
Requirements & Performance Metrics for Next Generation Switch Fabrics

December 2002

Author: **Vahid Tabatabaee**



**Zagros Networks
2 Research Place
Rockville, MD 20850**

www.zagrosnetworks.com

© 2002 Zagros Networks. All rights reserved.

About the Author

Vahid Tabatabaee is the co-founder and principal architect at Zagros Networks. He received his B.S. degree from Sharif University of Technology, M.S. degree from Tehran University, and is a Ph.D. candidate at University of Maryland at College Park. He has published several papers in the communication networks field. His area of research is focused on scheduling and quality of service (QoS) provisioning techniques for high speed switch fabrics and cable modems.

About Zagros Networks

Zagros Networks is a fabless semiconductor company headquartered in Rockville, Maryland focused on building silicon and software solutions for next generation metro and edge communications systems.

Two leading venture groups, nationally recognized Mohr, Davidow Ventures and leading regional fund Novak Biddle Venture Partners, backed the company in its Series A round. Zagros Networks was founded in April 2001 by Nariman Farvardin who is the Dean of Engineering at the University of Maryland and Chairman of the Board at Zagros.

The company is fundamentally changing the way system OEMs are designing their next generation platforms by providing the industries first rate-aware crossbar switching fabric and a unique software environment to leverage the power of this rate-aware architecture.

Zagros' technology allows system OEMs for the first time to displace the use of costly, low performance shared memory fabric architectures utilized today when service guarantees through the system are an absolute requirement. The company's chipset is targeted at replacing all ASIC, shared memory and merchant crossbar architectures in systems today.

Eight patents are pending or in process for the core intellectual property that enables our unique approach to not only guaranteeing bandwidth but also controlling latency, providing interflow isolation and enabling dynamic link provisioning.

Table of Contents

1.0	Introduction	3
2.0	Scheduling Performance Analysis for Switch Fabrics	4
2.1	Scheduling Stages Interdependence in a Switch	4
2.2	Performance Criteria for Next Generation Switch Fabrics	5
2.2.1	Throughput	5
2.2.2	Latency	7
2.2.3	Jitter	7
2.2.4	Compatibility with Line Card Schedulers	7
3.0	Rate-aware Scheduling	9
3.1	Line Card Schedulers	9
3.2	Rate-aware Scheduler in Shared Memory Architectures	10
3.3	Rate-aware Scheduler in Buffered Crossbar Architectures	11
3.4	Rate Awareness in Crossbar Architectures with Fairness Based Schedulers	11
3.5	Crossbar Architecture with Rate-aware Arbitration - The Right Solution!	12
4.0	Simulation Results	14
4.1	Guaranteed Bandwidth Delivery	14
4.2	Controlled Latency	16
4.3	Interflow Isolation	17
5.0	Conclusion	18
6.0	References	18

1.0 Introduction

The switch fabric is one of the key elements in data network switches and routers, and has a great impact on the overall performance of the system. Every data packet is processed and scheduled in three places in a system: ingress line card, egress line card, and the switch fabric. A switch fabric is in charge of scheduling and switching data from the ingress line cards to the egress line cards. In this white paper, we discuss the performance requirements and features that are important for next generation switch fabrics.

The first issue to address is to define a set of key metrics that define performance for next generation switch fabrics. The next generation fabrics should be capable of accommodating requirements of revenue generating applications that will be run over packet networks. We consider the following as the three main performance metrics:

1. Delay
2. Jitter
3. Throughput

For each performance metric, we elaborate on the real requirements and the levels of control and quality required for the parameter in the fabric. We intentionally do not mention packet loss as a metric, since switch fabrics are generally designed to be lossless; however, delay and throughput characteristics of the switch fabric have a direct impact on the packet loss characteristics of the overall system.

Besides performance controllability, compatibility between the switch fabric and the line card(s) traffic shapers and schedulers is another basic requirement for next generation switch fabrics. Since switch fabrics do not work in isolation, it is not sufficient to ensure that they perform adequately by themselves. We have to make sure that the overall system functions and performs correctly. This paper defines three levels of compatibility requirements between line cards and switch fabrics, which are listed below:

- Queue size compatibility
- Functional compatibility
- Flow control support

We introduce the concept of rate-aware switch fabrics as a methodology to provide all of the required features and characteristics for switch fabrics. A rate-aware switch fabric is capable of provisioning, measuring, controlling, and monitoring the bandwidth allocated and used by each of the switch fabric queues. We review available switch fabric architectures and scheduling techniques and we conclude that none provide the critical rate-aware functionality. We also show that the unique features of the switch fabrics make it impossible to effectively apply line card, rate-aware scheduling techniques, and reveal a need for the design of new switch fabric rate-aware scheduling algorithms.

Zagros Intelligent Bandwidth Arbiter (ZIBA), the switch fabric scheduling solution from Zagros Networks, is introduced as a new scheduling technique that is inherently rate-aware, and fulfills all other requirements of switch fabric schedulers. Finally, we provide some simulation results that illustrate and compare the performance of ZIBA with other common scheduling techniques.

2.0 Scheduling Performance Analysis for Switch Fabrics

2.1 Scheduling Stages Interdependence in a Switch

Figure 1 illustrates a simplified diagram of the data path in a switch. There are three main stages of scheduling in a switch: ingress line card, egress line card, and the switch fabric. Usually there are multiple layers of scheduling in each stage, and each of these layers has a direct impact on the overall performance of the scheduler.

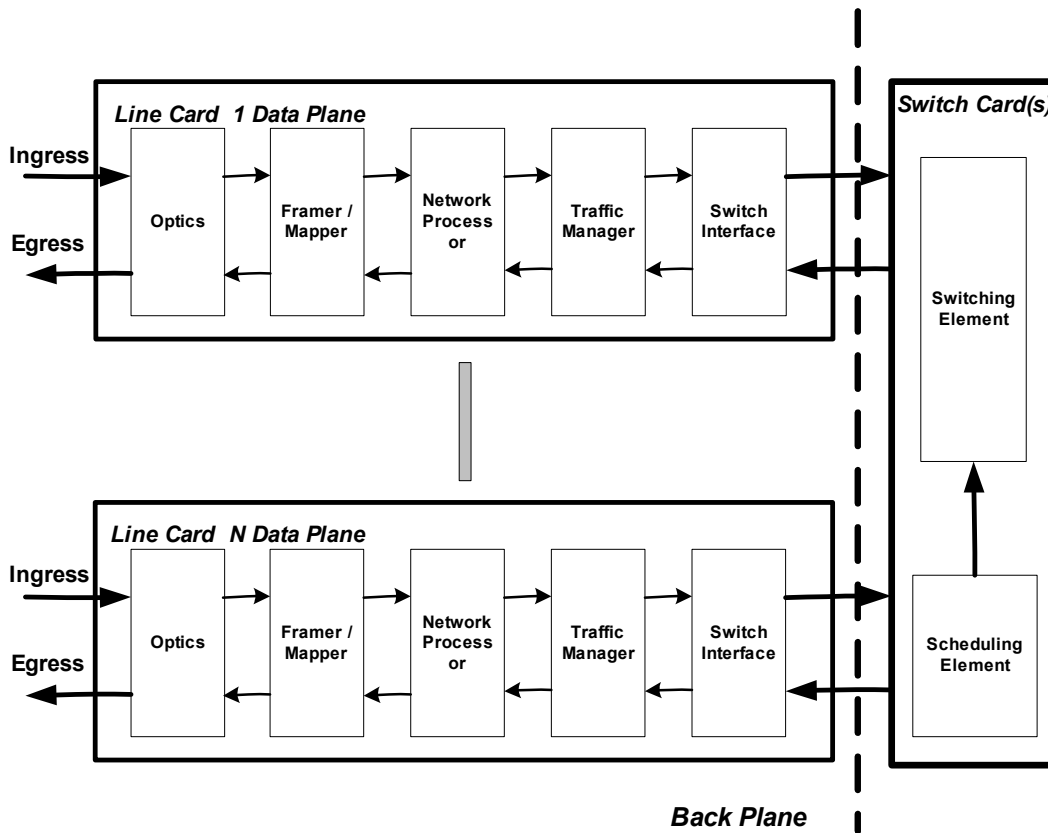


Figure 1. Simplified Diagram of a Data Path in a Switch

Both the scheduler architecture and the interconnection architecture impact the performance characteristics of the switch. For instance, the characteristic of a Weighted Fair Queueing (WFQ) scheduler in a hierarchical architecture, usually used in traffic manager applications, is different from an isolated WFQ scheduler [1]. Therefore, when comparing switch fabric alternatives, it is very important to analyze and understand the performance of all the schedulers as a system, and not to rely on their isolated characteristics and performance.

In order to predict and study the performance of a switch, we have to carefully select the architecture of the switch, as well as the scheduling and shaping methodologies used in each stage. Moreover, we should only rely on those characteristics of the scheduling algorithms that can extend to the adopted architecture. For

instance, if a Weighted Round Robin (WRR) scheduler is working under a non-work-conserving paradigm, it does not allocate the bandwidth appropriately among the contending flows, since it cannot account for idle times properly. Although this appears to be a simple and obvious design concept, it is usually missing in today's solutions especially in the switching fabric.

2.2 Performance Criteria for Next Generation Switch Fabrics

In this section we introduce the main performance metrics of next generation switch fabrics. Here, we focus only on those metrics that are directly related to the performance of the fabric. Besides performance, there are other implementation issues related to the switch fabric architecture that are equally important. These issues are not discussed here, but are covered in another white paper from Zagros Networks, "Switch Fabric Design Selection Analysis". For further information about this white paper, refer to [Section 6.0, "References," on page 18](#).

2.2.1 Throughput

Throughput of a switch fabric is the maximum percentage of the switch fabric interface bandwidth that can be utilized for data transmission in a stable regime. A switch is stable if the departure rate of data from the fabric equals arrival rate of data. For instance, the line card interface to a fabric may be able to work at 16 Gbps raw data rate; however, when load distribution is non-uniform, the system cannot operate higher than 10 Gbps without data loss. In this case throughput of the system is as low as 0.625 (10/16) Gbps.

For a given input traffic pattern and load distribution, we define throughput of a switch fabric as the ratio of the maximum stable data rate to the maximum possible data rate on the fabric interface. A data rate is stable if the queue lengths remain finite for that rate. Notice that we normalized throughput to the switch fabric interface data rate, not the incoming line rate. For instance, for a CSIX interface fabric with a 16 Gbps interface rate that is working on a 10 Gbps line rate, the normalizing factor is 16 Gbps rather than 10 Gbps. This may be counter-intuitive at the first glance, however, one should take into account that the traffic manager, CSIX interface, and cell tax overhead, as well as other factors, increase the incoming data rate to the fabric beyond the line rate (refer to [Figure 2, "Data Overhead in a Switch," on page 6](#)). In other words, a switch fabric interface should work at a rate higher than the line rate to be able to support the line rate at the line card interface. The converting factor varies from system to system, but as a general rule, we consider the interface data rate of the fabric as the normalizing factor for the throughput.

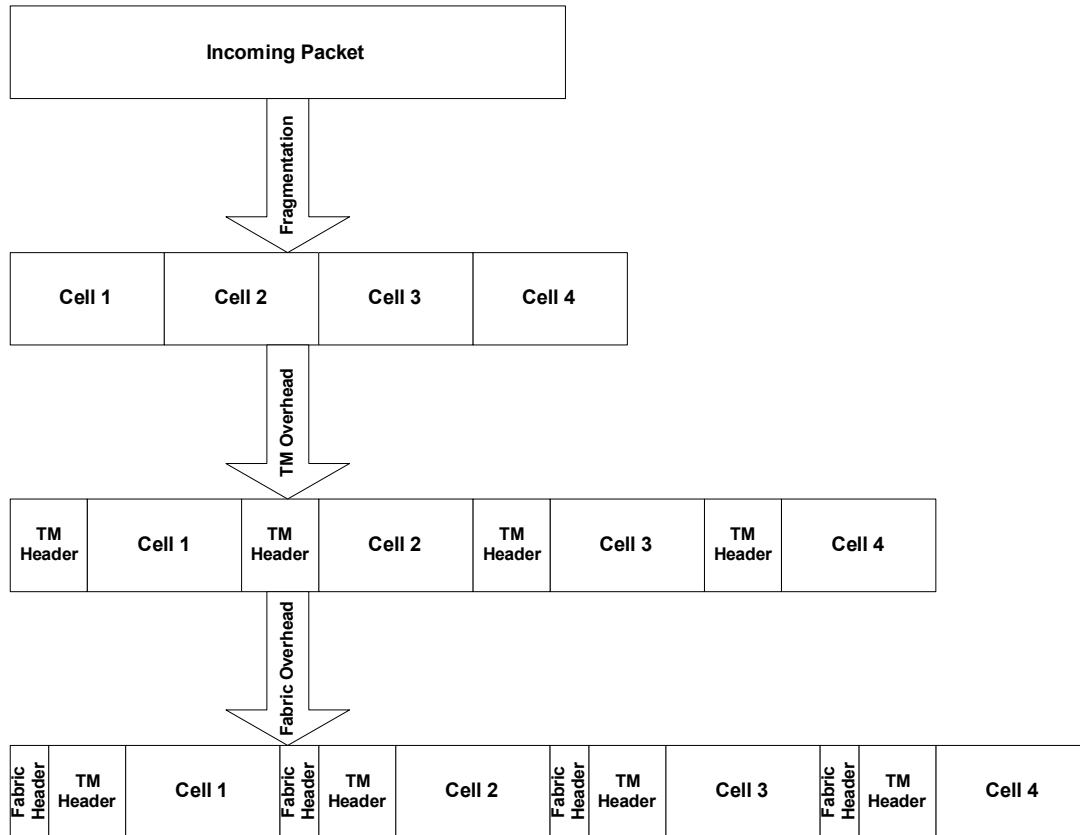


Figure 2. Data Overhead in a Switch

As is clear by the definition, throughput of a switch is defined as a function of the input traffic pattern and the load distribution. It is a misnomer that non-uniform load distribution and bursty traffic patterns are more challenging for switch fabrics and always cause the worst case throughput. In fact, there are switch fabric architectures and scheduling algorithms that are customized to work well with bursty traffic patterns and/or non-uniform load distributions. As a result, it is very important to study switch fabrics carefully and comprehensively to assure robust performance regardless of the traffic pattern and load distribution. Equally important is that the switch fabric does not rely on a specific traffic pattern as the benchmark.

The importance of robustness becomes more evident if we notice that we cannot assume any specific pattern for internet traffic. This is because internet traffic patterns depend on several variables that are subject to change. The problem is even more complicated, since the traffic pattern that is observed by the switch fabric not only depends on the network traffic pattern and distribution, but also on the data scheduling and shaping techniques used in the ingress line card. Therefore, even if we could consider a specific traffic pattern for the network traffic, the traffic pattern that is observed by the fabric may not be the same. This means that traffic pattern and load distribution of a fabric are unpredictable and the switch fabric should be robust and should perform well with all possible traffic patterns.

2.2.2 Latency

Many of the revenue-generating applications that will be deployed over future packet networks are delay sensitive. For instance, real time and interactive applications such as video streaming, video conferencing, and Voice over Internet Protocol (VoIP) will not gain broad range acceptance unless the service providers can provision and predict the latency on their network. The latency requirement is not only to have average low latency, but more importantly, to be able to control, predict and bound the delay of all system queues. We define latency of a switch fabric as the difference between the time that a cell enters the ingress port of the fabric and the time it departs from the egress port of the fabric.

The performance metric is the average latency. Usually system designers only look at the average overall latency of the fabric, which is a necessary but not a sufficient latency metric. For instance, when traffic is non-uniformly distributed among different destinations, some connections can experience excessive latency, while the average overall latency is still acceptable. In order to have enough granularities, we have to measure the latency over individual links (input and output connections) separately. Generally, it is desirable that a link that carries more traffic at a higher rate experiences lower delay. The average number of backlogged cells in a queue equals the average latency times the arrival rate of that link. Therefore, to balance the number of backlogged cells in different queues, the higher the rate, the lower the latency of that link should be.

2.2.3 Jitter

Due to the burstiness of data traffic, aggregation of flows, statistical multiplexing, and scheduling techniques that are used in data networks, cells and packets that belong to the same flow experience different delays. This phenomenon is referred to as jitter. Jitter is an important performance measurement for real-time traffic, such as voice and video, and it must therefore be studied and be tested as a performance metric for next generation switch fabrics.

We define jitter as the absolute delay variation between two consecutive cells or packets of the same Virtual Output Queue (VOQ). In a packet switch network it is neither necessary nor efficient to have a switch fabric with zero jitter. The egress line card usually uses a dejittering buffer to shape and smooth the outgoing traffic. However, the larger the jitter, the larger the size of dejittering buffer, and consequently, the longer the latency. Therefore, it is important to be able to control and to limit the jitter of switch fabrics.

2.2.4 Compatibility with Line Card Schedulers

In [Section 2.1](#), we talked about three stages of scheduling in a switch and their overall impact on the performance. In order for a switch to have a desirable and predictable performance, there should be many levels of compatibility between these three stages of scheduling in the ingress line card, switch fabric, and the egress line card. Scheduling in the line card is a “many to one” selection problem, since the scheduler has to select one flow out of all the present flows every time. On the other hand, the scheduling in a switch fabric is inherently a “many to many” problem, since at every time the scheduler should decide which ingress ports get connected to which egress ports. Because of this fundamental difference, the scheduling techniques that are used in line cards are not smoothly extendable to the switch fabrics. In this section we will elaborate on different dimensions of compatibility between the line card and the switch fabric scheduler that are essential for coherent operation of a switch or router.

Queue size compatibility: Hierarchical scheduling is the common method used in line cards. [Figure 3](#) illustrates a three layer hierarchical scheduler. In practice, there can be more than three layers of scheduling. The first layer schedulers are in charge of selecting one of the flows connected to them. The second layer

schedulers invoke one of the first layer schedulers so that they select a flow in turn. Depending on the total number of flows and the scheduler complexities we can have more than three layers of scheduling.

In Figure 3, “p” is the number of schedulers in the next to the last layer of schedulers in the line card, and determines how much aggregation has taken place in the line card. It is desirable to limit the aggregation. The cells scheduled through the last layer of the line card scheduler are passed to the switch fabric queues. Therefore, to avoid further collapse (aggregation) of queues in the switch fabric, we need to have p separate queues for the unicast traffic in the switch fabric to have the queue size compatibility.

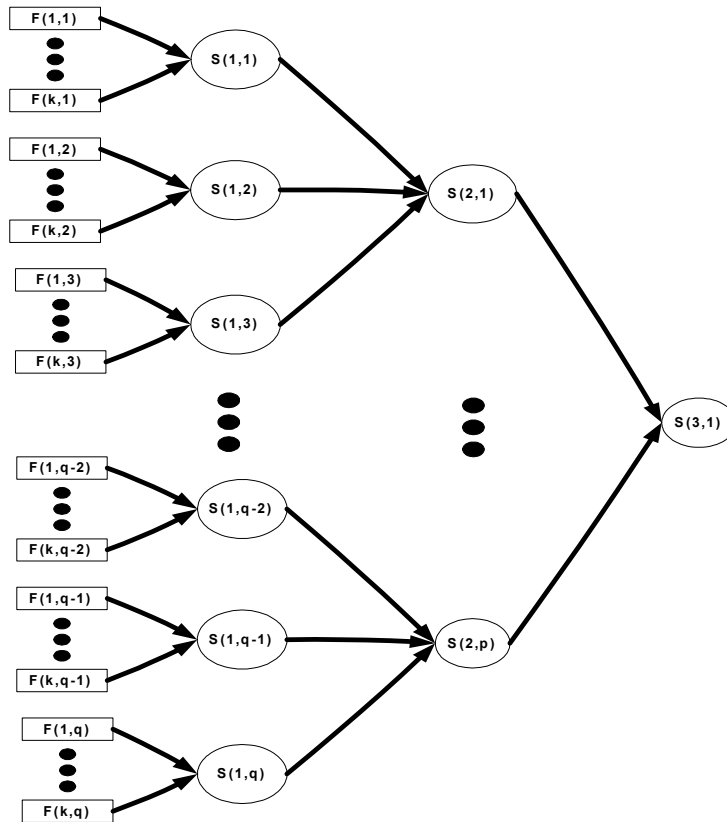


Figure 3. A Three-layer Hierarchical Scheduler

Functional compatibility: Besides the physical compatibility in the number of queues, there should also be functional compatibility between the scheduling in the line card and the switch fabric. The scheduling techniques that are used in the line cards, such as Weighted Round Robin (WRR) and Weighted Fair Queueing (WFQ) allocate a minimum certain amount of bandwidth to each flow. Similarly, the switch fabric schedulers should be able to allocate and provision bandwidth for each of the aggregated flows that are sent through the fabric.

Flow control support: Flow control granularity and accuracy are critical for appropriate functioning of the ingress line card schedulers. Line cards perform scheduling of their own flows, without any knowledge about the flows that are residing on other line cards. The switch fabric is the first place that all flows converge, and it has a global knowledge about the state of the flows. Therefore, a switch fabric can potentially monitor and

control the scheduling of the flows based on the global knowledge. It can in return, send a feedback message to the line card schedulers. Flow control is the only mechanism available to a switch fabric to influence the ingress line card schedulers.

There are two fundamental requirements for an effective flow control mechanism. First, the switch fabric should be able to individually flow control all of the flows that it can identify. For instance if the incoming traffic constitutes four levels of Class Of Service (COS) in a 16 x 16 switch, there would be 16x16x4 distinctive flows in the switch fabric, and the switch fabric should be capable of controlling each one of these flows. Secondly, it is not sufficient to be able to flow control each of these flows, but we need to be able to flow control each of them based on a quantifiable criterion. The most natural measure that is used for flow control is rate. Under this paradigm, a minimum serving rate is allocated to each of the flows, and flow control is only activated if the serving rate of a flow is greater than the assigned rate. To that end switch fabric should be able to measure and control the allocated rate to the individual flows.

In fact, most of the switch fabric solutions today lack the stated flow control requirements. For instance, in the shared memory architectures, queueing is usually done on per output, per class basis. Consequently, all traffic coming from different ingress ports with the same class and output port is buffered in the same queue. Therefore, there are not enough granularities to monitor, control, and flow control each one of the flows. Another example is a crossbar architecture with VOQs that utilizes a fairness based scheduler. Although it has the appropriate granularities in queue structure, it lacks a scheduler that is capable of monitoring and controlling rates, and therefore does not supply useful flow control information.

3.0 Rate-aware Scheduling

In the previous section, we elaborate on the basic requirements for the next generation switch fabrics. In this section, we introduce the rate aware scheduling concept for the switch fabrics and illustrate that it delivers all of those requirements.

Rate aware scheduling is the ability to schedule and measure the serving rate of the flows and to maintain the minimum guaranteed rate for each of the flows. The concept of rate aware scheduling is not new. In fact, as we will explain, the common scheduling techniques that are used in the line cards can be considered rate aware schedulers. However, these techniques do not work in the switch fabric framework, and we need to come up with new scheduling algorithms for that purpose.

3.1 Line Card Schedulers

Numerous scheduling techniques have been studied and used in the line cards. Even though these techniques are usually not extendible to the switch fabric schedulers, they can provide some valuable insight for the appropriate design methodology of switch fabric schedulers. One common trend that can be observed in most of the practical and successful scheduling techniques is the concept of proportional resource allocation. Two very good examples are Weighted Round Robin (WRR), and Weighted Fair Queueing (WFQ). In both of these algorithms and their extensions, a number (weight) is associated to each of the flows. The weight of a flow specifies the minimum allocated rate of that flow. Moreover, since these algorithms can guarantee the minimum serving rate of the flows, we would be able to predict and control latency, jitter, and throughput of the flows.

One important criterion that we stated for the switch fabric scheduler is compatibility with the line card scheduler. Conceptually, a rate aware switch fabric scheduler coordinates very well with the line card scheduler, and functions as the last layer of schedulers in a hierarchical scheduling architecture.

In summary, a rate-aware switch fabric scheduler enables system designers and users to provision, control and bound the critical performance measures, i.e. latency, jitter, and throughput throughout the switching system from port to port. In the following, we review some of the common switch fabric architectures and scheduling algorithms and their rate-awareness capability.

3.2 Rate-aware Scheduler in Shared Memory Architectures

Shared memory is one of the most common switch fabric architectures. Even though this architecture has major scalability and redundancy problems, our intent here is not to go over these disadvantages, and we refer the interested readers to another white paper from Zagros Networks, “Overview of Switch Fabric Architectures” (see Section 6.0, “References,” on page 18).

In order to have queue compatibility in a shared memory architecture there should be one queue for each flow in the shared memory. In practice, this is not the case; all flows destined for the same output port are buffered in the same queue. This is illustrated in Figure 4, “Schematic Architecture of a 2 x 2 Shared Memory Switch Fabric,” on page 10. Notice that VOQs are aggregated in the shared memory. Therefore, there is not enough queue granularity in shared memory architectures.

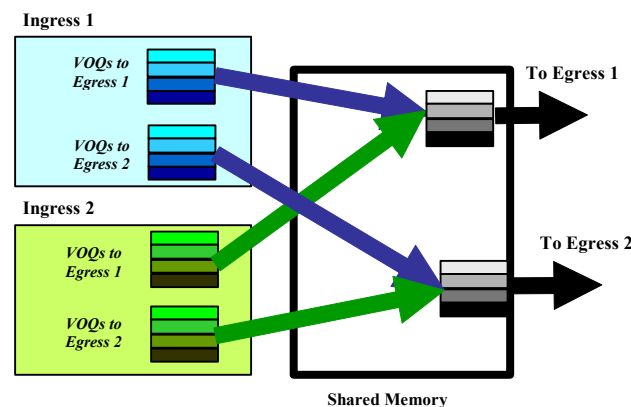


Figure 4. Schematic Architecture of a 2 x 2 Shared Memory Switch Fabric

In shared memory architecture, there are three different layers of scheduling, which are in the ingress interface unit, in the egress interface unit, and in the shared memory. The ingress interface scheduler unit selects one of the incoming flows and sends one cell from it to the shared memory. There are N schedulers working in parallel in the shared memory. Each of these schedulers is designated to one egress port and selects a cell to be sent to that egress port. The egress port scheduler selects outgoing cells from that egress port. The scheduling technique that is used in this architecture is usually similar to WRR or its variants.

The important point is that a WRR scheduler provides rate guarantees only under certain conditions that are not present in a shared memory switch fabric. For instance, consider a backlogged queue that is the first selection of the WRR scheduler. Even though this queue is backlogged and is the first choice of the WRR scheduler, it may not be served because there is an active flow control back pressure signal for that queue from the next stage. We will go over this problem in more detail in the next section, since this problem is more dramatic in the buffered crossbar architecture.

3.3 Rate-aware Scheduler in Buffered Crossbar Architectures

Buffered crossbar architectures can be considered as an extension of the shared memory architecture, where the central memory unit constitutes of queues per input, output and Class Of Service (COS). For instance, a 32×32 buffered crossbar switch with two classes of service has 2048 queues in the buffered crossbar unit. Due to the technology limitations, number of queues that can be implemented in the buffered crossbar unit is limited and it does not provide sufficient granularity. For a fixed number of ports, this means that there is a limitation in the number of classes of service. This is the main reason why we see smaller number of classes of service in buffered crossbar architecture, which in turn results in incompatibility with line card schedulers.

Capacity of the buffered crossbar queues is another limiting factor. This number is so low that the buffered crossbar queues hit their flow control threshold, and interrupt flow of the cells very often. The buffered crossbar queue size and the loop back delay between the buffered crossbar unit and the ingress line card determines how often the flow control signal is activated. For each cell time the buffered crossbar should send one bit of SEND/NOTSEND for each one of the queues to the corresponding ingress line card. As an example, if the unused capacity of a queue is 4 cells, and the loopback delay is 10, there cannot be more than 4 SEND messages out of the last 10 messages sent back to the line card for that particular queue. This means that it is quite possible that the memory space is available in the buffered crossbar unit, and cells are buffered in the ingress line card, but still they are not sent and are not scheduled due to the NOTSEND signal received from the buffered crossbar. Therefore, the WRR schedulers that are working in the ingress line card and in the buffered crossbars are not really work-conserving, and cannot provide rate guarantees anymore. This fact will be illustrated in the simulation results.

Internal cell size is yet another practical limiting factor in the buffered crossbar architectures. In order to reduce the access time to the buffered crossbar memories, designers have to increase the cell sizes. However, increased cell size results in more fundamental and dramatic problems. As the cell size increases, cell tax overhead factor increases as well. For instance, for a 96-byte payload the overhead is so much that it is not possible to maintain the line rate for 40-byte packet sizes (24.192 million of “40-byte packets” per second), even with 2X speedup. Consider that payload size is 96 bytes, the cell header is 16 bytes and the switch fabric has 20Gbps internal data rate (2X speedup). Using this data, the maximum possible cell rate for the switch fabric is 22.232 million cells per second. Even for 8 bytes of cell overhead, the maximum cell rate is 24.038 million cells per second and still cannot support line rate of 24.192 million of “40-byte packets” per second.

Another potential source of overhead is flow control bits. The buffered crossbar needs to send one bit of flow control overhead per queue to each one of the corresponding queues of a line card. This results in excessive overhead if this information is carried in the frame header. For instance, if there are 64 queues per ingress line card, this results in 8 bytes of overhead. The common information that should be carried on the header is around 8 bytes, so the flow control messaging increases the cell overhead by 100%. Notice that this is just for a 64 individual queues; in a 32×32 switch with four classes of service we need 128 bits for flow control messaging. The problem becomes more dramatic for the next generation switch fabrics with extensive sub-porting requirements. If we consider a 32×32 switch with 4 classes of service and 4 sub-ports, flow control requires 512 bits of overhead, which is basically impractical.

3.4 Rate Awareness in Crossbar Architectures with Fairness Based Schedulers

Crossbar architecture is considered to be one of the most common and suitable architectures for high-speed switch fabrics [3]. One of the main challenges of crossbar fabrics is the scheduling (arbitration) algorithm that is used for the crossbar. The scheduling problem in shared memory and in buffered crossbar architecture is a many to one problem. Therefore, the same scheduling solutions used in line cards can also be used for these

architectures, even though in the previous sections we showed that they cannot provide rate guarantees anymore. In the crossbar architecture, scheduling is intrinsically a many-to-many problem. In other words, the arbiter determines a match between input and output ports at every time slot.

Most of the proposed arbitration solutions that are currently used in the crossbar fabrics are fairness based schedulers. The most widely used one is iSLIP, which is based on simple Round Robin (RR) arbiters with slight modifications. Other arbitration solutions that are used in today's crossbar architectures are similar to iSLIP and have the same basic properties and performance.

Besides crossbar arbitration, there will be another level of scheduling in the ingress line card device. The scheduling techniques that are used in the ingress line card unit of the switch fabric are usually strict priority, or variants of WRR. In this model, ingress line card schedulers select one of the VOQs that are backlogged and send a request for it to the crossbar arbiter. The problem is that even if the WRR scheduler selects a particular VOQ for scheduling, that VOQ may not be served. Consequently, WRR cannot be considered as a rate-aware scheduler for the switch fabrics, since it is not working under a work-conserving environment in the ingress line card unit of the switch fabric. If the crossbar arbiter does not grant the queue, it will not be served. However, the WRR scheduler considers that flow as being served and is not capable of keeping track of rate requirements anymore. In general, the WRR scheduler on a line card schedules flows based on the status of the flows that are residing on that line card, regardless of the status of other line card flows.

The only way to have a rate-aware switch fabric is to have a rate-aware arbiter for the crossbar as well as rate-aware scheduler for the line cards.

3.5 Crossbar Architecture with Rate-aware Arbitration - The Right Solution!

In the previous section, we went over the possible solutions that proposed to achieve a rate-aware switch fabric. We explained why none of them can deliver a rate-aware switch fabric and why they all have fundamental problems. In the white paper "Overview of Switch Fabrics" (refer to [Section 6.0, "References," on page 18](#)), the benefits and advantages of the crossbar architecture are compared to the alternative solutions and it is concluded that the crossbar switch fabrics are the best suited architecture for the next generation switch fabrics. Therefore, it is desirable to design a rate-aware crossbar arbiter to achieve a real rate-aware switch fabric with the appropriate architecture for next generation packet networks.

There are several sequential rate-aware arbitration algorithms that are proposed in the literature. The problem with sequential algorithms is their sequential nature. These algorithms are executed in multiple steps, where the number of steps is proportional to the number of ports or in some cases the number of ports squared. This results in major scalability problems in terms of the number of ports for high speed switches. Therefore, these algorithms are not appropriate for next generation switch fabrics.

Parallel arbitration algorithms such as iSLIP and its extensions are the dominant choice for the high speed switch fabrics. The problem is most of these solutions are based on a RR scheduler that is not rate-aware. Zagros Intelligent Bandwidth Arbiter (ZIBA) is Zagros Networks patent pending, parallel, work-conserving rate-aware arbitration solution for crossbar scheduling. ZIBA is capable of providing rate guarantees up to the 95% of the switch capacity, regardless of the input traffic pattern (bursty or non-bursty), and traffic load distribution (uniform and non-uniform). ZIBA functionality is complemented by a WF2Q+ scheduling algorithm in the line card interface device in the Zagros Z1 switch fabric. As a result, the Z1 switch fabric can provide minimum bandwidth guarantees to each one of 1024 ingress unicast queues per line card. As shown in [Section 4.0, "Simulation Results," on page 14](#), this provides 10 times more guaranteed bandwidth compared to today's scheduling algorithms.

Similar to iSLIP, ZIBA is also a parallel, crossbar scheduling algorithm consisting of grant and accept arbiters. The iSLIP arbiters are basically RR schedulers, and cannot maintain rate guarantees. ZIBA arbiters measure the serving rate of ingress/egress connections, compare them with allocated rates to the flows, and always select the most eligible links to serve.

Consider the 4×4 switch fabric in Figure 5. If the arbiters use simple RR schedulers, under contention bandwidth is equally shared between contending ports. In Figure 5, all ingress ports have a request for egress port 1, and, as a result, each of them receives 25% of bandwidth. In essence, there is no controlling mechanism over bandwidth allocation among contending flows. On the other hand, in ZIBA we can configure the arbiters and allocate the bandwidth arbitrarily between contending flows. In Figure 5, the rate reservation matrix specifies how bandwidth should be allocated between contending flows. At the bottom of the figure the cell delivery sequence illustrates that ZIBA divides the bandwidth accordingly. We must emphasize that ZIBA is a work-conserving scheduler; therefore, if some connections are not using their bandwidth, ZIBA distributes the excess bandwidth among requesting connections.

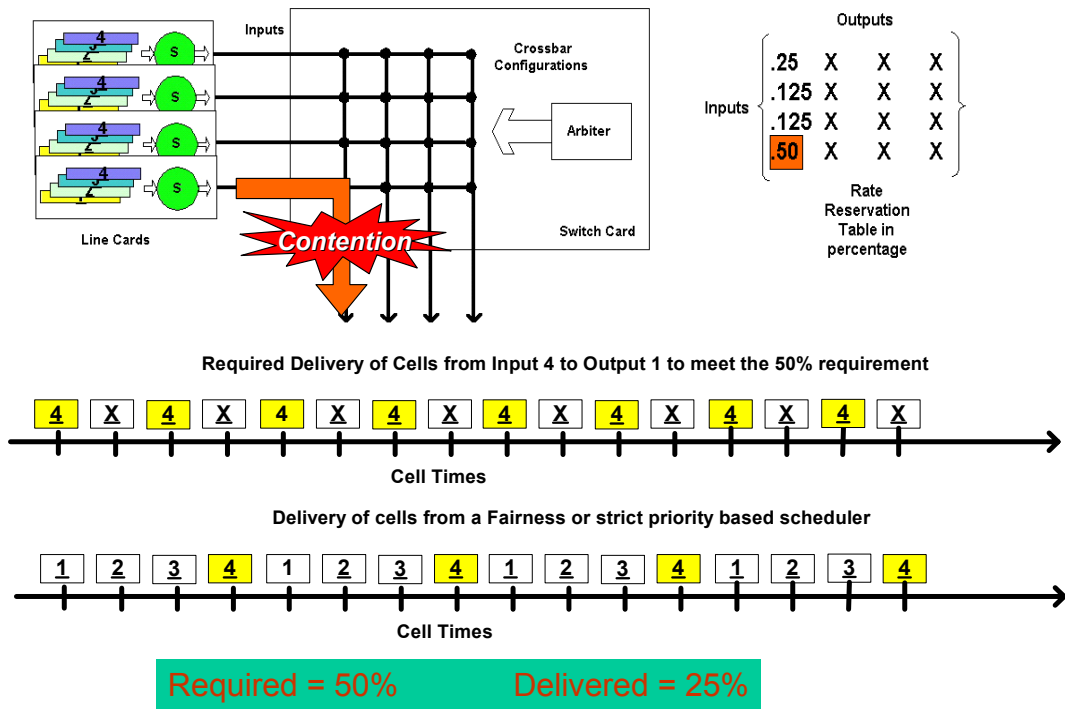


Figure 5. Cell Delivery Under Contention for a Fairness Based Arbiter and a Rate-aware Arbiter

The rate tracking module of ZIBA arbiters employ novel patent pending update rules that avoid synchronization problem between the arbiters. Synchronization is a common problem for crossbar parallel arbiters, and happens when multiple arbiters select the same input or output port. Synchronization results in low throughput of the switch fabric. Therefore, ZIBA not only provides rate guarantees for individual queues but also provides a high throughput for the whole switch fabric.

4.0 Simulation Results

In this section, we demonstrate some simulation results for the ZIBA scheduling algorithm and compare it with a fairness based scheduler. Both schedulers are used for a 32×32 switch fabric.

4.1 Guaranteed Bandwidth Delivery

The first attribute of a rate-aware switch fabric is capability of providing rate guarantees. In the first experiment we consider a non-uniform (log-diagonal) pattern for bandwidth guarantees over the links. Log-diagonal load distribution is a very common, non-uniform benchmark pattern for switch fabrics. Here we have used this pattern to generate a minimum guaranteed bandwidth matrix for the 32×32 switch fabric. We compute the ratio of the real delivered bandwidth to the guaranteed bandwidth for each one of the 32×32 input/output connections, and use the minimum value as the benchmark metric. If the ratio is greater than one, all of the guaranteed bandwidths are satisfied. As soon as the benchmark value drops below one, the guaranteed bandwidths are violated.

For the simulated pattern the fairness based arbiter delivers a guaranteed bandwidth up to 1 Gbps, which is 10% of the total line bandwidth, while ZIBA delivers guaranteed bandwidth up to 9.5 Gbps ([Figure 6](#)). This means that with ZIBA, we can provision, control, allocate, and sell 95% of the total line capacity. Note that with a fairness based arbiter the egress port utilization can reach 100% of utilization. However, this is unmanageable utilization, since when there are contending flows, there is no control on how bandwidth is distributed among them.

We have also simulated performance of the fairness based arbiter with 3X speedup. As the chart clearly demonstrates, speedup would not change the guaranteed bandwidth capability of the fairness based arbiter. This should be expected, since speedup can increase throughput of the fabric up to the line rate, but beyond that speedup would be limited by the flow control mechanism in the switch. Speedup is necessary for switch fabrics to provide additional bandwidth that is required for cell overhead, and arbitration inefficiency. We explained the cell overhead sources in [Section 2.2.1](#) (see [Figure 2](#)). In switching systems, cell overhead can easily consume more than 30% of the bandwidth and speedup is required to compensate for this. Speedup can also help to cover arbitration inefficiency. For instance, an iSLIP arbiter performance can be less than 65% of the line rate for non-uniform load distributions; speedup can generate enough headroom for the switch fabric to provide 100% throughput.

However, this is still unmanageable utilization. Speedup does not compensate for the intelligence that is required to allocate bandwidth according to the reserved rates between the contending flows. When contention occurs, a fairness based arbiter allocates bandwidth equally between the contending flows, regardless of the speedup factor. In fact, under contention, due to the speedup factor, the switch fabric sends data at a higher rate than the egress port can serve, until the flow control mechanism stops the switch fabric from sending traffic to the egress port. In essence, the flow control mechanism adjusts the speedup factor of the switch fabric to the line rate by swapping the rate of the switch fabric to the egress port connection between zero and the maximum speedup rate.

In [Figure 7](#), guaranteed bandwidth delivery capability of a buffered crossbar switch fabric with a WRR scheduler is compared to a crossbar fabric with a ZIBA scheduler. For the buffered crossbar fabric, we are assuming that all three stages of scheduling in ingress line card interface unit, buffered crossbar unit, and the egress line card interface unit are WRR schedulers. Also, we are assuming that the cell size and the overhead of both fabrics are the same; even though, as explained before, this is not the case and buffered crossbar demands more overhead.

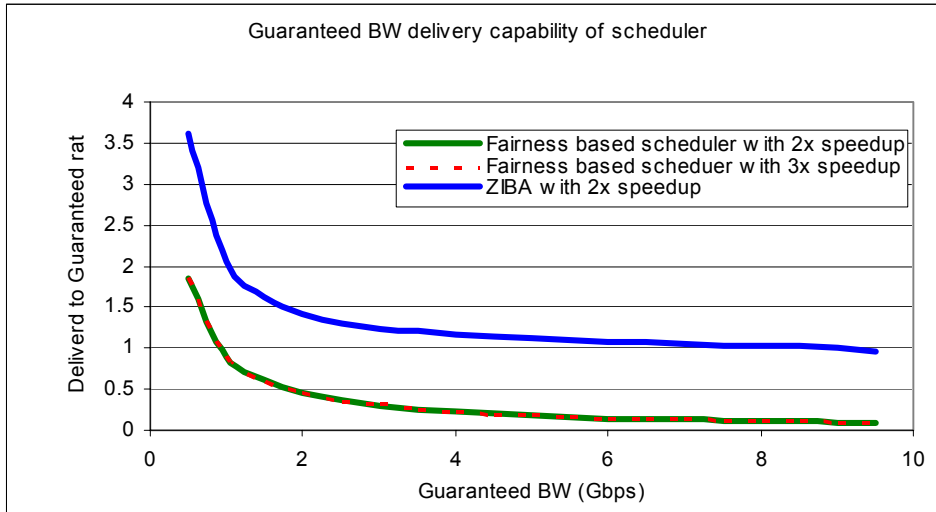


Figure 6. Minimum Guaranteed Bandwidth Delivery Ratio for ZIBA and Fairness Based Scheduler

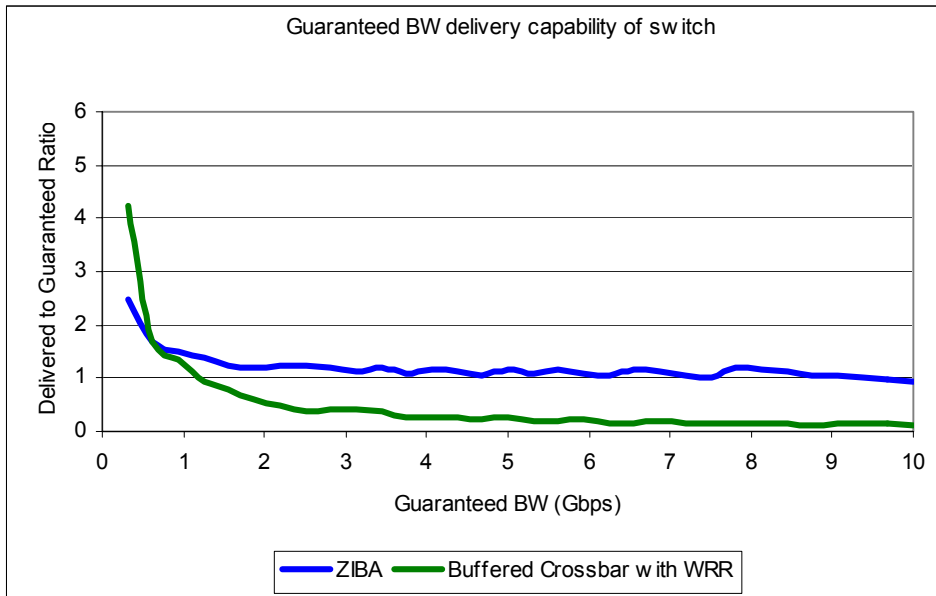


Figure 7. Minimum Guaranteed Bandwidth Delivery Ratio for ZIBA and Buffered Crossbar with WRR

Simulation Results

The guaranteed bandwidth load distribution matrix that is used here is a slight variation of what is called the diagonal load distribution. The diagonal entries of the matrix and their cyclic adjacent element of the matrix are shown below:

$$R(i, i) = 1.99 L / 3 \quad \text{where } i = 0, \dots, N-1$$

$$R(i, (i+1) \bmod N) = 0.99 L / 3 \quad \text{where } i = 0, \dots, N-1$$

All other elements of the matrix are $(0.02 \times L / 90)$, where L is the aggregate guaranteed bandwidth of a port. The buffered crossbar violates the guaranteed rates, when the total guaranteed bandwidth for a line is around 1.2 Gbps, while ZIBA provides guaranteed bandwidths up to 9.5 Gbps.

4.2 Controlled Latency

One of the important attributes of the rate-aware switch fabrics is controlled latency. In packet switch networks, due to statistical multiplexing, aggregation, and burstiness, traffic patterns vary in time and are not predictable. Obviously, under these circumstances it is not possible to control and bound latency of the packets. However, a controllable delay bound is desirable for those traffic flows that are shaped. In fact, this is one of the fundamental properties of line card schedulers such as WRR and WFQ.

A rate-aware switch fabric provides bounded latency through the fabric. To illustrate this feature, we measure the average latency for a specific link with fixed allocated rate and arrival pattern, but we increase the arrival rate over other links so that the aggregate rate of the corresponding egress port increases (refer to [Figure 8](#)). The average latency for the fixed rate link is bounded below 300 cell times, regardless of the aggregate rate. For comparison, the same parameter for a fairness based scheduler is also provided. For the fairness based scheduler latency goes as high as 100,000 cell times.

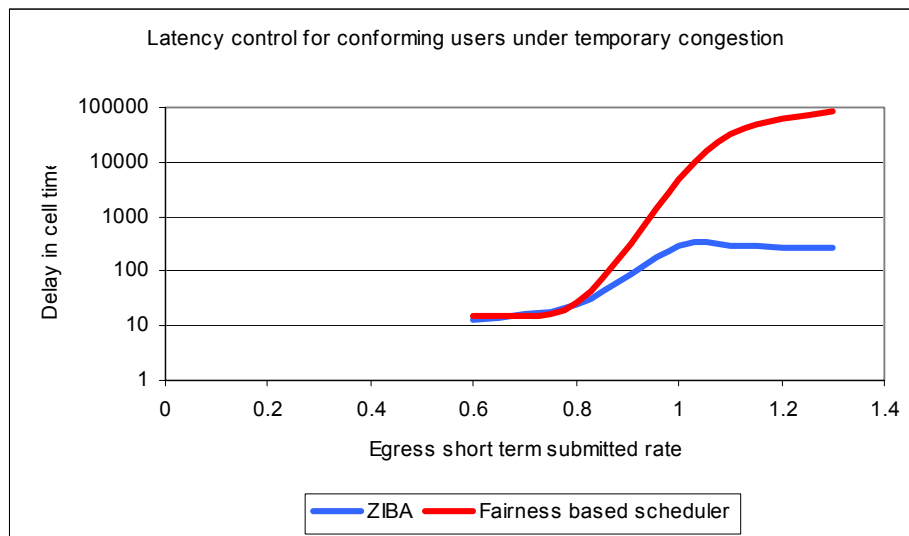


Figure 8. Average Latency of a Conforming Fixed Rate Link

4.3 Interflow Isolation

The other attribute of rate-aware scheduling is interflow isolation. Contrary to the circuit switched networks, in a packet network all flows share the bandwidth using statistical multiplexing (queueing) techniques. Therefore, quality of service of a flow depends on the state of the network and the behavior of other users. For instance, if a flow increases its rate, it consumes more buffer space in a switch that can result in packet drops for other users. A rate-aware scheduler guarantees a minimum serving rate for all flows. As long as the average arrival rate of that flow is below the reserved rate, that is, the flow is conforming to the provisioned rate, then the flow should not experience extensive delay and buffering.

Consider the following experiment: a 32 x 32 switch with two non-conforming flows, each with an incoming rate that exceeds their reserved rates. As a performance metric, the number of backlogged cells for six conforming flows as well as the two non-conforming flows is shown in [Figure 9](#).

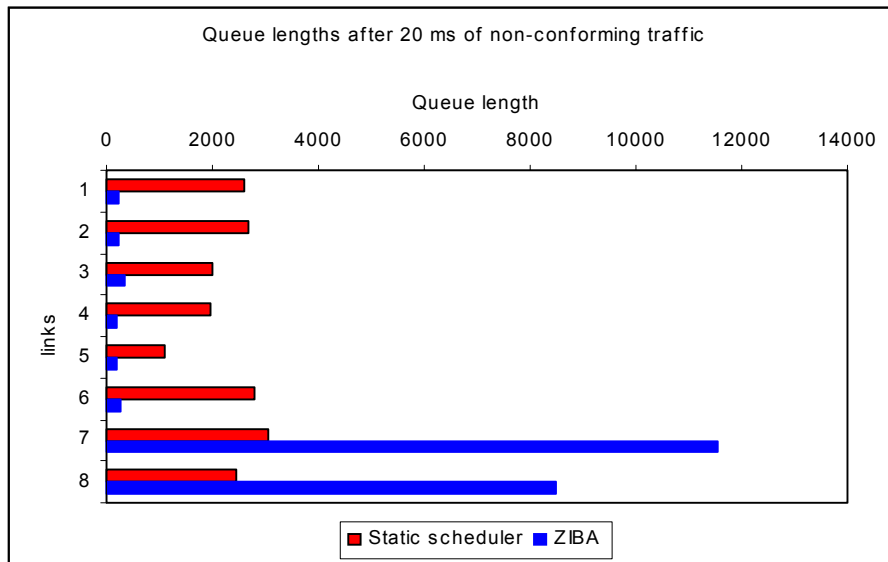


Figure 9. Queue Length of Conforming and Non-conforming Queues

Flows 1 through 6 are conforming and flows 7 and 8 are non-conforming. For the ZIBA scheduler only the two non-conforming flows experience excessive backlog, while for the fairness based scheduler, all eight flows are backlogged. In a practical system, because of the finite buffering space, this results in packet drop and excessive delay.

5.0 Conclusion

The overall performance of a switching system is determined by the queueing management mechanisms. One of the main reasons for queueing is contention in the switch fabric. Contention occurs when there are multiple cells in different ingress line cards having the same destination, or multiple cells in the same ingress line card having separate destinations. In either case, the switch fabric arbitration mechanism decides which non-contending subset of the contending cells are accepted and are forwarded to their destination. Therefore, switch fabrics scheduling and arbitration solutions are major players in the queue management of the system, and determines the overall performance of the system.

In this paper, we discussed the major performance metrics that should be studied for a switch fabric. We also elaborate on the compatibility aspects and requirements for the switch fabric in a system. We introduced the concept of rate-aware switch fabric, as the capability to control, measure, monitor, and maintain the serving rate of all queues in the switch fabric. We argued that next generation switch fabrics should be rate-aware in order to meet the requirements of future systems and applications. We then go one step further and reviewed the architectures and scheduling technologies that are currently used for switch fabrics, and make the point that none of them can be a rate-aware switch fabric. We introduced Zagros Networks switch fabric and ZIBA, its rate-aware arbitration technology as the first rate-aware switch fabric. Finally, we presented some simulation results that illustrate superior performance of Zagros technology.

6.0 References

1. J.C.R. Bennett and H. Zhang, "Hierarchical Packet Fair Queueing Algorithms," IEEE/ACM Transactions on Networking, 5(5):675-689, October 1997.
2. K. Nell, "Switch Fabric Design Selection Analysis," Zagros Networks white paper series on switch fabrics, July 2002, http://www.zagrosnetworks.com/docs/WP_Switch_Fabric_Selection_July2002.pdf.
3. K. Sayrafian, "Overview of Switch Fabric Architectures," Zagros Networks white paper series on switch fabrics, July 2002, http://www.zagrosnetworks.com/docs/WP_Rate-aware_scheduling_primer_July2002.pdf.