The Performance of Routing Algorithms under Bursty Traffic Loads

Jeonghee Shin and Timothy Mark Pinkston
SMART Interconnects Group
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2562, USA
Tel: (213) 740-4482, Fax: (213) 740-4418
E-mail: {jeonghee, tpink}@charity.usc.edu

Abstract
Routing algorithms are traditionally evaluated under Poisson-like traffic distributions. This type of traffic is smooth over large time intervals and has been shown not necessarily to be representative to that of real network loads in parallel processing and communication environments. Bursty traffic, on the other hand, has been shown to be more representative of the type of load generated by multiprocessor and local area network (LAN) applications, but it has been seldom used in the evaluation of network routing algorithms. This paper investigates how bursty traffic—specifically, self-similar traffic—affects the performance of well-known interconnection network routing algorithms. Various packet sizes, network resources (i.e., virtual channels) and spatial traffic patterns are used in the analysis. This allows the ability to evaluate performance under load non-uniformities in both time and space which differs from previous research that applies non-uniformity in only the space domain, such as with bit-reversal, matrix transpose, and hot-spot traffic patterns.

Keywords: network routing algorithms, self-similar traffic, interconnection network

1. Introduction
The performance of multiprocessor systems depends not only on how effectively communication and computation loads are balanced over each processor, but also on how efficiently processor nodes communicate with one another. The routing algorithm is one of the most important design factors of an interconnection network—the backbone for communication in a parallel processor environment. It significantly impacts the performance characteristics (latency and throughput) of a network under various workloads as well as resource cost. For this reason, many routing algorithms have been proposed over the last decade that incorporate several cost/performance enhancing techniques, including cut-through switching [1], virtual channel flow [2] and increased adaptivity [3]. These techniques can improve both latency and throughput in various ways. For example, adaptive routing allows the path taken by packets in the network to be decided dynamically—based on the local state of network resources—in order to more evenly distribute the traffic load over those resources, thus averting congested and/or faulty areas. As the routes taken are non-deterministic, it is important for the routing algorithm to effectively handle any anomalous behavior that may arise (such as deadlock, livelock, or starvation) by either avoiding [4] or recovering [5] from it.

Traditionally, most routing algorithms have been evaluated under traffic following a Poisson-like distribution [3, 4, 5]. This type of traffic is smooth over large time intervals and has been shown not necessarily to be representative to that of real network loads in a parallel processing or communication environment [6]. Nevertheless, according to extensive evaluations based on this traffic assumption, adaptive routing algorithms are shown to have superior performance over deterministic ones as they supposedly do a better job of balancing the traffic load over network resources and avoiding congested regions. Recovery-based true-fully-adaptive routing algorithms are shown to have higher saturation throughput capability than fully-adaptive and deterministic avoidance-based schemes given their
unrestricted ability to use network virtual channel resources [5]. The question arises, however, as to how results may be affected under a more realistic traffic model. Bursty traffic, for example, has been shown to be more representative of the type of load generated by multiprocessor and local area network (LAN) applications such as Splash-2 benchmarks and Ethernet traffic [6, 7, 8], but it has been seldom used in the evaluation of network routing algorithms. With the emergence of new bursty traffic models such as self-similar traffic, the performance of routing algorithms can be re-evaluated through simulation. It is now possible to find out whether the load balancing and head-of-line blocking benefits of adaptive routing and virtual channel flow remain, become more pronounced, or diminish in the presence of bursty traffic, and to determine how packet size and switching technique can further affect network behavior under bursty traffic. Comparisons can also be made across varying degrees of traffic burstiness and “hot spots” occurring both in time and space to challenge our current understanding of the benefits of previously proposed techniques.

This paper investigates how bursty traffic may affect the performance of well-known interconnection network routing algorithms proposed for multiprocessor and network-based computing systems. Various packet sizes, network resources (i.e., virtual channels) and spatial traffic patterns are used in the analysis. We evaluate performance under various degrees of non-uniformities in load in both the time and space domains. This differs from previous research that applies non-uniformity in only the space domain, such as with bit-reversal, matrix transpose, and hot-spot traffic patterns. Such analysis allows us to reason about the relative benefits of well-known techniques and how they can be modified to perform better under more realistic communication scenarios.

The next section describes how to synthetically generate self-similar traffic and proves that traffic generated this way indeed has self-similarity (bursty) behavior. Section 3 presents our evaluation analysis and results. Possible ways in which the performance degradation can be reduced is also presented. Related research is given in Section 4 and, finally, Section 5 presents our conclusions.

2. Self-Similar Traffic Generation

In this section, a way of generating self-similar traffic is described. This traffic is used in the performance evaluation presented in the next section. In addition, to ensure that the generated traffic has self-similarity characteristics prevalent in real traffic, a verification process is performed.

Self-similar traffic has the property of appearing and behaving similarly across different time scales [9]. In other words, the time sequence exhibits a similar pattern regardless of the degree of resolution. This means that self-similar traffic is bursty in both small and large time scales (i.e., has long-range dependence). This is opposed to Poisson-like traffic which is bursty only in small time scales but is smooth in large time scales (i.e., has short-range dependence).

One of the most popular approaches for synthetically generating self-similar traffic is by superimposing many Pareto-like ON/OFF sources [11]. In the ON/OFF source model or packet train model suggested in [12], packets arrive at regular intervals during ON-periods, i.e., the train length, while OFF-periods are periods without packet arrivals, i.e., the inter-train distances. Each source alternates between an ON and an OFF period. These ON- and OFF-periods on each source have high variability, which follows a Pareto distribution with parameter \( \alpha \) (i.e., the probability distribution function \( F(x) = 1 - x^{-\alpha} \) where \( 1 < \alpha < 2 \) and \( x \) is a non-negative value). The superposition of many ON/OFF sources produces aggregate network traffic, which is self-similar with Hurst parameter \( H \). The Hurst parameter represents the degree of self-similarity of the aggregate traffic stream, where \( H = (3 - \alpha)/2 \); thus, \( 1 < \alpha < 2 \) implies \( 0.5 < H < 1 \).

The benefit of this approach is that a traditional Poisson-like traffic generator can be used for generating self-similar traffic simply by adding an ON/OFF controller to it. During ON-periods, bursty traffic consisting of Poisson generated packets during both the current ON-period and in the previous OFF-period is injected into the network. During OFF-periods, no newly generated packets are injected into the network. As is stated, the length of ON- and OFF-periods is Pareto distributed: \( F(x) = 1 - x^{-\alpha} \), where \( 1 < \alpha < 2 \). Let \( R \) be a random number with a uniform distribution between 0 and 1. Then, \( x = (1 - R)^{1/\alpha} \).
According to a sampling of 1994 Ethernet traffic [11], $\alpha$ is around 2.0 for ON-periods ($t_{ON}$) and between 1.0 and 1.5 for OFF-periods ($t_{OFF}$). Therefore, $t_{ON} = (1 - R)^{-1/\alpha_{ON}}$, where $\alpha_{ON} \approx 2.0$, and $t_{OFF} = (1 - R)^{-1/\alpha_{OFF}}$, where $1.0 < \alpha_{OFF} < 1.5$.

Such generated traffic with parameter $\alpha_{ON} = 1.9$ and $\alpha_{OFF} = 1.25$, which is used for the experiments with 16-flit packets on 16×16 two-dimensional torus discussed in the next section, is shown in Figure 1. The number of packets generated during 500, 50 and 5 cycle time intervals are shown versus the number of time intervals over which statistics were gathered and the same segments of traffic with different time intervals are indicated by the same gray levels. As is seen in Figure 1, self-similar traffic maintains burstiness, while Poisson-like traffic becomes smoother as the time scale increases.

Another way to prove whether or not traffic generated synthetically exhibits self-similarity is through the use of variance-time plots [9]. The variance of the $m$-aggregated time series $X^{(m)}$ of self-similar processes for large $m$ is described by the following:

$Var(X^{(m)}) = Var(X)/m^H$, where $H = 1 - (\beta/2)$.

This can be rewritten as

$log[Var(X^{(m)})] = log[Var(X)] - \beta log(m)$.

Here, a slope of $-\beta$ in $Var(X^{(m)})$ versus $m$ on a variance-time plot implies the degree of self-similarity. As $\beta$ approaches 0 (alternatively, 1), traffic has a higher (alternatively, lower) degree of self-similarity. A variance-time plot of the generated traffic is shown in Figure 2, where self-similar traffic with 16-, 32- and 128-flit packets is compared to Poisson-like traffic with 16-flit packets. As given by the larger slope, self-similar traffic indeed has more self-similarity than Poisson-like traffic. Moreover, the Hurst parameter $H$ of self-similar traffic with 16-flit packets is 0.92, which is close to the Hurst parameter of multiprocessor systems [8] ($H = 0.93$ on average) as well as Ethernet traffic [11] ($H = 0.9$). Thus, the synthetic traffic used for performance evaluation in this paper highly reflects real traffic occurring in parallel processing and communication environments. The relationship between the Hurst parameter of self-similar traffic and the packet size will be discussed in Section 3.2.

Figure 2. Degree of self-similarity of generated traffic.
3. Evaluation Methodology and Results

A performance evaluation that shows the effect of self-similar traffic is carried out with various well-known interconnection network routing algorithms, network design parameters and spatial traffic patterns. Deterministic e-cube routing and two adaptive routing algorithms—avoidance (Duato [4]) and recovery (Disha [5])—are evaluated with different packet sizes (16, 32 and 128 flits), and four or eight virtual channels. In terms of the spatial traffic pattern, uniform and non-uniform traffic (bit-reversal and hot spot) are used. Each node under uniform traffic sends packets to all other nodes with equal probability. On the other hand, a node sends messages to the node with its reversal coordinates under bit-reversal traffic. In case of hot spot traffic, up to 5% of the network traffic is sent to a single hot spot in the network; the other 95% is uniform traffic. All simulations are run on a 16×16 two-dimensional wormhole torus with full-duplex links.

As is described in Section 2, Poisson-like traffic created by a traditional traffic generator is modulated by an ON/OFF controller to produce self-similar traffic generation. During OFF-periods, generated Poisson traffic is queued in the self-similar generator, while during ON-periods, the queued traffic and newly generated Poisson traffic are presented to the network. In generating the self-similar traffic, the controller has a length of ON- and OFF-periods which is Pareto distributed with the parameter $\alpha = 1.9$ and 1.25, respectively. In generating the Poisson traffic, the controller maintains a 100% duty factor (i.e., always ON state with no OFF state).

In each simulation, the first 10,000 cycles are excluded from the performance measurements (throughput and latency) so that the network reaches steady state before collecting data. Latency is plotted versus throughput for increasing applied load rate in Burton Normal Form [10]. In all cases, applied load rate is defined as a fraction of the full bisection bandwidth of a network assuming uniform traffic distributed evenly over both space and time. Throughput is measured as the number of arrived flits at each node per cycle and latency is measured as the average number of cycles needed to deliver each packet.

3.1 Comparison of the Benefits of Increased Routing Adaptivity

A performance comparison of the routing algorithms under different spatial traffic patterns with 16-flit packets and 4 virtual channels is provided in Figure 3(a) - (c), where solid and dotted lines indicate self-similar (SS) and Poisson-like (PO) traffic, respectively. Figure 3(a) shows that under spatially uniform and self-similar traffic, the deterministic algorithm (E-cube) has slightly better performance than the adaptive ones (Disha and Duato). This not only indicates that the performance degradation of the adaptive routing algorithms is worse than that of the deterministic routing algorithm—as is seen in Figure 3(a), adaptive routing has over 40% performance degradation while deterministic routing has only 5%—but it also indicates that bursty traffic makes the adaptive routing networks saturate at a slightly lower load rate, compared with the deterministic routing networks. A possible reason for this is the following. In the adaptive routing algorithms, the number of channels available for injecting new packets into the network may be larger than that in the deterministic routing algorithms since routing adaptivity provides more choices of channels for injecting the packets into the network. These channels occupied by the injected packets may then be released earlier by providing multiple routing paths—the packets can be routed through different paths instead of waiting for a particular path to be released. Therefore, a larger portion of bursty traffic could be accepted into the adaptive routing networks, thus causing the network to be saturated at lower load rate.

The performance of self-similar traffic with different spatial traffic patterns is shown in Figure 3(d), where U, B-R and H-S depict uniform, bit-reversal and hot-spot traffic, respectively. One interesting result is that throughput in the adaptive routing algorithms is about the same under uniform or bit-reversal traffic with self-similarity. The reason for this is that the adaptive routing network under non-uniform traffic in time reaches an early saturation point, thus no more performance degradation is caused by adding non-uniformity in space. On the other hand, in the deterministic routing algorithm, a 30% decrease in throughput between uniform and bit-reversal traffic with self-similarity is observed.
Consequently, adaptive routing algorithms have better performance than deterministic ones under both spatial and temporal non-uniform traffic.

From these results, it can be said that routing adaptivity is an important factor in relieving spatial bursts, but it is not sufficient for relieving temporal bursts. This fact is clearly revealed by Figure 3(e). In adaptive routing algorithms (Duato and Disha), self-similar traffic with uniform traffic pattern (U, SS) which has only temporal non-uniformity produces more performance degradation than bit-reversal traffic with a Poisson distribution (B-R, PO) which has only spatial non-uniformity. On the other hand, in deterministic routing algorithms (Ecube), spatial non-uniformity causes more performance degradation than temporal non-uniformity.

### 3.2 Comparison of the Benefits of Increased Packet Size

Performance comparison of different packet sizes (with 4 virtual channels and under uniform traffic) is presented in Figure 4. Compared to 16-flit packets in Figure 3(a), larger packets mitigate performance degradation caused by bursty traffic since larger packets are injected less often than smaller ones with the same load-rate, thus making traffic less bursty. For example, assume that 4096 flits should be injected to the network. In case of 16-flit packets, at most 256 nodes inject a packet at the same time, while at most 32 nodes inject 128-flit packets simultaneously. In other words, 16-flit packets make more burst than 128-flit packets. This fact is also well represented by a variance-time plot which indicates the degree of self-similarity, i.e., burst in Figure 2. The Hurst parameter $H$ of 32-flit packets, 0.91, is slightly less than that of 16-flit packets, 0.92, while in case of 128-flit packets the Hurst parameter $H$, 0.67, is close to 0.5.

A couple of factors which affect performance can be considered as well. Larger packets with the same load-rate occupy less virtual channels, thus providing more freedom of virtual channels. In addition, as packet size increases, transmission time per flit decreases due to less set-up time [10]. Therefore, messages consisting of larger packets could be delivered faster than the same size of messages consisting of smaller packets. This fact might cause less traffic to be present in the network, thus saturating the network more slowly.
### 3.3 Comparison of the Benefits of Increased Virtual Channels

To observe the effect of the number of virtual channels, twice more virtual channels than in Figure 3 and Figure 4 are provided in Figure 5. Compared with 4 virtual channels, Figure 5 shows that the increase in the number of virtual channels improves throughput of all routing algorithms regardless of traffic uniformities in time. In particular, it alleviates performance degradation of routing algorithms under bursty traffic since more virtual channels are helpful to distribute traffic over the network and relieve burstiness. In case of 128-flit packets, performance degradation is almost resolved by using twice more virtual channels.

### 3.4 Discussion

As is shown in performance results, burstiness in traffic causes severe performance degradation. Therefore, the way to improve the performance under this traffic is to make traffic injected into the network smooth even though bursty traffic is generated. One of the best ways for this is to provide a congestion control mechanism. The congestion control mechanism makes traffic injected into network not exceed a given maximum level, thus helping the network to avoid being saturated. So far, several congestion control mechanisms [13, 14, 15] have been proposed, and the self-tuning mechanism proposed in [11] in particular works well under loads created by alternating low loads and high loads, which is not exactly self-similar. The alternative for making traffic smooth could be the use of more virtual channels or lager packets.

### 4. Related Work

So far, many network applications and models for LAN or WAN have been reevaluated under self-similar traffic [7]. Recently, this traffic has been explored in multiprocessor systems as well [8, 16, 17]. In particular, the observation of self-similarity in interconnection network traffic generated among multiprocessors [8] motivated the performance reevaluation of interconnection
network properties proposed for multiprocessor systems.

The performance of SeverNet SAN—the wormhole-routed and point-to-point network for server systems—was reevaluated under self-similar traffic in order to improve the routers and end devices, and to modify the optimization method which was developed on the basis of Poisson-like traffic [16]. That work shows the design consideration and evaluation results not in the generic system, but in the specific system. However, our paper provides the results which can be applicable for the generic systems.

In addition, an analytical performance model for self-similar traffic [17] has been proposed. It supports pipelined circuit switching (PCS) routing algorithms with the uniform traffic patterns in k-ary n-cubes. Moreover, analytical models for various environments such as the wormhole-routed torus, adaptive wormhole routing or circuit-switched network are currently being researched. That performance model does not consider burstiness of traffic in both space and time, and measures performance only in terms of latency. However, throughput is one of the most important quantities to present the performance and should not be ignored. In our paper, the effect of burstiness in both time and space is provided, and the performance is measured by both latency and throughput.

5. Conclusion

This paper has investigated the effect of self-similar traffic on the performance of previously proposed routing algorithms with various spatial traffic patterns, packet sizes and number of virtual channels. Consequently, adaptive routing algorithms under non-uniform traffic in both time and space have better performance than deterministic ones. However, compared with deterministic routing algorithms, adaptive routing algorithms have more performance degradation caused by temporally bursty traffic. This implies that routing adaptivity is not enough to relieve temporal non-uniformity. Thus, the additional congestion control mechanisms to relieve temporal burstiness would be useful. In addition, larger packet sizes and more virtual channels could help smooth out bursty traffic.

References