Flow Control in ServerNet^R Clusters

Vladimir Shurbanov[1], Dimiter Avresky[1], Pankaj Mehra[2], and William Watson[3]

[1] Boston University, 8 Saint Mary's St., Boston, MA 02215, USA,
{vash,avresky}@bu.edu
[2] Compaq Tandem Labs, 19333 Vallco Parkway, Cupertino, CA 95014
pankaj.mehra@compaq.com,
[3] Compaq Tandem Labs, 14231 Tandem Blvd., Austin, TX 77728
william.watson@compaq.com

Abstract. This paper investigates the performance implications of several end-to-end flow-control schemes based on the ServerNet^R system-area network. The static window (SW), packet pair (PP), and the simplified packet pair (SPP) flow control schemes are studied. Additionally, the alternating static window (ASW) flow control is defined and evaluated. Previously, it has been proven that the packet-pair scheme is stable for store-and-forward networks based on Rate Allocation Servers. The applicability of a PP flow control to wormhole-routing networks is studied and evaluated through simulation. It is shown that if high throughput is desired, ASW is the best method for controlling the average latency. On the other hand, if low throughput is acceptable, SPP can be applied to maintain low latencies.

1 Introduction

The term flow control refers to the techniques that enable a data source to match its transmission rate to the currently available service rate in the network and at the receiver [9, 11]. Apart from this main goal a flow control mechanism also should adhere to the following requirements: be simple to implement, use a minimum of network resources (bandwidth, buffers, etc.), and operate effectively when used by multiple sources. Additionally, the principles of fairness should be observed for shared resources. And finally, the entire networked system should be stable, i.e., for a constant configuration the transmission rate of each source should converge to an equilibrium value.

This paper considers two closed-loop flow control schemes - the static window and the packet pair flow control protocols. In the static window scheme [12] the source stops transmitting when in has sent a number of unacknowledged (outstanding) request equal to the size of the defined window. The main problem with this approach is that the optimal window size depends on many factors which vary over time and differ among connections. Therefore, choosing a single static window size that is suitable for all connections is impossible. In the packet pair scheme [8] the source estimates and predicts the network conditions based on the delay observed for a pair of consecutive packets and adjusts its transmission

A. Bode et al. (Eds.): Euro-Par 2000, LNCS 1900, pp. 1148–1156, 2000.
© Springer-Verlag Berlin Heidelberg 2000

rate accordingly. The scheme is proved [8] to result in a stable system for store-and-forward networks based on Rate Allocation Servers.

This paper investigates the applicability of packet pair flow control to wormhole-routing networks that are not based on Rate Allocation Servers. Since the packet pair flow control does not limit the maximum number of outstanding requests, the static window protocol is employed in conjunction with it.

Flow Control in the ServerNet SAN.

The ServerNet system area network (SAN) is a wormhole-routed, packet-switched, point-to-point network with special attention paid to reducing latency and assuring reliability [4, 5]. It uses multiple high-speed, low-cost routers to rapidly switch data directly between multiple sources and destinations.

ServerNet implements two levels of flow control - hop-by-hop flow control and end-to-end flow control. Hop-by-hop flow control is performed by the exchange of special flow control flits (busy and ready) between the two devices connected through the link. Busy flits signal that the receiver queue is full. When the transmitting device receives a busy flit it ceases sending data until it receives a ready flit. End-to-end flow control is performed through the static window protocol. In this scheme each request packet has to be acknowledged by a response packet. The size of the static window limits the number of unacknowledged (outstanding) requests that can be transmitted. When a source reaches this limit it ceases transmitting requests until it receives at least one response.

Simulation Model.

The simulation model is discrete-event and unit-time [7]. Each device enters a particular state during each time step. The devices are activated in an random order. All performance measures collected during the course of the simulation are averaged over a number of packets sufficient to achieve the desired level of data accuracy for a confidence level of 95%. Collection begins when the system enters a steady state. Steady state is determined by the method of moving averages presented in [7]. The statistical data produced by the simulator was validated using experimental data collected at Compaq Tandem Labs. Discrepancies between the simulation and experimental results were found to be less that 5%. Since the simulation operates at a data accuracy of 3% these discrepancy are insignificant.

2 Packet Pair Flow Control

The packet pair flow control [9] (PP) belongs to the class of rate-based flow control protocols. PP estimates the conditions in the network by observing the time interval between the receptions of the responses to a pair requests of requests (packet pair) transmitted back-to-back. Moreover, it predicts the future service rate in the network and adjusts incorrect past predictions. The PP flow control is subject to the following limitations: packets must always be transmitted in pairs; the service rate of non-bottleneck servers is assumed to be deterministic.

To circumvent these limitations the simplified packet pair (SPP) flow control defined in [6] is described below.

Implementation of SPP.
The simplified packet pair flow control (SPP) is implemented as follows:

1. The inter-request delay is determined by a variable, I, which is 0 initially. After a packet is transmitted the next packet may not be transmitted before a time period of I expires.
2. The difference between the RTTs of every pair of consecutive packets to/from the same destination is compared with a threshold parameter *delta*. If *delta* is greated a "win" is registered.
3. A history of the last h comparisons is kept.
4. If the number of wins, h_W, are more than $\frac{h}{2}$, I is decremented by a value that depends on h_W. The greater h_W, the greater the decrement.
5. If $h_W < \frac{h}{2}$, I is incremented by a certain value that depends on h_W. The smaller h_W, the greater the increment.

Evaluation of SPP.
Some statistics for the operation of SPP are shown in Table 1. They are based on the topology shown in Fig. 1-a with a Uniform traffic distribution and a generation rate of 200 requests/μs, which is selected to be past the saturation point of the network. Consider the statistics for the number of "wins" (Table 1-a). A window of $h = 8$ comparisons is kept. When the number of "wins" is equal to 4, SPP does not modify the inter-request delay. It can be concluded based on the average number of "wins" and the low deviation that generation rate controlled by SPP converges to an equilibrium state, i.e., the system is stable.

Average:	4.34
Std. Dev.:	1.16
Minimum:	0
Maximum:	8

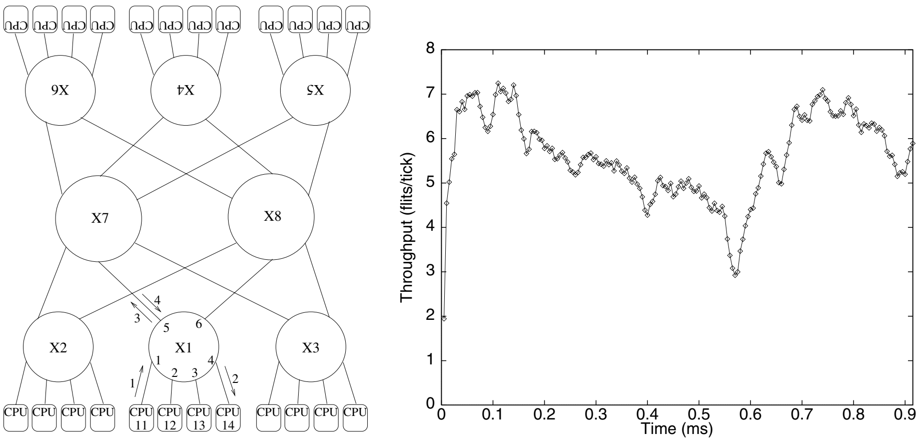
(a) Number of Wins

Average:	25.96
Std. Dev.:	29.47
Minimum:	0
Maximum:	141

(b) Inter-Request Delay (μs)

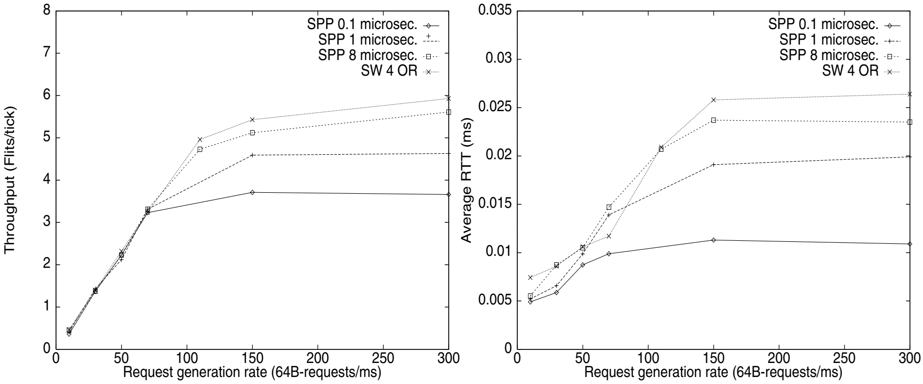
Table 1. Statistics for SPP

The inter-request delay statistics (Table 1-b) show that the SPP algorithm introduces a significant inter-request delay - an average of approximately 26 μs. During this period requests are held at the source devices. By adding the RTT (5.94 μs is the average observed in this case), the average request-to-response time totals approximately 32 μs. This value is higher than the RTT in the absence of the SPP algorithm, where the request-to-response time is equivalent to the RTT of 30.2 μs. It becomes apparent that the RTT is reduced in SPP by introducing an approximately equivalent delay at the source device. Essentially, the SPP scheme changes the location (source device vs. network and destination device) where delays are incurred, but does not reduce the total request-to-response delay.



(a) 24-CPU Topology (b) Throughput snapshots

Fig. 1. Network Topology and Throughput Snapshots



(a) Throughput (b) Round Trip Time

Fig. 2. Performance of SPP, $delta = 0.1, ..., 8\mu s$

The effect of the parameter *delta* on the operation of the SPP algorithm in the 24-CPU topology shown in Fig. 1-a is evaluated for $delta = 0.1, 1, 8\mu s$. The results are presented in Fig. 2 and Table 2. The window size is limited to 4 outstanding requests (OR). The data shows that as *delta* is increased both the throughput and the RTT increase, growing closer to the performance characteristics achieved with the static window (SW) protocol alone. This trend is also observed in the average ORs, shown in Table 2.

It can be concluded that the parameter *delta* essentially limits the generation rate of devices. As *delta* is decreased, SPP introduces higher inter-request delays, thus reducing the generation rate. For low values of *delta*, a less requests are transmitted into the network. This results in low congestion and hence low RTTs. However, the low generation rate also leads to low throughput. Conversely, increasing *delta* leads to increases in both the RTT and the throughput.

	SPP			SW	ASW
	delta (μs)				
	0.1	1.0	8.0		
Throughput ($flits/tick$)	3.6	4.63	5.61	5.93	5.53
Avg. RTT (ms)	10.9	19.9	23.9	26.4	21.9
Avg. OR (packets)	1.07	1.21	3.38	3.5	2.83

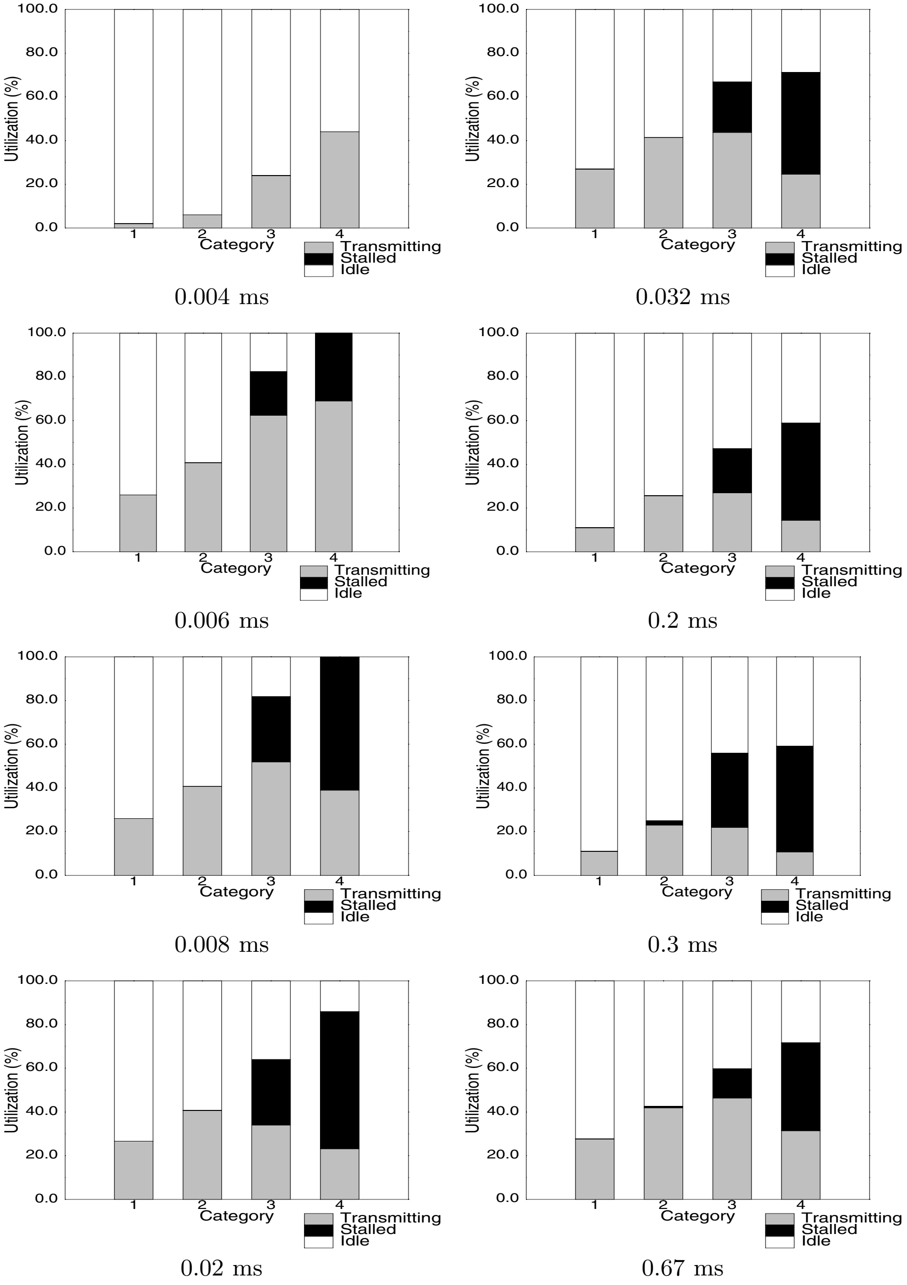
Table 2. Flow Control Schemes: Throughput, Average Round Trip Time (RTT), and Average Outstanding Requests (OR)

3 Alternating Static Window Flow Control

Ideally, the flow control scheme should maintain a high number of ORs to maximize throughput by overlapping (pipe-lining) the request-propagation and the request-processing delays but at the same time it should limit the number of ORs to minimize queueing delays at the end devices. In the SW scheme the number of ORs is maintained at maximum regardless of the delays, which leads to high throughput and high delays. An alternative approach is to halt the generation of requests when the high window mark is reached and to resume generation when the low window mark is reached. Reaching the high window mark is taken as an indication that the RTT is large, i.e., the network is overloaded. While the low window mark indicates that a sufficient amount of requests have been processed and it can be assumed that the network load has decreased to an acceptable level. It can be expected that this scheme will maintain high throughput because it pipe lines the requests, however it should lead to reduced queueing delays since high delays are detected and generation is halted until the network load is relieved.

To further support this conjecture we analyze the dynamic behavior of the network characteristics, based on the throughput and link usage snapshots shown in Figs. 1-b and 3. The link categories used in Fig. 3 are specified in Fig. 1-a. The following observations are made:

- initially (0.004 ms) there is no stalling of the links and transmission is not at 100%; this occurs because the transfer of data from memory to the interface has a start-up delay and the interface is not fully utilized;
- next (0.006 ms) there is more data available than the capacity of link 4 and stalling is observed;



0.004 ms

0.032 ms

0.006 ms

0.2 ms

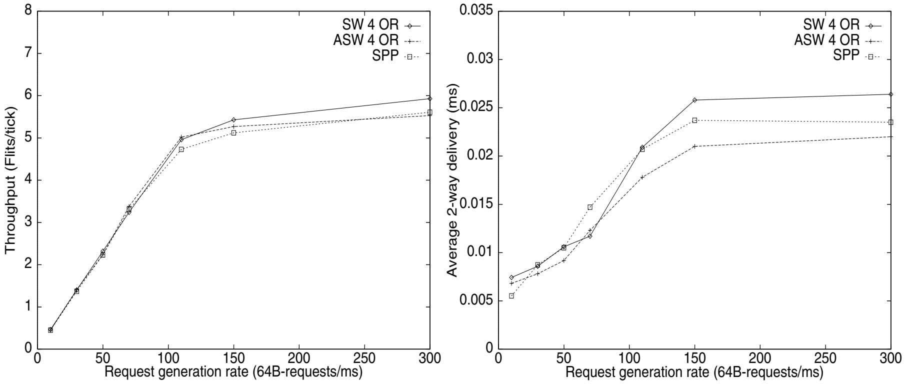
0.008 ms

0.3 ms

0.02 ms

0.67 ms

Fig. 3. Link Usage Snapshots for Static Window Flow Control



(a) Throughput

(b) Round Trip Time

Fig. 4. Performance of Flow Controls: (1) ASW, 4-0 OR; (2) SW, 4 OR; (3) SPP, $delta = 8\mu s$

• the stalled time continues to increase until the static window limit (SWL) is reached and request transmission is halted, this causes idle time to appear at 0.01 ms and increase in proportion from there on; the idle time does not appear due to lack of data since it is constantly available; the increasing stalled time causes increasing delivery times;
• the increasing delivery times cause the SWL to be reached more often and remain in effect for longer periods, thus causing increased idle periods, during which packet generation is halted.

It is concluded that continuous transmission after the static window limit (SWL) is reached leads to a prolonged deterioration of the throughput, due to the increasing latencies displayed in the stalled time of link usage statistics. As shown in Fig. 1-b this deterioration continues for a period of time after which improvement is observed until the next period of deterioration commences. These trends alternate in a cyclic manner. It is desirable to maintain a controlled amplitude for this cyclic behavior, so that the average latency has a lower variance. One way to achieve a controlled amplitude is to halt transmission when the SWL is reached and to resume when the low window mark is reached. This would allow the network to recover from the large load and to transport the next burst of data more efficiently, with a lower latency. Such a scheme is implemented using the following rules.

Definition 3.1 *Alternating Static Window (ASW) flow control.*

1. *Transmission is allowed while the number of ORs is less than the high window mark (HWM);*
2. *once HWM is reached transmission is not allowed until enough acknowledgements are received to reduce the window size to the low window mark (LWM).*

The performance of the network with ASW ($HWM = 4$, $LWM = 0$), is presented in Fig. 4 along with that of the 4-OR SW and the SPP algorithm with 4 OR and $delta = 8\mu s$. It can be seen that all three approaches achieve approximately equivalent throughput. However, ASW displays an average RTT approximately 25% lower that the other two approaches. This demonstrates that if high throughput is desired, ASW is the best method for controlling the average latency. On the other hand, if low throughput is acceptable, SPP can be used to provide very low latencies.

4 Summary

The simplified packet-pair (SPP) flow control is evaluated. It is shown that the operation SPP can be adjusted by varying the value of the threshold parameter *delta*. The alternating static window (ASW), is defined. It is demonstrated that ASW achieves a throughput equivalent to that of SW and SPP with a large *delta*. Additionally, ASW displays a significantly lower (approx. 25%) average two-way delivery time. It is concluded that SPP is a flexible mechanism which allows sources to maintain different generations rates for different destinations. The performance of a system with SPP can be adjusted ranging from high throughput and high latency to low throughput and low latency. On the other hand, ASW is a significantly simpler mechanism that provides high throughput and reduced latency in comparison with SW and SPP. Consequently, if high throughput is desired, ASW is the best method for controlling the average latency. On the other hand, if low throughput is acceptable, SPP can be used to provide extremely low latencies.

References

[1] D. Avresky, V. Shurbanov, and R. Horst. The effect of the router arbitration policy on the scalability of ServerNet™ topologies. *J. of Microprocessors and Microsystems*, 21:545–561, 1997. Elsevier Science, The Netherlands.

[2] D. Avresky, V. Shurbanov, and R. Horst. Optimizing router arbitration in point-to-point networks. *J. of Comp. Comm.*, 22(5), April 1999. Elsevier Science, The Netherlands.

[3] D. Avresky, V. Shurbanov, R. Horst, W. Watson, L. Young, and D. Jewett. Performance modeling of ServerNet™ topologies. *The J. of Supercomputing*, 14(1), August 1999. Kluwer Acad. Pub.

[4] W. Baker, R. Horst, D. Sonnier, and W. Watson. A flexible ServerNet-based fault-tolerant architecture. In *Proc. of the 25th Int. Symp. Fault-Tolerant Computing*, pages 2–11, Pasadena, CA, June 1995.

[5] R. Horst. TNet: A reliable system area network. *IEEE Micro*, pages 37–45, Feb. 1995.

[6] R. Horst and P. Mehra. ServerNet Rate Control. Tandem Labs Technical Memorandum TL.17.2, Dec. 1998.

[7] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, Inc., 1991.

[8] S. Keshav. A control-theoretic approach to flow control. In *Proc. of SIGCOMM'91 Conf.*, volume 21 of *ACM Comp. Comm. Review*, pages 3–15, Zurich, Switzerland, September 1991.

[9] S. Keshav. *An Engineering Approach to Computer Networks.* Addison Wesley Longman, Inc., 1997.

[10] J. Kim and D. Lilja. A network status predictor to support dynamic scheduling in network-based computing systems. In *Proc. IEEE 13th Int. Par. Proc. Symp.*, pages 372–378, San Juan, PR, April 1999.

[11] S. Low and D. Lapsley. Optimization flow control - i: Basic algorithm and convergence. *IEEE Trans. on Networking*, 7(6):861–874, December 1999.

[12] C. Petitpierre and A. Zea. Implementing protocols with synchronous objects. In D. Avresky, editor, *Dependable Network Computing*, chapter 6, pages 109–140. Kluwer Acad. Pub., November 1999.