# Realistic Workload Characterization and Analysis for Networks-on-Chip Design

Paul V. Gratz
Department of Electrical and Computer Engineering
Texas A&M University
Email: pgratz@tamu.edu

Stephen W. Keckler
Department of Computer Sciences
University of Texas at Austin
Email: skeckler@cs.utexas.edu

## Abstract

*As silicon device scaling trends have simultaneously increased transistor density while reducing component costs, architectures incorporating multiple communicating components are becoming more common. In these systems, networks-on-chip (NOCs) connect the components for communication and NOC design is critical to the performance and efficiency of the system. Typically, in NOC design traditional synthetic workloads are used to characterize the performance of the system. Previous work shows, however, that these traditional synthetic workloads do not accurately represent the characteristics of realistic network traffic. In this paper we propose the analysis of realistic traffic via a set of new workload characteristics. We show that these characteristics more accurately correlate with network congestion in realistic workloads than traditional metrics.*

## 1 Introduction

Over the past 40 years silicon device scaling, as characterized by Moore's law [1], has simultaneously increased transistor density while reducing component costs. Moore's law has recently led to architectures that incorporate multiple communicating components, such as systems-on-chip (SOC) and chip multiprocessors (CMP) designs. Examples include the CMP Intel Core 2 Quad [2], and the SOC IBM Cell processor [3]. In the past, ad-hoc interconnect and bus protocols were used to convey information within a chip. The scaling of these interconnects with frequency and greater numbers of components motivates a shift to on-chip interconnection networks to take an analogous role to off-chip interconnection networks seen in large scale multiprocessor systems.

Networks-on-chip (NOCs) are used in several contexts. Some, such as the Tilera Tile64 processor's "Static Network" (STN) [4], serve as a replacement for the bypass bus found traditional superscalar processors, carrying operands between producing and consuming instructions. Others, such as the TRIPS OCN [5], interconnect processors with their second-level cache and primarily carry cache blocks between the first- and second-level caches. Chip multiprocessor (CMP) NOCs are another important class of on-chip interconnection network. CMP NOCs primarily carry cache coherency traffic, as well as first- and second-level cache block transfers. As each of these different NOCs must be designed to accommodate the traffic that traverses them, an understanding of the typical workload characteristics of each network will have is of paramount importance in producing a well balanced network design.

Typical network design is informed by a combination of broad aggregate characteristics of the expected workloads, such as the average injection rate of a given benchmark, and network performance under a small set of synthetic workloads in simulation. Prior work shows these aggregate characteristics, combined with typical synthetic workloads, inaccurately model the performance of the network under realistic workloads [5, 6].

This paper makes the following contributions:

1. We introduce a new set of metrics for the evaluation of realistic NOC workloads. Metrics which characterize the temporal and spatial imbalances in network traffic distribution.
2. We show traditional synthetic workloads driven by broad aggregate workload characteristics do not accurately model the diversity of realistic workloads.
3. We evaluate the correlation between the new workload metrics and NOC packet latency, validating their effectiveness in predicting workload performance on a given network.

This paper is organized as follows, Section 2 discusses several commonly used realistic and synthetic workloads. This is followed by a discussion of the workloads examined and the methodology of their capture in Section 3. Section 4 introduces several new workload characteristics and uses them to contrast the various workloads. Section 5 explores the correlation between individual workload characteristics and packet latency within each suite of benchmarks. Section 6 describes the prior work in NOC workload characterization, and contrasts it with the work presented here and Section 7 concludes.

## 2 Background

### 2.1 Synthetic Workloads

Balanced network-on-chip (NOC) design requires the network accommodate the traffic traversing it without over-provisioning the network and wasting resources such as power or area. Traditionally the ability to accommodate a given traffic load has been evaluated under synthetic traffic workloads. In synthetic traffic, the source and destination node patterns are typically driven by a stochastic uniform random injection process. The spatial characteristics of the source and destination node patterns and the temporal characteristics of the uniform random injection process are intended to model the characteristics of realistic workloads. Synthetic traffic patterns used in NOC research and design include the following [7]:

**Bit-complement:** In *bit-complement* traffic, the destination coordinates are the bit-wise inversion of the source coordinates. This load stresses the horizontal and vertical network bisections. Under this load, DOR statically spreads traffic across all of the bisection links, providing a perfectly balanced network load.

**Uniform Random:** In *uniform random* traffic, source and destination nodes are chosen via a uniform random process. The load is balanced because the source and destination coordinates are uniformly distributed, although the random process that generates the coordinates may cause transient hotspots.

**Transpose:** In *transpose* traffic, the destination coordinates are the transpose of the source coordinates. Under this load the network's diagonal bisection is a bottleneck as all packets must cross it. *Transpose* traffic, in combination with the dimension order routing (DOR) algorithm commonly found in NOCs, produces a highly imbalanced network load. In *transpose* traffic, the links counter-clockwise about the center of the mesh are utilized while the clockwise links are unused.

While these three synthetic workloads may provide some insight into a network's performance bottlenecks, prior work has shown them to be ineffective in predicting the performance of realistic workloads [5, 6]. As we will show, this is because the static, evenly distributed source and destination patterns and the aggregated mean injection rate do not characterize the variance in injection rate from node to node in locale and over the course of the benchmark in time seen in realistic workloads.

Other synthetic workloads exist, including *tornado*, *bit-reversal*, and *shuffle*. These other synthetic workloads either do not apply to the mesh topologies typically found in NOCs (*tornado*), or produce traffic with similarities to the three listed above (*bit-reversal* and *shuffle*). The remainder of this paper will utilize *bit-complement*, *uniform random*, and *transpose* as representative of typical synthetic workloads.

### 2.2 Realistic Workloads

One way to ensure the network design matches its intended use is to perform design exploration experiments using realistic workload traces. Three realistic, application-driven workload suites commonly used in NOC design research include:

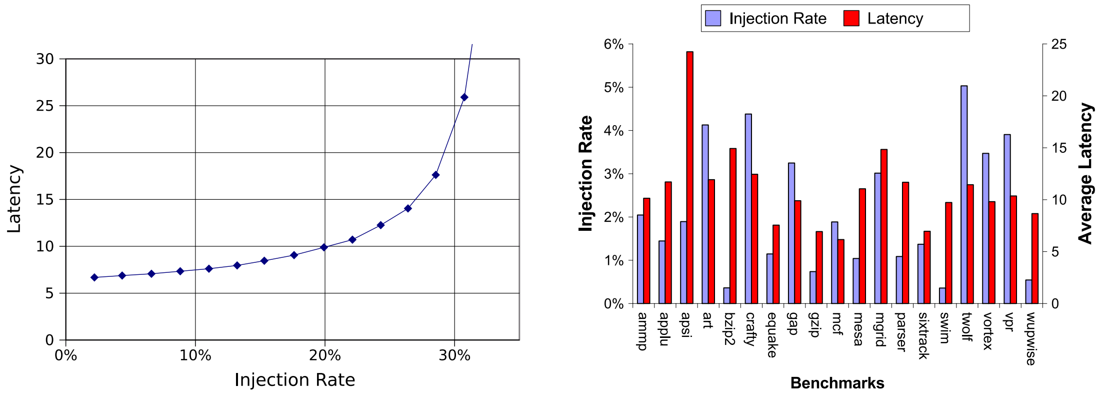
**SPEC CPU2000:** The SPEC CPU2000 benchmark suite is a standardized, uniprocessor benchmark suite designed to provide a measure of CPU-intensive performance [8]. Researchers use uniprocessor benchmark suites, like the SPEC CPU2000 suite, to generate memory system request traffic representing the traffic seen when independent programs are executed on shared memory multiprocessors [5, 9].

**SPLASH-2:** The SPLASH-2 benchmarks represent a traditional HPC-centric, shared-memory multiprocessor scientific workload [10], and is frequently used in NOC research [11, 12, 13, 14].

**PARSEC:** The PARSEC benchmarks represent an emerging, shared-memory multiprocessor workload focused on chip-multiprocessing environments [15]. The PARSEC benchmarks have only recently been introduced, as a result a limited amount of published research has incorporated their analysis [16, 17].

It should be noted the term “realistic workload” has different meanings for different NOC designs. Each of the above benchmark suites is intended to model a completely different portion of the overall application space for current and future NOCs. It would be effectively impossible to cover the entire space in any meaningful way. As the benchmarks suites we have chosen are quite divergent from one-another, we feel that it reinforces the commonalities we find in traffic characteristics between.

Ideally, application-driven workload suites would provide the best means of evaluating a network's performance. In practice, however, realistic workload traces are not easily generalizable to a wide range of network sizes and often require a great deal of simulation time, making them too cumbersome to use during



(a) Uniform random synthetic traffic on the TRIPS OCN network.

(b) SPEC CPU2000 traffic on the TRIPS OCN network.

**Figure 1. Injection rate and latency for synthetic and realistic traffic on the TRIPS OCN network.**

the design exploration phase of NOC design.

## 3 Workload Simulation

The remainder of this paper will focus on the synthetic workloads described in Section 2.1 and on traces from the realistic workloads described in Section 2.2. The synthetic workloads contain statistically generated traffic, therefore the simulation of a full system is not required. Realistic application-driven workload traces were simulated and captured as follows:

**SPEC CPU2000 on the TRIPS OCN:** The TRIPS OCN network interconnects the TRIPS processor core with the individual banks that form the second level cache [5]. The TRIPS OCN is an Y-X DOR routed, 4x10, 2D mesh network. A TRIPS full processor simulator executing the subset of SPEC CPU2000 benchmarks currently supported by the TRIPS simulation infrastructure captured all OCN traces. In each case the MinneSPEC [18] reduced input data set was used in conjunction with SimPoint checkpoints [19] to approximate the application behavior.

**SPLASH-2:** The Bochs full-system functional simulator was used to obtain traces for a forty-nine node, shared memory multiprocessor system arranged in a 7x7 2-D mesh topology simulating a subset of the SPLASH-2 benchmarks [11]. In the simulated system each processor node contains a private L2 cache bank of 2MB. Cache banks are kept coherent via a MSI directory based protocol. The traffic in the start-up phases of these benchmarks is much greater than traffic during the remainder of the benchmark causing the benchmark performance to be dominated by the start-up phase. For this reason the results in the remainder of this paper will focus on the first two million cycles of each SPLASH-2 benchmark.

**PARSEC:** The M5 full-system functional simulator was used to obtain traces for a sixty-four node, shared memory multiprocessor system arranged as a 8x8 2-D mesh topology simulating a subset of the PARSEC benchmarks [16]. The simulated system implements a fully shared, S-NUCA style L2 cache of 16MB. Only benchmarks currently supported by our simulation infrastructure were included. The “simmedium” input data set was used for all benchmarks with the exception of x264 where “simlarge” was used.

In each case, the traces were captured from different processor types and in different simulation environments. Each benchmark suite trace was taken from previously published works (as cited above), having thereby been vetted by the submission process associated with each publication. We chose to use traces from published works rather than recreating our own, even if that meant that the simulation environment was not consistent, because we felt it was important that we showed that our analysis would hold regardless of simulation setup.

## 4 Workload Characteristics

Networks-on-chip are implemented in many different contexts and have diverse workloads. Understanding a network’s expected workload characteristics is important in producing a well balanced network design. Typically network design is informed by broad aggregate characteristics of the expected workloads, such as the average injection rate of a given benchmark. In this section we will examine this traditional workload characteristic as well as several new characteristics we introduce. These improved workload characteristics have applications both in directly identifying imbalances in the network that may lead to poor performance as well as informing the generation of generalizable synthetic workloads that more accurately represent realistic application-driven workloads.

Figure 1(a) shows the average packet latency of a *uniform random* synthetic workload on the TRIPS OCN network for various injection rates. In the figure, packet latencies are low for low injection rates, approximately 7 cycles on average. As the load is increased, packet latencies gradually increase until a tipping point occurs at injection rates of approximately 30%, where latencies increase exponentially with injection rate. By convention the injection rate at which the packet latency is three times the low-load latency is labeled the saturation point for the network under that load.

Figure 1(b) shows the injection rate and latency for the TRIPS OCN under the SPEC CPU2000 workload. These realistic workloads show great variance between the injection rate of each benchmark and its corresponding average packet latency. For example, *apsi* exhibits a low injection rate, less than 2%, while it has the highest packet latency at 25 cycles. Conversely, *twolf* has the highest injection rate, at 5% while it has an average packet latency of only 11 cycles. In each case the realistic workload injection rates are well below the saturation point for the synthetic workload, yet all of the realistic benchmarks exhibit latencies that are above the low-load latency for the synthetic workloads. These results imply a disconnect between the average aggregate injection rate of a realistic workload and its average packet latency.

In the remainder of this section we examine the workloads from TRIPS and multiprocessor NOCs with respect to their variance in injection rate with respect to time and source-destination location. We will show that these benchmark characteristics are key in determining the network's behavior over the course of those benchmarks. In each case we contrast the realistic workload traffic against the typical synthetic workloads introduced in Section 2.1.

## 4.1 Injection Rate Variance in Time

In most synthetic workloads, including *bit-complement*, *uniform random* and *transpose*, injection time of individual packets is determined by a uniform or Poisson random process. Prior work, however, shows that many realistic workloads exhibit periodic behavior characterized by a high variance in their injection rates over time [5, 6]. The periodic variance in injection rate creates bursts and troughs of injections over a wide range of timescales. Rather than largely homogeneous execution, as modeled by synthetic benchmarks, programs typically have phases or periods in which characteristics such as memory references, and instructions per cycle are radically different from other phases [19, 20].

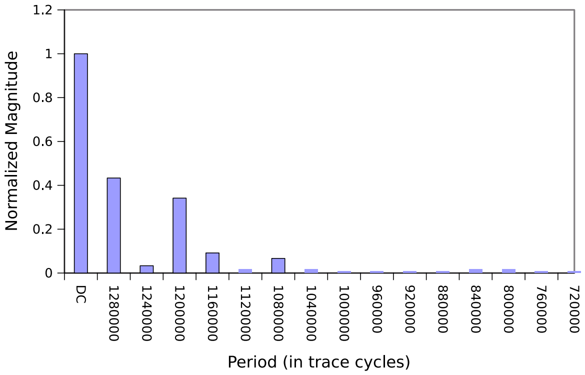
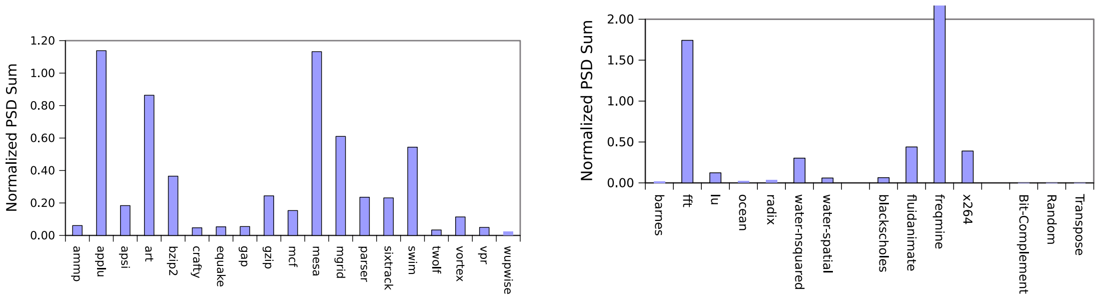


**Figure 2. Power Spectral Density for the injection rate of the OCN *mesa* benchmark.**

The exponential relationship between packet latency and injection rate during saturation, combined with phase based traffic's periodic high injection rates can produce packet latencies that dominate simulation performance. The wide range of offered rates caused by program phases include very high offered rates disproportionately affecting the average packet latency by placing the network in saturation for certain phases. Understanding the magnitude and frequency of program phase behavior within a given workload informs the need to over-provision the network at design time.

A measure the frequency composition of a given signal is its Power Spectral Density (PSD). The PSD of a time series is calculated as two times the square of its Fourier coefficients and provides an estimate of the magnitude of the frequency components of a given signal [21]. Figure 2 shows the PSD of the injection rate of the OCN *mesa* benchmark. The PSD values shown are normalized against the magnitude of the DC component (the first element in the figure). In the figure the X axis denotes the frequency component's period, in OCN network cycles. The height of the bar denotes the magnitude of periodic phase behavior at that frequency. In this diagram the DC component of the injection rate is equivalent to the aggregate average injection rate over the course of the entire benchmark.

The diagram shows significant magnitudes at several low frequencies, in particular those with periods of 1.28 million cycles and 1.2 million cycles. These peaks indicate the presence of program phases which repeat with a period of 1.28 to 1.2 million cycles, that strongly effect the injection rate. We found that the dominant injection rate variances occurred with periods larger than 40000 cycles, therefore the injection



(a) OCN SPEC CPU2000 normalized PSD.

(b) Multiprocessor SPLASH-2, PARSEC and synthetic trace normalized PSD.

**Figure 3. Normalized power spectral density of injection rate for TRIPS and multiprocessor traces.**

rate was sampled every 20000 cycles to highlight the low frequency program phase components.

Figure 3 shows the sum of the PSD's non-DC frequency components of the injection rate of SPEC CPU2000 benchmarks in the TRIPS OCN, SPLASH-2 benchmarks and PARSEC benchmarks as well as the three synthetic benchmarks. In these diagrams the PSDs were summed and normalized against the DC component to provide a relative measure of the presence or absence of non-DC, periodic components within each benchmark's injection rate over time. The intuition behind this metric is that the individual periodic components of the injection rate are less important to network performance than the sum of the magnitudes of the periodic components because injection rates with greater numbers of measurable frequency components are likely to express more highly variable injection rates having a much larger impact upon performance than those with fewer measurable frequency components.

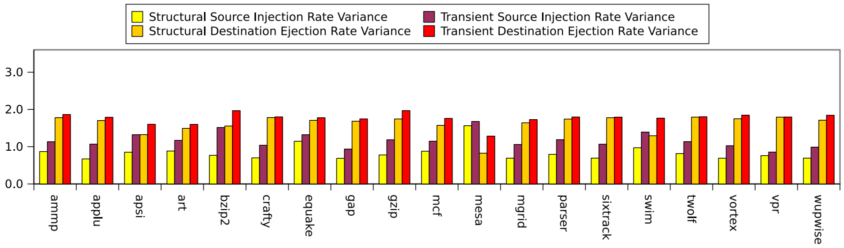
Figure 3(a) shows the normalized sum of the PSD for the OCN traces. The figure shows a great diversity in phase behavior from benchmark to benchmark. In the figure several benchmarks standout with significant periodic components in their injection rate over time due to phase based behavior. In particular *applu* and *mesa* show spectral components larger than the DC component, indicating the presence of very large swings in injection rate over time that are not captured in the average aggregate injection rate. Many other benchmarks show small normalized PSD sums, indicating little variation in their injection rate over time.

Similarly Figure 3(b) shows the the normalized sum of the PSD for the multiprocessor and synthetic traces. Here two benchmarks in particular show very high normalized PSD sums, SPLASH-2's *fft* and PARSEC's *freqmine*. The other multiprocessor benchmarks show much less phase-based, periodic behavior. By contrast, all of the synthetic benchmarks show normalized PSD sums near zero, as would be expected from a uniform random injection rate. In the design of a network to carry one of the benchmark driven workload suites examined, simply ensuring that the network was capable of carrying a load equal to the average injection rate over the course of the trace would not be sufficient because cycle program phase behavior would create times where the injection rate would be much higher than the average. It should however be possible to design a synthetic workload with a periodic injection rate that would be a better model for the traffic the network would carry.

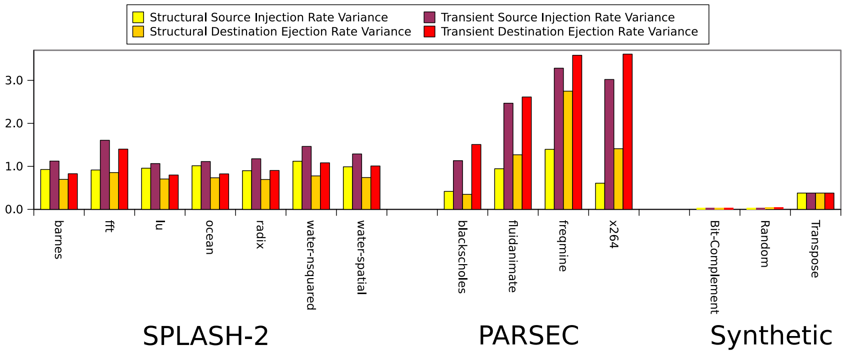
## 4.2 Injection Rate Variance in Locale

Network hot-spots, locations within the network that have over-committed resources, can have a disproportionate effect on network performance similar to injection rate bursts. Two places hot-spots often form are around over-committed destination or source nodes. In these cases, either the injection or ejection rate for a subset of nodes is significantly higher than the average injection rate across all nodes. As with injection rate variability, detection of source or destination node hot-spots within a given workload gives important information about the expected performance of that workload. One way to detect source or destination hot-spots is to examine the variance in source injection or destination ejection rates. While the mean source injection and destination ejection rates will be a static fraction of the overall injection rate, the variance or standard deviation of these rates indicates the degree of skew in rate from node to node and the likelihood of hot-spots.

We will examine two forms of source or destination node injection rate variance, *transient* and *structural*. *Transient* node rate variances persist for a limited time and cause hot-spots which vary over the course of the benchmark. *Transient* node injection rate variances are caused by program phase behavior in the benchmarks, creating temporary resource hot-spots. *Struc-*



(a) *Structural* and *transient* standard deviation of source injection and destination ejection rates normalized against the mean rates for OCN traces.



(b) *Structural* and *transient* standard deviation of source injection and destination ejection rates normalized against the mean rates for multiprocessor and synthetic traces.

**Figure 4.** ***Structural* and *transient* source and destination rate variance for TRIPS, multiprocessor and synthetic traces, normalized against the mean rate for each benchmark.**

*tural* node rate variances persist over the entire course of the benchmark causing permanent hot-spots. *Structural* node rate variances are caused by functional differences from node to node. Understanding the degree of injection rate variance between source nodes or destination nodes within a workload can inform need to over-provision the network at design time, potentially toward heterogeneous topologies (i.e. topologies in which some network links have greater bandwidth than others), if common workloads are structurally skewed.

The traditional synthetic workloads vary by the function used to determine the source and destination node for each packet, however in many synthetic workloads each source node injects packets destined for one and only one destination. In these synthetic workloads the variance in per-node injection and ejection rate is very low, meaning that each node injects and ejects packets at approximately the same rate. Synthetic workloads driven by uniform random injection processes also show little difference in the node rate variance over time because uniform processes tend towards the mean for large enough sample sizes. Realistic workloads that do not match these source or destination rate characteristics may cause network hot-spots at the nodes which have higher injection or ejection rates.

Figure 4 shows the *structural* and *transient* standard deviation of source injection and destination ejection rates for the TRIPS OCN, multiprocessor and synthetic traces normalized against the mean rate for each benchmark. In this figure the *structural* rate variance is calculated by measuring the standard deviation in the aggregate average injection rate for each node over the course of the entire benchmark. By contrast, the *transient* rate variances are calculated by sampling the standard deviation in injection rate for each node once every 1000 cycles then performing a flit-count weighted average of these samples across the entire benchmark.

The source injection rate is the rate at which packets are injected from a particular source. The destination ejection rate is the rate at which packets are injected destined for a particular destination. While the mean source injection and destination ejection rates will reflect a simple fraction of the overall injection rate, the standard deviation of these rates when normalized against the mean rate indicates the degree of injection rate skew from node to node, orthogonal to the injection rate itself. For example if one particular source node injects most of the packets within a trace, its injection rate will be much higher than the rest of the nodes in the network for that trace and the standard deviation of the node injection rate for that trace will be greater than a trace where all sources inject packets equally. Generally a larger spread in rates indicates a greater likelihood of injection or ejection hot-spots.

Across the all traces in Figure 4, the *transient* rate variances are greater than the *structural* rate variances. This is to be expected, as the method used to capture the *transient* rate variances encapsulates both variances due to structural differences as well as variances that are transient in nature, while the structural variance shown will have the transient variances normalized out due to aggregation of the injection rates over the course of the benchmark.

Figure 4(a) shows the OCN network has a relatively large *structural* variance for injection and ejection rates. In these traces, only one processor and its associated cache banks are legal nodes for injection or ejection, structurally skewing the source and destination rate standard deviations in the OCN traces. This extreme structural skew in network source and destination node rates causes severe hot-spots and congestion for benchmarks with an injection rate of any significance. In the TRIPS OCN, the primary injectors and ejectors are known a priori, and are unchanging across all workloads. These results argue for a heterogeneous network design, possibly taking advantage of the available wire bandwidth to over-provision the network links near the primary sources or destinations. For most benchmarks, the *transient* rate variances are only slightly higher than the *structural* rate variances indicating that, in these workloads, differences due to program phases do not have a large effect on the distribution of injection or ejection rates.

Interestingly, the SPLASH-2 and PARSEC multiprocessor benchmarks in Figure 4(b) show significant *structural* rate variances despite the homogeneous design of the multiprocessors execute on. These differing rates indicate multi-threaded load imbalances between processors which persist over time or hot, shared-memory locations, leading to source and destination hot-spots in the network. Unlike the OCN, it is unlikely that the source and destination rates occur in the same nodes from benchmark to benchmark, so a heterogeneous network design is not called for. The PARSEC benchmarks also show a great deal of *transient* rate variance, much larger than *transient* rate variance seen in the SPLASH-2 benchmarks. These results indicate the PARSEC benchmarks exhibit more skewed injection and ejection due to program phases than the SPLASH-2 benchmarks and nodes have heterogeneous network usage during those phases. In both cases, some amount of over-provisioning of all network nodes would be a fruitful approach to hot-spots.

The synthetic traces, also in Figure 4(b) show little deviation in rate from node to node, as expected because of the uniform random injection process determining source and destination for each packet. *Transpose* traffic shows a slightly higher deviation in source and destination rates primarily because nodes along the diagonal bisection of the network, where the X coordinate is equal to the Y coordinate, do not inject traffic. Even in the case of *transpose*, the source and destination node rate variances are not as high as in the other workload suites, highlighting the difference between realistic and synthetic loads.

Analysis shows that even though average offered rate is low, applications can easily create network injection hot-spots that may approach local saturation, producing higher than expected transmission latencies. The operating system schedules threads to more evenly distribute the network traffic; however such optimizations must be balanced against the effect of increasing the average source to destination hop count.

This section examined the TRIPS NOC and multiprocessor network workload traces with respect to their variance characteristics along several axes. We have shown that these benchmark characteristics can provide useful indications of types of workloads those networks can expect to encounter. We also found that the synthetic workloads traditionally used in interconnection network evaluation do not match the characteristics of realistic benchmark driven workloads.

Based upon these observations one potential use for workload characteristic information is to inform the design of the network under examination. For example, workload characteristics could be used to determine the networks that may need to be over-provisioned to avoid potential network congestion, and what form of over-provisioning would be most beneficial. With this in mind, the next section examines how sensitive each network's performance is to workload characteristics.

## 5 Network Performance Sensitivity to Workload Characteristics

In Section 4 we established a new set of workload characteristics based upon spatial and temporal variances injection rate. While we have shown how these characteristics differentiate realistic benchmark driven workloads from traditional synthetic workloads, the relevance of the characteristics to network design is dependent upon their correlation with network perfor-

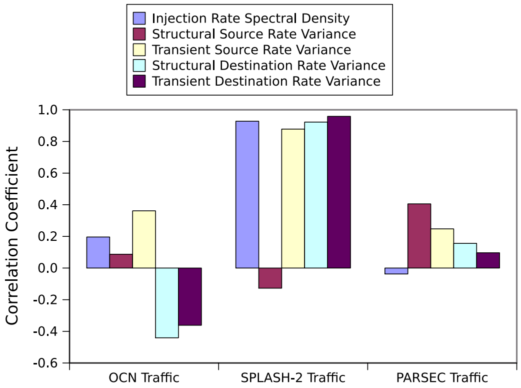


**Figure 5. Correlation between workload characteristics and average packet congestion.**

mance. This section explores the relationship between the workload characteristics of each network's benchmark suite and the performance of the network. In particular the section examines the correlation between workload characteristic and network performance in terms of the packet congestion of that network. Packet congestion in this context is measured as the packet's latency beyond the static, zero-load latency caused by hop distance and serialization delay.

In this section, packet latency is measured for each of the benchmarks via a trace driven network simulation configured to match the parameters of the original system from which the traces were captured. In each case the simulated network used YX-DOR routing. The SPLASH-2 and PARSEC benchmarks used virtual channel flow control with four virtual channels each eight flits deep, emulating a heavy-weight CMP NOC. The TRIPS OCN used worm-hole flow control with two-flit-deep input FIFOs, to mimic the design of the actual TRIPS OCN.

The Pearson product-moment correlation coefficient is commonly used as a measure of the strength and direction of a linear relationship between two random variables. Correlation coefficients range from -1 to 1, where a correlation value of 1 indicates a positive-sloped, perfectly linear relationship between two variables, while a correlation coefficient of -1 indicates a negative-sloped, linear relationship. Generally, correlation coefficients greater than 0 indicate that the two variables have some positive relationship, stronger as the correlation coefficient goes to 1.

Figure 5 shows the statistical correlation coefficient between the characteristics discussed in this section and packet congestion. Figure 5 indicates great diversity across the benchmark suites and networks.

In the figure, the TRIPS OCN trace results are mixed. While a significant positive correlation is shown with packet congestion and injection rate spectral density and transient source rate variance there is little correlation shown with structural source rate variance. These benchmarks show a significant negative correlation with both transient and structural destination rate variance with packet congestion. The OCN traffic is predominately first-level to second-level cache system traffic. The second-level cache of the TRIPS processor was partitioned such that applications with small second-level cache footprints can be cached entirely in the top half of the cache, close to the processor in use. Applications in need of more second-level cache are more evenly spread across both the top and bottom halves of the cache. This leads to the counter-intuitive result that applications which evenly spread traffic out across the full OCN are those which also have greater packet congestion because more of the traffic must cross the north-south bisection, a bottleneck due to OCN network's skewed aspect ratio.

The SPLASH-2 traces and to a lesser extent the PARSEC traces show correlations between packet congestion and injection rate spectral density as well as transient source rate variance and both transient and structural destination rate variance. These results indicate network performance in these workloads depends upon the presence or absence of source and destination node hot-spots as well as program phases. Benchmarks that have diversity in their source or destination nodes injection rates or injection rate with time tend to have greater latencies than those which do not. In the multiprocessor workloads, high source or destination rate variance is indicative of either multi-threaded load imbalances or hot memory in those benchmarks.

Generally, the correlation results between the new workload characteristics and congestion corroborate the influence that spatial and temporal imbalances in injection rate have upon the performance of NOCs. Interestingly, the characteristics which correlate best with performance vary greatly from network to network. The characteristics which most strongly correlate with performance for a particular network's traces are indicative of potential weaknesses of those networks under those loads. For example, the SPLASH-2 benchmarks appear to be sensitive to phase behavior in the injection process, while the performance of the PARSEC benchmarks are dominated by source node hotspots. Results that point toward design improvements that could be made in each network to improve performance on these workloads.

## 6 Related Work

While much work exists in statistical modeling of realistic workloads and synthetic traffic generation in the WAN/LAN literature [22, 23, 24], there are relatively few such works for NOCs. Varatkar and Marculescu examined the self-similarity of MPEG-2 video applications on a system-on-chip NOC and used this information to develop a simple model for the pairwise communication of this application [25]. Soteriou, Wang and Peh examined the self-similarity, packet hop count and injection rate variance of realistic workloads [26]. More recently Bahn and Bagherzadeh developed a traffic model based upon packet hop count, aggregate injection rate combined with the self-similarity of each node's injection rate individually to capture the spatial distribution of packet injection separately from the temporal distribution [27]. Although somewhat orthogonal to this work, also of interest are several recent efforts which have begun to improve benchmark suites and simulation environments for NOCs more generally [28, 29, 30].

In contrast to prior work, the characteristics developed here evaluate the temporal distribution of traffic in terms of the frequency domain impact caused by cyclic program phase behavior. Program phase behavior is largely missed in self-similar traffic analysis because self-similarity metrics do not capture events that are cyclic in nature. Also unlike prior work, these characteristics are evaluated in terms of their correlation between traffic characteristics and their effect on packet latency.

## 7 Conclusions

Networks-on-chip (NOCs) are used in several contexts, from replacements for buses in uniprocessors to cache coherence networks in chip multiprocessors. Each of these different NOCs must be designed to accommodate the traffic that traverses them, therefore, understanding the characteristics of typical workloads is of paramount importance in producing a well balanced network design. This paper examined the benchmark suite driven workloads from TRIPS OCN and multiprocessor NOCs with respect to the following key characteristics, (1) Injection rate spectral density, (2) Transient source node injection rate variance, (3) Structural source node injection rate variance, (4) Transient destination node ejection rate variance and variance and (5) Structural destination node ejection rate variance and variance. These characteristics were shown to have a stronger correlation with network performance than traditional aggregate measures of traffic load.

We showed that, viewed through the workload characteristics examined, the synthetic benchmarks do not appear to model realistic benchmarks such as the TRIPS and multiprocessor benchmarks very well. Based upon these observations one potential use for workload characteristic information is to inform the design of the network under examination. For example, workload characteristics could be used to determine the networks that may need to be over-provisioned to avoid potential network congestion, and what form of over-provisioning would be most beneficial.

Interestingly, the characteristics which correlate best with performance vary greatly from network to network. The characteristics which most strongly correlate with performance for a particular network's traces are indicative of potential weaknesses of those networks under those loads. For example, the SPLASH-2 benchmarks appear to be sensitive to phase behavior in the injection process, while the performance of the PARSEC benchmarks are dominated by source node hot-spots. Results that point toward design improvements that could be made in each network to improve performance on these workloads.

Surprisingly, the characteristics developed in this paper are network independent workload characteristics. Network independent workload characteristics can be determined solely through an examination of a trace of network activity listing source, destination, packet size and injection time. This result implies that one can determine the potential network bottlenecks *before* the network topology and routing algorithm have been defined, simply through an off-line examination of sample workload traces, and use that information to assist in the selection of topology and routing algorithm.

## 8 Future Work

We are currently pursuing two avenues to extend the work presented here:

**Heterogeneous Network Design** The existence of *structural*, and to a lesser extent *transient*, injection and ejection rate imbalances indicates the potential for workload characteristic driven heterogeneous network design. For example the existence structural destination node ejection rate imbalances which correlate with performance for a given network can be used to inform the need for increased bandwidth for the ejection ports of those nodes which see increased traffic. By provisioning the extra bandwidth only to those nodes which form performance bottlenecks in the system, improved performance can be achieved while minimizing impact on power and area consumption.

**Improved Synthetic Workloads** As shown in this work, existing synthetic workloads are poor proxies for network performance under realistic workloads. We are investigating the generation of synthetic network traffic based upon the characteristics developed here. This new synthetic workload would allow performance testing of networks at an early development stage, in simulation, without the requisite extremely long simulation times required by realistic workload traces.

# References

[1] G. Moore, "Cramming more components onto integrated cicuits," *Electronics*, pp. 114–117, April 1965.

[2] "World's first quad-core processors for desktop and mainstream servers." http://www.intel.com/quad-core/index.htm.

[3] T. Ainsworth and T. Pinkston, "On Characterizing Performance of the Cell Broadband Engine Element Interconnect Bus," in *The 1st ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pp. 18–29, 2007.

[4] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, and J. Zook, "TILE64 - Processor: A 64-Core SoC with Mesh Interconnect," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 88–598, Feb. 2008.

[5] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger, "Implementation and Evaluation of On-Chip Network Architectures," in *IEEE International Conference on Computer Design (ICCD)*, 2006.

[6] P. Gratz, K. Sankaralingam, H. Hanson, P. Shivakumar, R. McDonald, S. W. Keckler, and D. Burger, "Implementation and Evaluation of a Dynamically Routed Processor Operand Network," in *The 1st ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pp. 7–17, 2007.

[7] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., first ed., 2004.

[8] J. L. Henning, "SPEC CPU2000: Measuring CPU Performance in the New Millennium," *IEEE Computer*, vol. 33, no. 7, pp. 28–35, 2000.

[9] V. Soteriou, N. Eisley, and L.-S. Peh, "Software-Directed Power-Aware Interconnection Networks," in *The 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pp. 274–285, 2005.

[10] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: characterization and methodological considerations," *SIGARCH Compututer Architecture News*, vol. 23, no. 2, pp. 24–36, 1995.

[11] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express Virtual Channels: Towards the Ideal Interconnection Fabric," in *The 13th International Symposium on Computer Architecture (ISCA)*, pp. 150–161, 2007.

[12] K. Strauss, X. Shen, and J. Torrellas, "Flexible Snooping: Adaptive Forwarding and Filtering of Snoops in Embedded-Ring Multiprocessors," in *The 33rd International Symposium on Computer Architecture (ISCA)*, pp. 327–338, 2006.

[13] N. Eisley, L.-S. Peh, and L. Shang, "In-Network Cache Coherence," in *The 39th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 321–332, 2006.

[14] P. Gratz, B. Grot, and S. W. Keckler, "Regional Congestion Awareness for Load Balance in Networks-on-Chip," in *The 14th International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 203–215, 2008.

[15] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite:Characterization and Architecutural Implications," in *Technical Report TR-911-08*, Princeton University, January 2008.

[16] B. Grot, J. Hestness, O. Mutlu, and S. W. Keckler, "Express Cube Topologies for On-Chip Interconnects," in *The 15th International Symposium on High-Performance Computer Architecture (HPCA)*, 2009.

[17] T. Mattson and M. Wrinn, "Parallel programming: can we PLEASE get it right this time?," in *The 45th annual conference on Design automation (DAC)*, pp. 7–11, 2008.

[18] A. J. KleinOsowski and D. J. Lilja, "MinneSPEC: A New SPEC Benchmark Workload for Simulation-Based Computer Architecture Research," *Computer Architecture Letters*, vol. 1, 2002.

[19] T. Sherwood, E. Perelman, and B. Calder, "Basic Block Distribution Analysis to Find Periodic Behavior and Simulation Points in Applications," in *The International Symposium on Parallel Architectures and Compilation Techniques (PACT)*, 2001.

[20] T. Sherwood, S. Sair, and B. Calder, "Phase Tracking and Prediction," in *The 30th annual international symposium on Computer architecture (ISCA)*, pp. 336–349, 2003.

[21] M. P. Norton and D. G. Karczub, *Fundamentals of noise and vibration analysis for engineers*. Cambridge University Press, second ed., 2003.

[22] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[23] C. L. Williamson, "Network Traffic Measurement and Modeling," in *The 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pp. 56–57, 1995.

[24] V. J. Ribeiro, R. H. Riedi, M. S. Crouse, and R. G. Baraniuk, "Simulation of nonGaussian Long-range-dependent Traffic using Wavelets," *SIGMETRICS Performance Evaluation Review*, vol. 27, no. 1, pp. 1–12, 1999.

[25] G. Varatkar and R. Marculescu, "On-chip Traffic Modeling and Synthesis for MPEG-2 Video Applications," *IEEE Transactions on VLSI Systems*, vol. 12, pp. 108–119, Jan. 2004.

[26] V. Soteriou, H. Wang, and L.-S. Peh, "A Statistical Traffic Model for On-Chip Interconnection Networks," in *The 14th IEEE International Symposium on Modeling, Analysis, and Simulation (MASCOTS)*, pp. 104–116, 2006.

[27] J. H. Bahn and N. Baherzadeh, "A Generic Traffic Model for On-Chip Interconnection Networks," in *The First International Workshop on Network-on-Chip Architectures*, 2009.

[28] C. Grecu, A. Ivanov, P. Pande, A. Jantsch, E. Salminen, U. Ogras, and R. Marculescu, "Towards Open Network-on-Chip Benchmarks," in *The First International Symposium on Networks-on-Chip (NOCS)*, p. 205, 2007.

[29] S. Mandal, N. Gupta, A. Mandal, J. Malave, J. Lee, and R. Mahapatra, "NoCBench: a Benchmarking Platform for Network on Chip," in *WORKSHOP on UNIQUE CHIPS and SYSTEMS (UCAS)*, 2009.

[30] E. Salminen, C. Grecu, T. Hamalainen, and A. Ivanov, "Application Modelling and Hardware Description for Network-on-Chip Benchmarking," *IET Computers & Digital Techniques*, vol. 3, no. 5, pp. 539–550, 2009.