flow control packets will be sent until the congested state ceases.

$$lp\_usage = \frac{lp\_usage \cdot (LP - 1) + usage}{LP}, \quad LP = \begin{cases} 16 \ (Agressive \ \ and \ Reduce) \\ 512 \ (Original) \end{cases}$$

$$increment = \frac{1}{INC\_RATE} \cdot (max\_rate - max\_allow), \quad INC\_RATE = \begin{cases} 4 \ \ (Reduce) \\ 16 \ (Aggressive) \\ 64 \ (Original) \end{cases}$$

$$max\_allow = \begin{cases} adverticed & (Original \ and \ Aggressive) \\ max\_allow - \dfrac{max\_allow - adverticed}{2} & (Reduce) \end{cases}$$

lp_usage: low pass filtered version of a station's send rate and the send rate advertised in the flow control packets to upstream neighbors.

adverticed: *lp_usage* value received from downstream station.

usage: the number of bytes the station has sent from its ingress buffer the last 100 microseconds.

max_allow: this station's maximum allowed send rate.

increment: value to use when incrementing *max_allow* if the station has not received a flow control packet the last 100 microseconds.

max_rate: maximum link rate.

Equation 1. The Fairness Algorithm, Equations and their Assosicated Parameter Values.

3. Performance Scenarios and Performance Evaluation Platform

Ed Knightly has developed 5 scenarios that can be used to test whether RIAS fairness is achieved [25]. Only four of the five scenarios are relevant for evaluating back-pressure (the fifth uses only one sender).

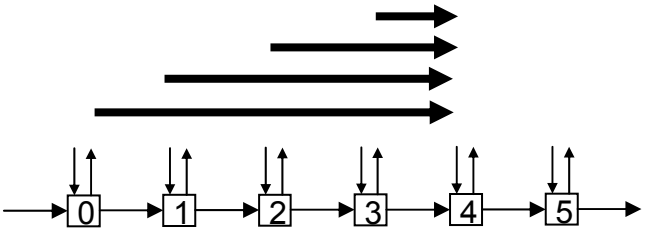


Figure 1. Parking lot scenario

The basic traffic scenario is shown in figure 1. Here a number of stations (four) are sending to the same destination station (station 4). By RIAS definition, they should share the bandwidth of the most congested link, i.e. the link from station 3 to station 4 ($l_{3\text{-}4}$), equally. This means that they should all send at a rate equal to one quarter of the full bandwidth of $l_{3\text{-}4}$.

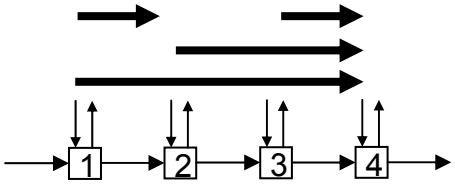


Figure 2a
Parallel parking scenario

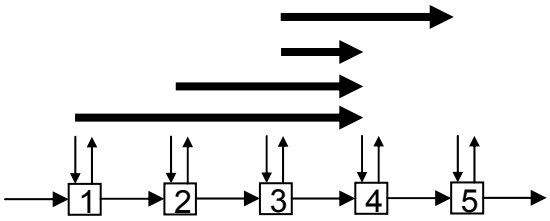


Figure 2b
Two-exit parking scenario

In the parallel parking scenario in figure 2a, the three flows into station 4 should share the bandwidth of $l_{3\text{-}4}$ equally, i.e. they should get one third of the bandwidth each. If station 1 sends data to station 4 at one third of the full bandwidth of $l_{3\text{-}4}$, it may send data to station 2 at two thirds of the full bandwidth of $l_{1\text{-}2}$.

In the two-exit parking lot scenario of figure 2b, the bandwidth of $l_{3\text{-}4}$ should be shared equally between all sending stations. This means that each of stations 1 and 2 should be able to send at one third of full bandwidth of $l_{3\text{-}4}$, while the flow from 3 to 4 as well as the flow from 3 to 5 should operate at one sixth of full bandwidth of $l_{3\text{-}4}$.

We have used a 16-station dual ring for all experiments reported in this article. However, the experiments use a few stations only and traffic is sent on one of the rings only. Ring behavior when both rings are utilized for data traffic, and also when there is traffic all the way around the ring is reported in [24].

The distance between stations in our model is approximately 2000 meters. This gives a propagation delay on one link (including the transit delay through one station with an empty transit buffer) of about 10 microseconds. With a non-empty transit buffer, the station pass-through time will vary.

Of the three priority levels implemented in our RPR-model, only two are used in the reported experiments. Our transit-path model includes two buffers, one for high priority- and one for low- and medium priority traffic. The link speed used is one Gbyte/sec (for all links). The data packets used are low priority and 500 bytes long [7]. The flow control packets used are 32 bytes high priority packets. The (medium and low priority) transit buffer of a station is said to be congested when the number of bytes in the buffer is above a threshold set to 12,500 bytes. The size of the transit buffer is 40.000 bytes.

The RPR model is a Java program, consisting of approximately two thousand code-lines. The main components of the model have been running since April 2001, and have been verified and tested for a number of different traffic scenarios.

4. The Results of the Experiments

The main observation when comparing results for the three scenarios (from figure 1, 2a and 2b) is that the Original algorithm converges much slower than the two others. As will be shown, all algorithms can cause controlled oscillations under certain conditions, but we have not seen results indicating that oscillations caused by the Original algorithm is smaller in magnitude than the Aggressive- or the Reduce algorithms. In the sequel we report on the most interesting and explanatory results.

The three algorithms were first applied to the scenario in figure 1. The four flows attempt to send at full link speed. With the Original algorithm they converge (to one quarter of the link capacity each) in approximately 70 microseconds. The Aggressive and the Reduce algorithm converge much quicker, in about 10 microseconds. We do not show plots for these runs.

Performance graphs for the three algorithms applied to scenario 2a are shown in figures 3, 4a and 4b. The flow from station 1 to station 2 is not started until time t=100 ms in fig. 3 and t=15 ms in figure 4. Hence we first see (from time 0 to time 100 and 15 ms respectively) the three other flows competing for the link capacity of $l_{3\text{-}4}$. This initial behavior is similar to scenario 1, except now there are only three flows.

In the run depicted in figure 3 (the Original algorithm), station 3 has sent flow control packets to stations 1 and 2 and asked them to reduce their send-rate. The usage value sent in this packet is very low, because station 3 has had the opportunity to send very little before it sends the first flow control packet. Station 2 is sending somewhat less than station 1 because it has two reasons not to send: **(1)** reception of flow control packets from 3, **(2)** filling of its transit buffer with packets from station 1. Station 1 does not experience the latter problem. Because the flow control works much faster in figure 4 (the Aggressive- and the Reduce algorithms), the described effect is much less noticeable here.

Notice also in figure 4b how the Reduce back-pressure algorithm gradually reduces the traffic sent from station 1 (from t=0 to t=11 ms). But also notice that this reduction is much faster than the reduction of the flow from station 3 seen in figure 3.

Figures 3 and 4 also show what is happening when the flow from station 1 to station 2 is introduced at respectively 100 and 15 ms. Station 1 then starts sending every other packet to stations 2 and 4 respectively. What we observe here is the Head of Line blocking (HOL) effect -- a packet destined for station 4 is at the head of the single-queue ingress-buffer of station 1. Due to the congested state of the bottleneck link ($l_{3\text{-}4}$), the packet can't be transmitted. Also, it blocks for packets in the queue destined for stations reachable through uncongested links (i.e. station 2). The result is underutilization of the available bandwidth on $l_{1\text{-}2}$. In fact figures 3 and 4 shows that the complete system is now oscillating.

In general, oscillation in a feedback-based system occurs when the system fails to enter a stable state. The reason why a stable state is not found in this case is that: Assume that the system is close to a stable state, but that station 2 or 3 is a little congested. Stations 2 and 3 send flow control packets upstream to station 1 and ask it to reduce it traffic over $l_{3\text{-}4}$. Station 1 reduces its traffic to the advertised value. The resulting reduction is too large (since half of its traffic is sent locally, over the uncongested $l_{1\text{-}2}$, to station 2). When station 1 sends less than anticipated over $l_{3\text{-}4}$, stations 2 and 3 increase their send-rate (they try to send at full link speed all the time). The result is that the system moves further away from the stable state (equal division between stations of the bandwidth resources of the most congested link, $l_{3\text{-}4}$), resulting in large oscillation every time $l_{3\text{-}4}$ becomes congested and new flow control messages are sent upstream by station 2 or 3.

In this case it seems like HOL blocking and the fact that the station had no per station knowledge about how much it was allowed to send, caused oscillations.

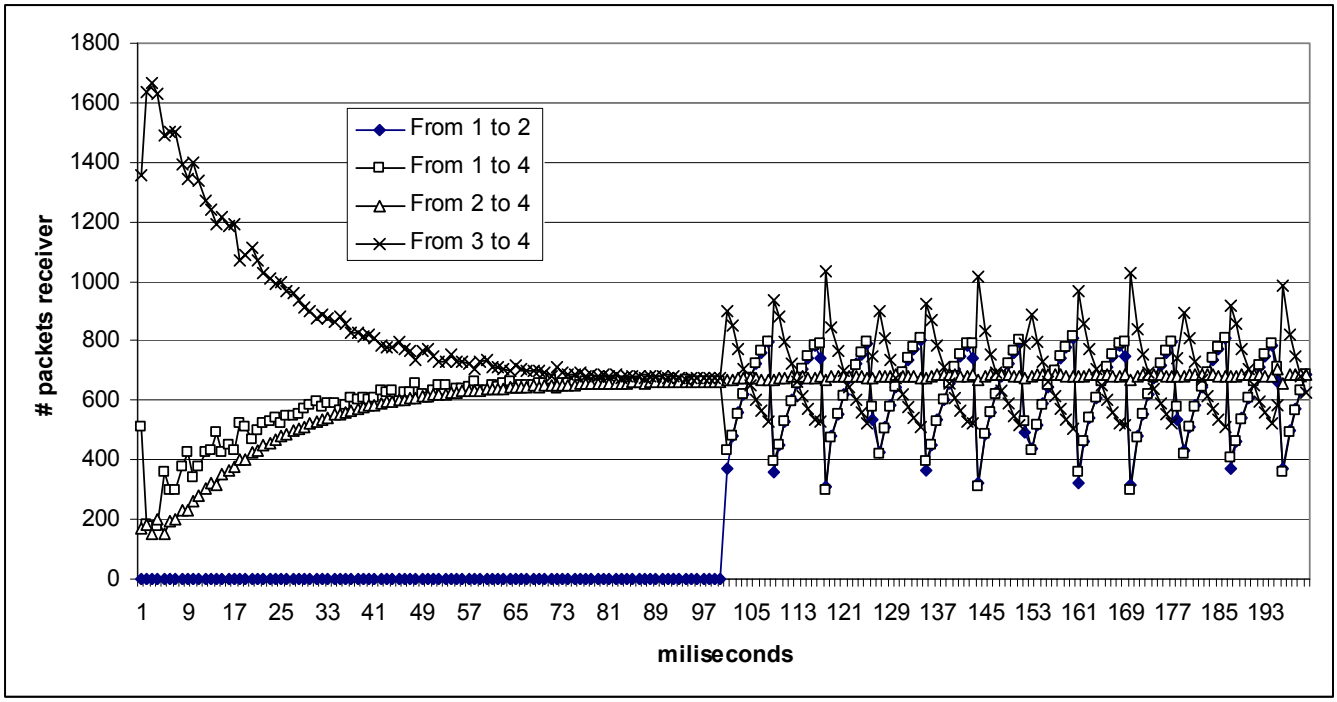
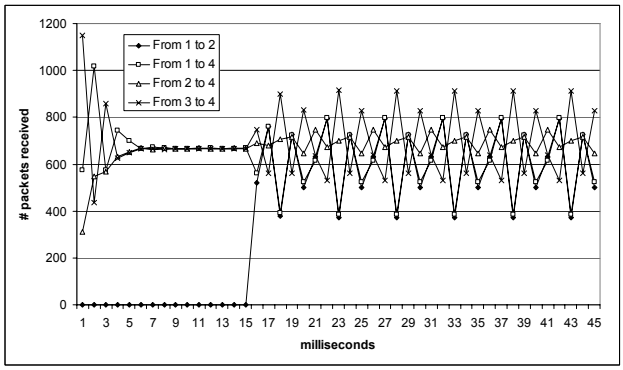


Figure 3. Scenario 2a with the Original algorithm. The flow from 1 to 2 is not started until t=100 ms.



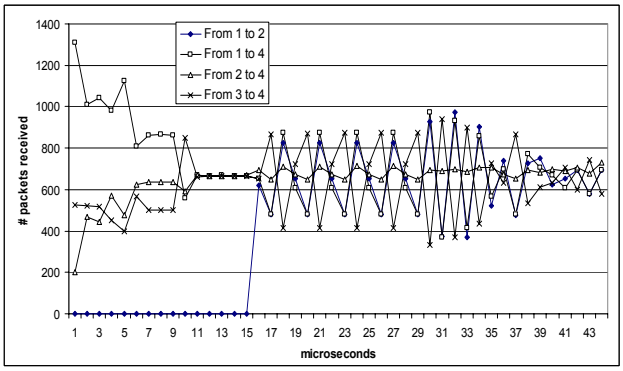


Figure 4a . Figure 4b

Scenario 2a with the Aggressive (4a) and the Reduce (4b) algorithm. The flow from 1 to 2 is not started until t=15 ms.

Finally, results from a combined scenario are shown in figure 5. Until t=120ms, scenario 1 is used with three flows running from stations 1, 2 and 3 to station 4. At t=120 ms, station 1 starts sending at two thirds of the link capacity to station 2, while continuing to send at one third of the link capacity (666 packets per ms) to station 4. The plot shows that this only gives the added effect of 1333 packets per ms transmitted from station 1 to station 2. The next event is at t=150 ms when station 3 starts to send every other packet to station 5. The plot shows that about 333 packets per ms are sent from 3 to 4 and from 3 to 5. At t=200 ms, station 1 reverts to its original pattern of sending at full link speed to station 4 only, and finally, at t=250 ms, station 3 also reverts to its original pattern of sending to station 4 only. At this time, the traffic pattern is back

to its starting point. The change that started at t=120 ms is repeated over again starting at t=300 ms.

This experiment was run with the Original as well as the Aggressive algorithm. They both reached stable states, but not even the Aggressive algorithm converged as fast as the Reduce algorithm (fig. 5).

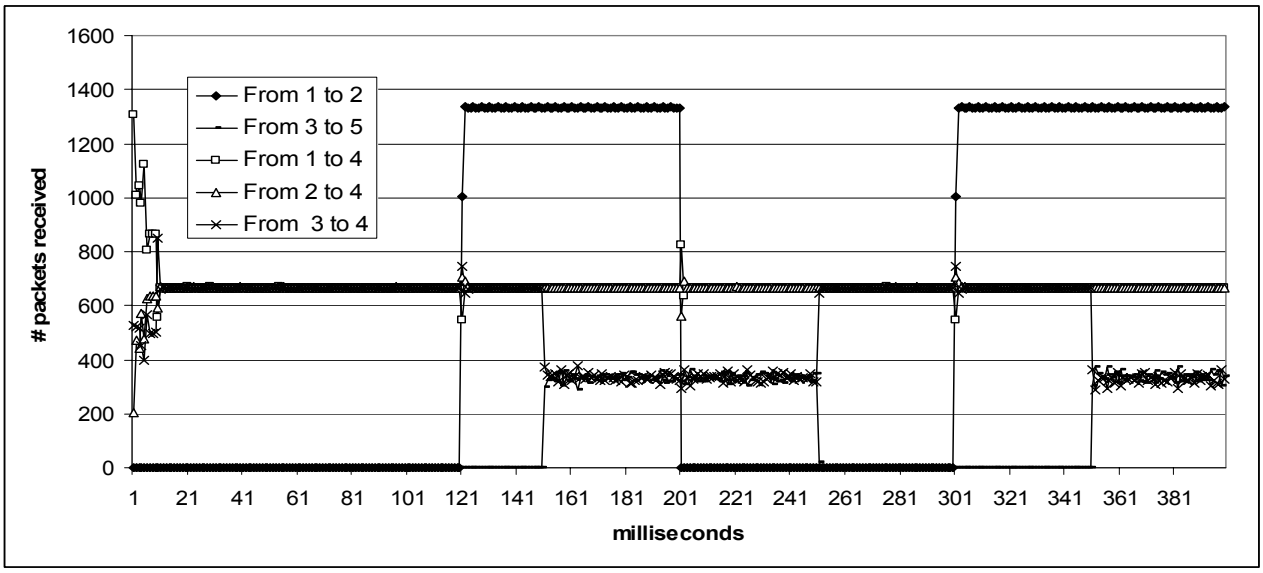


Figure 5 A combined scenario of 1, 2a and 2b with the Reduce algorithm.

5. Conclusions and Further Work

It has been demonstrated that back-pressure flow control may give stations fair access to the ring in accordance with the RIAS fairness principle. The use of more aggressive aging parameters (i.e. newer values have more weight) seems to be required in adapting to quick flow changes. Also, our Reduce algorithm, where stations only reduce their send rate somewhat, appears to be a promising technique.

The instabilities reported in this article seems to stem from the fact a system with HOL blocking has a more complex stable state than can be regulated by single flow control messages. We want to investigate further the stability of a RPR network having stations using non-HOL blocking ingress queues.

Hence, we need to run more experiments with our network simulator model to learn more. On a longer time scale we want to run execution driven simulations and investigate how an RPR-based network behaves when running user level applications.

References

1. ANSI T1.105.01-2000: Synchronous Optical Network (SONET) - Automatic Protection.
2. H.R. van As: Major Performance Characteristics of the DQDB MAC Protocol. Telecommunications Symposium, 1990. ITS'90 Symposium Record, SBT/IEEE 1990
3. S. Breuer, T.Meuser: Enhanced Throughput in Slotted Rings Employing Spatial Slot Reuse. INFOCOM '94. Networking for Global Communications. IEEE. 1994
4. I. Cidon, L. Georgiadis, R. Guerin, Y. Shavitt: Improved fairness algorithms for rings with spatial reuse.
INFOCOM '94. Networking for Global Communications. IEEE, 1994
5. I. Cidon, Y. Ofek: Distributed Fairness Algorithms for Local Area Networks with Concurrent Transmissions. In: Lecture Notes in Comp. Sci., Vol. 392, Springer, 1988
6. I. Cidon, Y.Ofek: MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse.
IEEE Trans on Communications, Vol. 41, No. 1, January 1993.
7. K.C. Claffy: Internet measurements: State of DeUnion.
http://www.caida.org/outreach/presentations/Soa9911
8. M.W. Garrett, S.-Q. Li: A study of slot reuse in dual bus multiple access networks**.**
IEEE Journal on Selected Areas in Communications, Vol. 9 Issue 2, Feb. 1991.
9. A. Grebe, C. Bach: Performance comparison of ATMR and CRMA-II in Gbit/s-LANs.
SUPERCOMM/ICC '94, IEEE Int. Conf. on Serving Humanity Through Communications, 1994.
10. IEEE Standard 802.5–1989, IEEE standard for token ring.
11. IEEE Standard 802.6–1990, IEEE standard for distributed queue dual bus (DQDB) subnetwork.
12. IEEE Standard 1596–1990, IEEE standard for a Scalable Coherent Interface (SCI).
13. ISO/IECJTC1SC6 N7873: Specification of the ATMR Protocol (V. 2.0), January 1993.
14. I. Kessler, A. Krishna: On the cost of fairness in ring networks.
IEEE/ACM Trans. on Networking, Vol. 1 No. 3, June 1993.
15. W.W. Lemppenau, H.R.van As, H.R.Schindler: Prototyping a 2.4 Gbit/s CRMA-II Dual-Ring ATM LAN and MAN**.**
Proceedings of the 6th IEEE Workshop on Local and Metropolitan Area Networks, 1993.
16. M.J. Marsan et al.: Slot Reuse in MAC Protocols for MANs. IEEE J. on Selected Areas in Communications. Vol. 11, No. 8, October 1993.
17. H.R. Muller et al: DQMA and CRMA: New Access Schemes for Gbit/s LANs and MANs. INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. IEEE , 1990.
18. R.M. Needham, A.J. Herbert: The Cambridge Distributed Computing System.
Addison-Wesley, London, 1982.
19. T. Okada, H. Ohnishi, N. Morita: Traffic control in asynchronous transfer mode.
IEEE Communications Magazine , Vol. 29 Issue 9, Sept. 1991.
20. D. Picker, R.D. Fellman: Enhancing SCI's fairness protocol for increased throughput.
IEEE Int. Conf. On Network Protocols. October, 1993.
21. F.E. Ross: Overview of FDDI: The Fiber Distributed Data Interface.
IEEE J. on Selected Areas in Communications, Vol. 7, No. 7, September 1989.
22. I. Rubin, H.-T. Wu: Performance Analysis and Design of CQBT Algorithm for a Ring Network with Spatial Reuse. IEEE/ACM Trans on Networking, Vol. 4, No. 4, Aug. 1996.
23. D. Tsiang, G. Suwala: The Cisco SRP MAC Layer Protocol.
IETF Networking Group, IETF RFC 2892, Aug. 2000.
24. S. Gjessing and B.F. Davik: Avoiding Head of Line Blocking using an Enhanced Fairness Algorithm in a Resilient Packet Ring.
2002 International Conference on Telecommunications (ICT 2002).
25. Edward W. Knightly.: RIAS Fairness Reference Model. http://www.ece.rice.edu/networks/RIAS/