
The Network Processing Forum Switch Fabric Benchmark Specifications: An Overview

Itamar Elhanany, Derek Chiou, Vahid Tabatabaee, Raffaele Noro, and Ali Poursepanj

Abstract

The Network Processing Forum chartered a fabric benchmarking task group to establish a set of switch fabric benchmark specifications that allows the characterization of a wide range of switch fabrics for diverse networking applications. A unique characteristic of the benchmarks is their ability to produce comparable performance results for different switch fabrics, regardless of their underlying architecture and technology. This article provides an overview of the NPF fabric benchmark specifications by describing the various topics addressed by the standard as well as their potential impact.

The Network Processing Forum (NPF) was initiated in February 2001 “to encourage the growth and effective use of network processing technology through standards, testing, benchmarking and education.” Although switch fabrics are fundamentally different than network processors, they were deemed critical to the performance of switching platforms where network processors are employed. Consequently, a fabric benchmarking task group was launched within the benchmarking working group.

The switch fabric is a central communication highway in network switches and routers. Logically, a typical fabric consists of three main building blocks, as shown in Fig. 1. The first building block is the ingress interface that connects an ingress line card’s network processing/traffic management subsystem to the switching engine. The second building block is the switching engine that enables multiple ingress ports to be dynamically linked to egress ports. The third building block is the egress interface that concentrates traffic from the switching engine and forwards the traffic to an egress line card’s network processing/traffic management subsystem.

Many fabrics today support the CSIX interface [1]. The CSIX interface specification was defined by the CSIX industry consortium, which eventually evolved into the NPF. Although the NPF benchmarks are defined around CSIX interfaces, they are easily adaptable to other fabric interfaces such as the recently ratified NPF Streaming Interface (NPF-SI) specification. Throughout this article, however, we assume a CSIX-compliant switch fabric. In addition to presenting the NPF CSIX fabric benchmarks, this article includes additional discussion on topics that were extensively debated within the fabric benchmark task group but excluded from the standard.

In general, a unit of data traversing a CSIX device is called a CFrame. A CFrame consists of a header section and a data (payload) section. The header contains all relevant management information, including routing and priority status. Packets arriving from various sources are first segmented into CFrames and forwarded to the switch fabric. Upon exiting the switch fabric, packets are reassembled. This process is commonly referred to as segmentation and reassembly (SAR).

Switch fabrics can be implemented in a variety of ways ranging from a shared memory architecture to a fully distributed nearly stateless architecture, and anywhere between [2–5]. One of the most challenging aspects of the benchmarking effort was to define a set of stimuli models, performance metrics, and test methodologies that produce comparable results across a wide range of possible architectures. Similar to other testing methodologies [6], the approach taken here yields comparable results across virtually any fabric architecture.

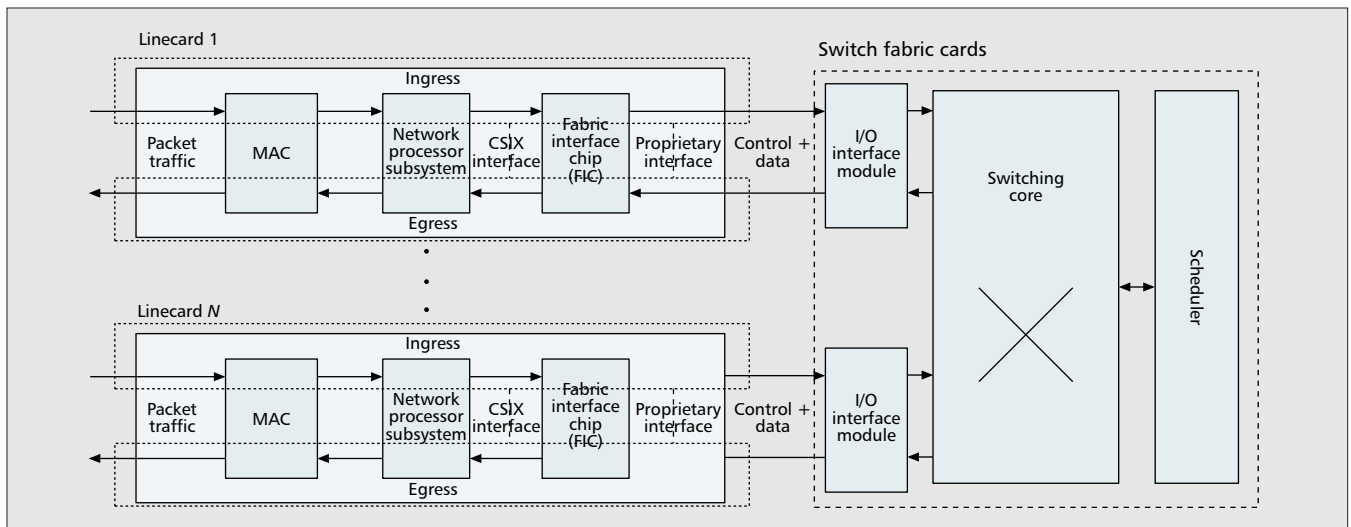
The primary enabler for producing comparable results regardless of architecture is the definition of an architecture-independent *black box model* with a twist, specifically adding a pseudo-traffic-manager (pTM) to each ingress port as illustrated in Fig. 2. The need for a black box model arises from the wide variety of internal implementations that can characterize switch fabrics. To create coherence and uniformity in testing, each fabric is treated as a black box with CSIX ingress ports and egress ports. By strictly defining the various parameters of the black box, including number of ports, data rates, and other fundamental attributes, a canonical fabric specification is attained. Moreover, a key advantage of the black box model lies in its ability to measure the switch fabric’s performance without requiring knowledge of its internal architecture.

Traffic Modeling

Model Fundamentals

An elementary step in evaluating switch fabrics is the establishment of a comprehensive set of traffic models. Modeling real-life traffic is a tremendously difficult task as the nature of traffic varies greatly with network type, protocol, time of day, and other diverse parameters. A main focus of the NPF fabric benchmarking task group has been to develop a set of statistical characteristics according to which traffic models can be explicitly described.

In recent years, there has been much academic and industrial research work on modeling Internet traffic. Although this task remains a primary research topic, several statistical models for pragmatic traffic generation have been introduced [7,



■ Figure 1. A generic switch fabric architecture comprising an ingress path, an egress path, and a switching engine.

8]. Rather than trying to accurately portray network traffic patterns, the aim has been to cover a wide range of scenarios that, when applied to different fabrics, accentuate the differences between them. These models all assume that the traffic arrival patterns are independent of the switch state.

Destination Distribution

Traffic is modeled at the packet level where each packet is mapped into one or more CFrames by the pTM. Since each test bench suite may focus on studying a different set of performance characteristics, a distinct traffic model is associated with each test bench scenario. To that end, several key statistical characteristics have been defined for each model including the packet destination distribution and the manner in which arriving packets are distributed among the destinations.

Real-life packet destinations are not uniformly distributed; rather, traffic tends to be focused on preferred or popular destinations. As such, a flexible destination distribution model based on Zipf's law has been proposed [9, 10]. Zipf's law states that the frequency of occurrence of some events (P), as a function of the rank (i), where the rank is determined by the above frequency of occurrence, is a power-law function: $P_i \sim i^{-k}$, with the exponent, k , close to unity. It was shown that many natural and human phenomena such as Web access statistics, company sizes, and biomolecular sequences all obey Zipf's law with k close to 1. Recent studies [7] show that Internet traffic also obeys the Zipf law. Zipf's law states that the probability that an arriving packet is heading toward destination i is given by

$$\text{Zipf}(i) = \frac{i^{-k}}{\sum_{j=1}^N j^{-k}},$$

where i is the packet destination, k is the Zipf order, and N is the system order (i.e., the number of switch ports); $k = 0$ corresponds to uniform distribution. As k increases the distribution becomes more biased toward preferred destinations. Since the Zipf model pertains to a single source, aggregating the traffic arriving from all sources while ensuring admissibility yields the traffic arriving to a given output.

Packet Arrival Process

In addition to the nonuniformity observed with respect to the packet destination distribution, the traffic pattern by which packets arrive at a given port is significant. In particular, the burstiness of Internet traffic has been studied extensively [7]. Real-life traffic tends to be bursty due to the diversity of mod-

ern networking applications and aggregation of traffic in the network. The NPF has adopted a set of bursty models that are both simple to implement in hardware and software platforms and flexible enough to cover a diverse range of traffic scenarios. To that end, the following arrival processes have been included:

- Bernoulli i.i.d. arrivals — representing the ideal case of uncorrelated packet streams
- Leaky bucket — a simplistic bursty model in which deterministic trains of packets arrive with a predefined burst size
- ON/OFF model — a basic two-state Markov modulated arrival process in which packet bursts are geometrically distributed in length
- Modified ON/OFF model — an extension to the basic ON/OFF model that offers 100 percent throughput via non-delimited burst streams
- ON/OFF model with a minimal burst size — an additional extended ON/OFF model that enforces a minimal burst size on any arrival of a packet burst

Multicast and QoS

The specification includes a simplistic multicast generation process described in the following three steps:

- Determine whether the packet/burst is of a multicast type.
- Draw a number for the multiplicity factor (realistically should not exceed 10 with an average value of 2–4).
- For each multiplicand, the destination is chosen with respect to the destination distribution.

With multicast modeling it is important to avoid unintentional oversubscription. Consequently, the offered load, as defined by the arrival process, is to be bounded. For example, for 10 percent multicast with average multiplicity of 4, the unicast traffic must not exceed 60 percent of the maximum load.

Performance Metrics

Metric Definitions

While modeling incoming traffic is an important component in benchmarking fabrics, a necessary complement is the manner in which the performance of a switch is measured. In today's environment, it is often possible for one fabric vendor to present performance results that are measured and calculated in a way that is very different from those presented by other vendors, thus making it hard to accurately compare different products. To avoid confusion, the definition of each metric should be consistent across fabrics being benchmarked as well as the benchmarks

themselves. Thus, another goal was to establish an unambiguous specification that defines performance metrics such as latency and jitter, as well as the way they are to be measured.

There are three primary performance metrics considered with respect to the black-box model: latency, accepted vs. offered bandwidth, and jitter. Latency and jitter both have two sets of measurement points that produce four submetrics: fabric latency, total latency, fabric jitter, and total jitter. We begin with a generic definition of latency as the time taken for a CFrame to traverse between two points in the reference model. Latency is specified in seconds. There are two forms of latency, described below. The difference between total and fabric latency is where the measurement is taken. The points referred to in the following metric definitions are illustrated in Fig. 2.

- *Fabric latency* is measured between the CSIX interface where CFrames are inserted (point 2) and the CSIX interface where CFrames are extracted (point 3).
- *Total latency* is measured between the output of the SAR unit (point 1) and the interface where CFrames are extracted (point 3) minus the flow-through latency of the pTM. Thus, if there is no backpressure, total latency is identical to fabric latency. Latency is defined to be from the start of the CFrame (RxSOF for fabric latency) at the source to the start of the CFrame (TxSOF for both total and fabric latency) at the destination.
- *Accepted vs. offered bandwidth* is the number of CFrames the fabric accepts at point 2 divided by the number of CFrames offered to it at point 1. Since this metric is a simple ratio, it has no unit of measure. Accepted vs. offered bandwidth is a ratio of the number of CFrames inserted into the fabric divided by the number of CFrames inserted into the pTM. Such a definition eliminates ambiguity of using a raw bandwidth number such as 10 Gb/s, since it is unclear whether the latter incorporates CFrame fragmentation losses, headers, and so on. Moreover, the ratio clearly illuminates whether the fabric is forwarding everything it is offered or dropping some fraction of the offered traffic.
- *Jitter* is the difference in the time interval between a pair of consecutive CFrames belonging to the same flow at the ingress and the time interval between that same pair of CFrames at the egress. Jitter is also specified in seconds. It is important to emphasize that jitter pertains to the difference in latency between sequential CFrames belonging to the same flow. As with latency, the time is always measured from the start of the CFrame.
- *Fabric jitter* measures the ingress time interval at point 2 and the egress time interval at point 3.
- *Total jitter* measures the ingress time interval at point 1 and the egress time interval at point 3.

Pseudo-Traffic Manager and Latency Measurements

Latency is difficult to define for a broad range of switch fabrics. Measuring latency between points 2 and 3 ensures that the performance results are “all fabric” and avoids introducing testing infrastructure performance artifacts. However, there is justification for measuring latency between points 1 and 3 as an expression of total latency. The reason is that the pTM inherently introduces testing infrastructure latency into the overall latency, although it also (arguably) increases the accuracy of the latency metric. To that end, it has been decided to include both metrics as valid and valuable performance indicators.

As mentioned earlier, the fabric latency metric pertains to CFrames that have been accepted by the fabric. It is quite possible for two fabrics to have precisely the same latency for CFrames that have been accepted by the fabric, but due to different scheduling algorithms yield vastly different throughput. One clear benefit of the fabric latency metric is that it is easier to implement since it is only measured for CFrames

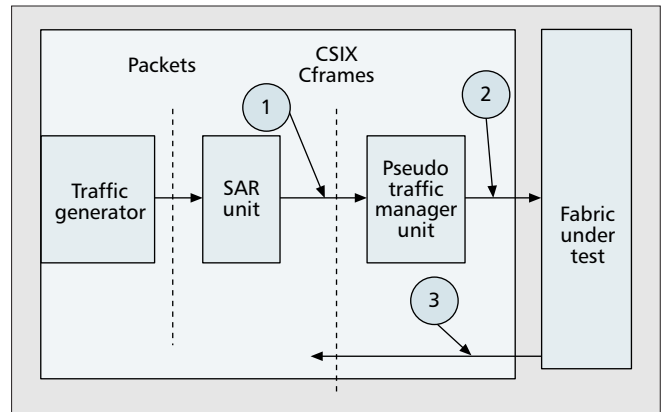


Figure 2. The NPF fabric benchmark black box reference model.

that arrive at the ingress of the fabric.¹ Consequently, if the fabric is overloaded and discarding CFrames within the pTM, the semantics of fabric latency is consistent and easy to understand. CFrames that do not make it out of the pTM are not counted. Thus, the total latency metric, like the fabric latency metric, only includes CFrames that arrive at the ingress (and therefore the egress) of the fabric.

Including the pTM as part of the latency metric requires not only a strictly defined traffic manager, but also a faithful implementation of the specification. Such implementation should be fairly straightforward if simulated, but may be challenging for hardware testing, posing a drawback of the approach. To that end, the following traffic manager attributes have been specified with respect to the pTM:

- It incurs a fixed latency when there is no backpressure from the fabric.
- It enqueues CFrames in designated channel queues when there is backpressure from the fabric.
- It arbitrates between waiting CFrames in a work conserving round-robin fashion. The pTM must be able to send back-to-back CFrames on all channels that are not currently back pressured.
- It tail-drops packets whose channel queues have reached a threshold.

The viability of these performance metrics depends on the existence of such a CSIX traffic manager. Since CFrames can be generated on the fly and do not require real data, the problem is likely to be more tractable than it might appear.

Testing Methodology

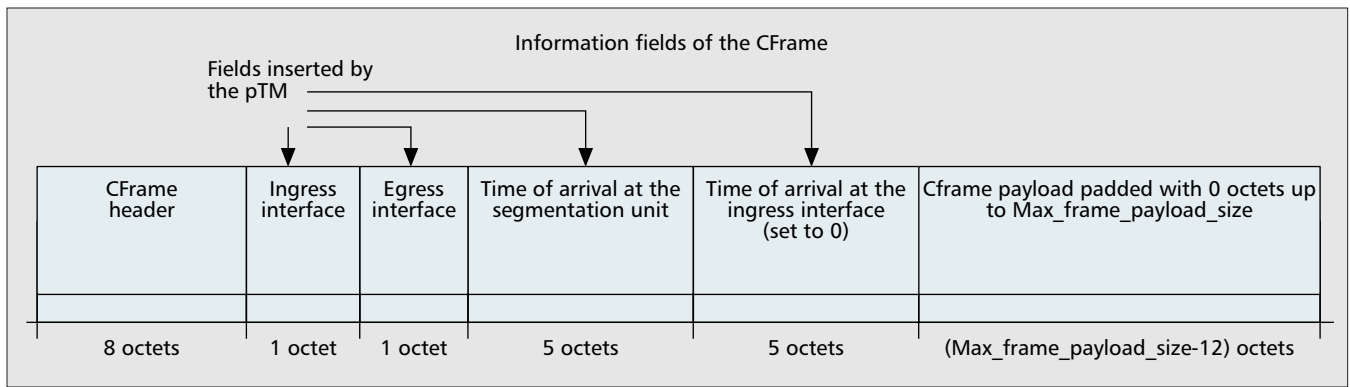
A testing methodology ensures that the result of each performance test is unambiguous, reliable, and, more important, replicable. The performance testing methodology is organized as a set of rules and instructions that are subdivided into three categories:

- Testing requirements rules for the features of traffic generation
- System configuration rules for setting up the pTM and switch fabric under test
- Data collection rules for gathering the performance data

The party in charge of the performance test (the switch fabric customer, switch fabric vendor, or a third-party performance tester) must provide a value for every parameter used in the test. This allows anyone to accurately replicate the performance test.

Testing requirement rules define the precise features of the

¹ CSIX fabrics are lossless; therefore, any CFrame that enters the fabric is assumed to arrive sometime later at the egress.



■ Figure 3. Format of the CFrame used in the performance tests.

traffic under test and the details pertaining to packet segmentation into CFrames. The latter is performed by the pTM, which also handles their subsequent scheduling for transmission to the ingress module. Figure 3 illustrates the format of the CFrame used in the performance tests.

System configuration rules define the features of the scheduling of CFrames by the pTM and the settings of the switch fabric under test. The pTM maintains a virtual output queuing (VOQ) structure for the scheduling of CFrames to the ingress CSIX interface. The aggregate memory size of the pTM is a parameter controlling the VOQ space, and must be specified for each performance test. Figure 4 depicts the VOQ structure of the pTM. The switch fabric under test is characterized by an arbitrarily long list of parameters. Since the standard does not favor any specific switch fabric architecture, this list of parameters varies from vendor to vendor. Therefore, it is the responsibility of the fabric vendor to provide a complete list of switch fabric parameters and offer either default values or a range of valid values for each.

Data collection rules address the manner in which performance data is collected. A traffic filter parses the information fields of the CFrames and separates the traffic under test from background traffic. The parameters identifying the traffic under test (ingress or egress CSIX interface, class, and type) must be specified for each compliant performance test. Also, each experiment should be long enough to reach a *stable state* and collect a significant statistical sample of performance data. The standard specifies a start time and a stop time based on the CFrame buffering capacity at the fabric under test. This ensures, within a reasonable error bound, that the fabric is in a stable state and sufficient statistics are gathered.

Benchmark Suites

Traffic models, performance metrics, and testing methodology are the imperative ingredients in developing test benches. Benchmark test suites are sets of well-defined experiments to be used in studying, evaluating, and comparing switch fabrics under different operational scenarios. In general, the performance of a switch fabric is determined by hardware, architecture, arbitration, and scheduling (contention resolution) mechanisms. The test suites have been partitioned into three categories, each focusing on a unique functional aspect of the switch fabric. The following sections describe the recognized aspects of fabric benchmarking.

Hardware Benchmarks

The first set of test suites focuses on the hardware and architectural aspects of the switch fabric. Hardware technology and architecture have a profound impact on some basic characteristics of the switch fabric, such as memory speed, processing speed, port-to-port minimum latency, switch fabric overhead,

internal cell size, and cell generation procedure. It is not possible to distinguish and study these factors completely separately; however, in order to minimize the impact of the scheduling and arbitration mechanisms, there is no contention between packets generated at the same time in these test suites. In other words, all CFrames generated at the same time at different ingress ports have separate destinations. These tests quantify:

- The zero load latency, which is the minimum latency a CFrame can experience when there is no contention
- The buffering in the fabric
- The maximal port load, which denotes the maximum stable arrival rate a switch fabric can support when there is no contention between CFrames

Arbitration Benchmarks

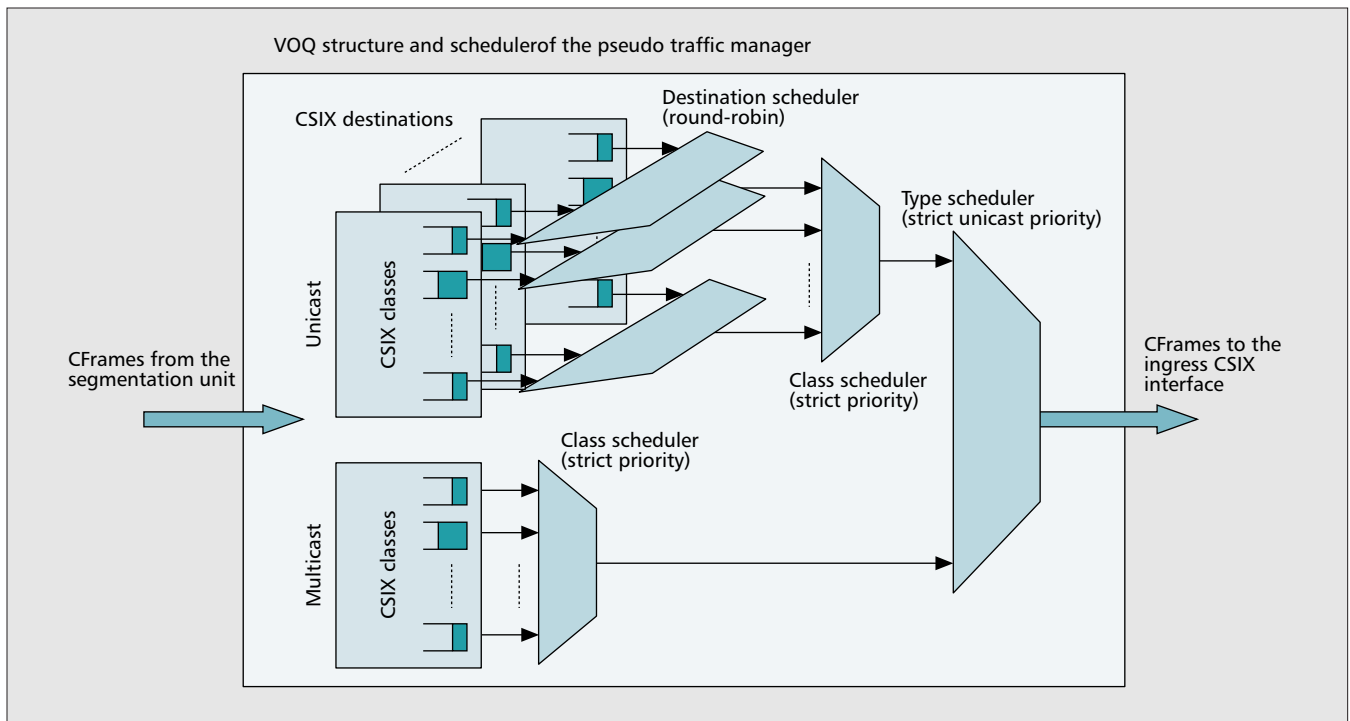
Arbitration benchmark tests study the performance of the switch fabric when there is contention caused by CFrames generated by and/or destined for the same port. The arbitration mechanism is one of the key challenges in switch fabric design and has a significant impact on the performance of the fabric as it determines which CFrames are to be forwarded and which buffered at the contention points. At the ingress side of a switch fabric, contention occurs between the CFrames that share the same ingress port but are destined for separate egress ports. Similarly, at the egress side, contention occurs between CFrames that are destined for the same egress port but originate from separate ingress ports. In addition, depending on the architecture, there can be some internal contention points inside the fabric.

In the arbitration benchmarks, we study the switch fabric performance with different traffic patterns and load distributions when contention occurs. The primary goal of these tests is to measure throughput, latency, and jitter of the fabric in different scenarios. In each scenario the traffic pattern is either bursty or nonbursty, and the load distribution is either uniform or nonuniform. It is essential to cover all these cases in the benchmark since, depending on the architecture and arbitration mechanism, the performance of a specific switch fabric can be good in some conditions but poor in others.

Multicast Benchmarks

The multicast benchmark test suites study fabric performance when presented with multicast traffic. These tests are further divided into two subgroups; in the first subgroup, performance of the multicast traffic is investigated when there is no unicast traffic, while in the second the impact of multicast traffic on the performance of unicast traffic, and vice versa, is studied.

When there is no overload or contention, a good multicast solution should be able to efficiently handle all traffic patterns. However, if the multicast solution relies on replication of multicast cells at the input port, it may generate artificial



■ Figure 4. The VOQ structure of the pTM.

overload at the input. The primary goal of these tests is to see how well the multicast scheduler performs under such conditions. In the second set of multicast tests, input traffic consisting of both multicast and unicast components is applied whereby the primary objective is to study the impact of multicast traffic on unicast traffic performance and vice versa.

Conclusions

This article provides a detailed overview of the Network Processing Forum's switch fabric benchmark specifications, from traffic models employed, through the performance metrics to the testing methodology and, finally, the benchmark suites. This effort constitutes the first attempt to design a comprehensive benchmarking framework for a general class of packet switching fabrics. By leveraging research in academia and industry, the specifications have been carefully crafted to be as neutral and pragmatic as possible. As future packet switching platforms evolve and become more diverse, it is expected that this standard will facilitate their comparison and evaluation.

Acknowledgments

The authors would like to thank all members of the Network Processing Forum who have been active in contributing to the efforts of the fabric benchmarking task group. In particular, the authors would like to extend their gratitude to Chris Bergen and Anita Weemaes, who have directly contributed in structuring and revising the different sections of the specifications.

References

- [1] CSIX-related material: <http://www.npforum.org>
- [2] N. McKweon *et al.*, "The Tiny Tera: A Packet Switch Core," *Proc. IEEE Hot Interconnects V*, 1996.
- [3] S. Keshav, and R. Sharma, "Issues and Trends in Router Design," *IEEE Commun. Mag.*, 1998, pp. 144–51.
- [4] I. Elhanany and D. Sadot, "DISA: A Robust Scheduling Algorithm for Scalable Crosspoint-Based Switch Fabrics," *IEEE JSAC*, vol. 21, 2003, pp. 535–45.
- [5] C. Minkenberg and T. Engbersen, "A Combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technology," *IEEE Commun. Mag.*, vol. 12, 2000, pp. 70–77.

- [6] R. Jain and G. Babic, "Performance Testing Effort at the ATM Forum: An Overview," *IEEE Commun. Mag.*, Aug. 1997.
- [7] C. Williamson, "Internet Traffic Measurement," *IEEE Internet Comp.*, vol. 5, 2001, pp. 70–74.
- [8] C. Barakat *et al.*, "On Internet Backbone Traffic Modeling," *Proc. ACM Sigmetrics*, 2002.
- [9] L. Breslau *et al.*, "On the Implications of Zipf's Law for Web Caching," *Proc. IEEE INFOCOM '99*, 1999.
- [10] G. K. Zipf, *Psycho-Biology of Languages*, Houghton-Mifflin, MIT, 1965.

Biographies

ITAMAR ELHANANY (itamar@ieee.org) received B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering and an M.B.A. degree, all from Ben-Gurion University in Israel. He is an assistant professor in the Department of Electrical and Computer Engineering at the University of Tennessee. During 2000–2003 he was with TeraCross, where he was involved in the development of Terabit-per-second switch fabric integrated circuits. His primary research interests include packet scheduling algorithms, switch architectures, and performance analysis.

DEREK CHIOU is an assistant professor in the Department of Electrical and Computing Engineering at the University of Texas at Austin. He received S.G., S.M., and Ph.D. degrees in electrical engineering and computer science from MIT. From 2000 until 2004 he was a system architect at Avici Systems, responsible for system architecture, fabric architecture, and performance modeling. His research interests include router architecture, parallel computer architecture, computer architecture, and simulation technology.

VAHID TABATABAEE received his B.Sc. degree from Sharif University of Technology, his M.Sc. degree from Tehran University, and his Ph.D. from the University of Maryland at College Park, all in electrical engineering. From 2001 to 2003 he was a principal switch fabric architect at Zagros Networks. He is currently a researcher with the University of Maryland Institute for Advanced Computer Studies (UMIACS). His interests include scheduling, traffic management, routing, and QoS provisioning for communication networks.

RAFFAELE NORO received a Laurea degree in electronics engineering from the University of Trieste, Italy, in 1995, and a Ph.D. degree in communication systems from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2000. He is currently a member of technical staff in the Network Products Division, Vitesse Semiconductor Corp., Camarillo, California. His research interests are in high-speed packet switching architectures, QoS, and multiservice platforms.

ALI POURSEPANJ (alip@poursepanj.com) is a system performance engineer in the System Architecture Group of Motorola Freescale Semiconductor. He has been chairman of the Switch Fabric Benchmarking Task Group of the NPF since 2001. He received his Ph.D. degree in electrical and computer engineering from the University of Texas at Austin. His research interests are processor and system architecture, performance analysis, and workload characterization.