Performance Evaluation of Router Switching Fabrics

Nian-Feng Tzeng and **Malcolm Mandviwalla**
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504, U. S. A.

Abstract

Switching fabrics with distributed control for scalable routers have been proposed recently [7]. Such a fabric consists of small routing units (RU's) interconnected by multistage-based connecting components (CC's) according to grid structures, thereby referred to as a grid-oriented, multistage-connected RU's, dubbed GMR, and is a direct interconnect with distributed routing. Performance of GMR is evaluated analytically and by simulation, with the packet mean latency being a key performance measure of interest. Under simplified assumptions and uniform traffic distributions, our analytic results are found to be very close to the simulation results (usually within 3% of each other) for a wide range of sizes, providing confidence to our simulation tool. Our simulation study demonstrates that GMR can deliver packets to their destination ports effectively in practical settings even when many ports run at a high-speed rate of 40 Gbps and non-uniform traffic exists. GMR is readily applicable to scalable routers with large numbers of high-speed ports.

1. Introduction

In light of explosive growth of the Internet and numerous enterprise networks in recent years, there is strong interest in the industry to design and build a new class of "super routers" able to offer access to hundreds (or even thousands) of ports on a single router. Such a router can increase its capacity as the need arises. Many routers commercially available cannot scale well since they employ switching fabrics like crossbars or shared busses to interconnect key components (such as LC's and routing engines). A next-generation must provide extensive scalability, able to connect a large number of ports, which are terminated at its LC's. Several terabit switching fabrics based on the crossbars with centralized control have been unveiled lately. They can scale up to size 32 only with the LC speeds elevated; e.g., LC's in the Yuni switch [8] have a top speed of 80 Gbps, which is achieved by aggregating multiple links. Hardware required for link aggregation to such a high data rate will be rather complex and expensive. Separately, a few scalable switching fabrics with direct interconnection styles have been developed [2, 6].

We have considered novel switching fabrics for scalable routers recently [7]. Such a fabric comprises small routing units (RU's) interconnected by connecting components (CC's) in accordance with grid structures, where a CC is composed of a multistage interconnect.

This work was support in part by the National Science Foundation under Grants EIA-9871315 and CCR-0105529.

For constructing practical sized fabrics, the 2-D grid structure is sufficient, with each RU connected to one or multiple copies of CC's along the *x*-direction and the *y*-direction. The proposed switching fabric, dubbed GMR (grid-oriented, multistage-connected RU's), routes packets (of fixed length) in a distributed manner and needs merely two types of chips for constructing any sized fabric. It follows direct interconnection style (i.e., a grid) to achieve great scalability while employing multistage networks to lower the hop count (and thus the mean latency) along each direction in the grid. GMR is shown to be cost-effective, ideally suitable for scalable routers [7].

2. GMR Switching Fabrics

A. Description of GMR

GMR is designed for high-performance routers operating at aggregate speeds of multi-terabits per second and connecting to large numbers of (external) ports. It includes two types of basic building blocks, one for providing routing decisions and connecting external links to the fabrics, called routing units (RU's), and the other for interconnecting RU's based on a grid style. A design of size 256 is depicted in Fig. 1, where an RU is a small crossbar for connecting two external links (through its external ports) plus up to three *x*-directional connecting components (CC's) and up to three *y*-directional CC's (through its internal ports), with a CC being a multistage interconnect, like the Omega network, composed of 4×4 crossbars. Such a CC is known to be scalable with reasonably low hardware complexity, and it routes packets in a distributed manner according to the destinations of the packets. RU's are interconnected by CC's following a 2-D grid fashion, with N_x and N_y RU's along the *x* and the *y* directions, respectively, for a fabric of size $2\cdot(N_x\times N_y) = N$. Here, we assume that N_x and N_y are multiples of 4, since CC's are composed of 4×4 crossbars. GMR takes advantage of the direct interconnection style to achieve high scalability while employing a multistage structure to connect all RU's along each *x* row (or *y* column) for reducing the hop count needed to travel along the *x* (or *y*) direction. It enjoys good scalability without requiring a linear traversal along each direction presented by a typical grid.

Routing Units

An RU connects two external links via its external ports to the switching fabric. A GMR with size *N* contains a total of (*N*/2) RU's, which are numbered from RU_0 to $RU_{(N/2)-1}$, as shown in Fig. 1. The external ports in those RU's are labeled from 0 to *N*-1. Given an external port (e.g., numbered *p,*

$0 \le p \le N-1$), the RU to which the port is connected can be derived directly by a shift register (to the right by one bit).

Upon admitting a packet, an RU produces the routing tag of the packet based on its destination port address, say p, as follows. Given a size $N = 2 \cdot (N_x \times N_y)$, the tag is a 2-tuple $<t_x, t_y>$, where t_x (or t_y) is the x (or y) coordinate of the RU to which the destination port is connected, namely, t_x equals the remainder of $(p/2)/N_x$ and t_y is the quotient of $(p/2)/N_x$. Computing t_x and t_y is done easily using a pair of shift registers, if N_x is a power of 2.

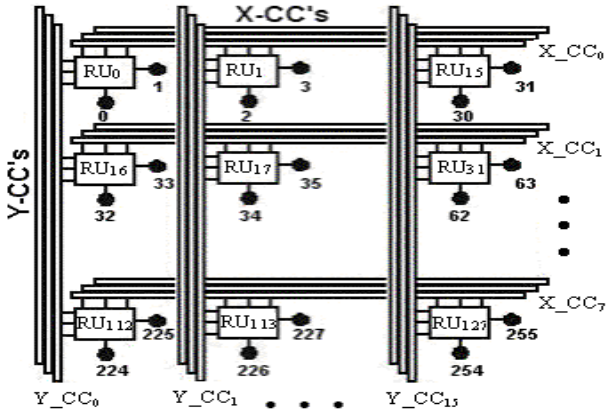

Figure 1. GMR of size 256 (i.e., $N_x \times N_y = 16 \times 8$, from [7]).

If a packet is destined for an external port, which is connected to the same RU where it arrives, the packet is routed to its destination by the RU directly without going through any CC. Otherwise, the packet is delivered to its destination by taking either an x-directional CC, a y-directional CC, or both. Note that packets arriving at RU's are of fixed sizes for delivery to their destinations. A packet takes at most one x-directional CC and one y-directional CC before reaching its destination. It is sent by the arrival RU to a CC, if needed, in one cycle and is moved in a cycle from one crossbar within a CC to another crossbar in the next stage. An RU handles two types of traffic, one is newly injected from its two external ports and the other is pass-through (transient) traffic, which was injected earlier at other RU's before being routed over an x-directional (or y-directional) CC to reach this RU. On receiving a packet, the RU checks its tag to see if the packet is destined for one of its two external ports (by simply comparing the packet tag with the contents of the RU's ID register pair). If the packet is not destined for the RU, it is routed forward over a CC according to the routing algorithm implemented in the RU.

Connecting Components

Each connecting component (CC), whether an x-directional or a y-directional one, is composed of identical 4×4 crossbars. Routing in a CC is distributed, on the basis of two bits of the x (or y) coordinate of the destination tag of a packet, if it is x-directional (or y-directional) as explained in [7]. A packet takes at most one x-directional CC and one y-directional CC before reaching its destined RU. The sequence of taking the two CC's, if needed, is decided by the routing algorithm implemented in the RU (at which the packet arrives). Note that an RU may connect up to three x-directional CC's (and also the same number of y-directional CC's), through its internal ports.

B. Routing in RU's

The routing algorithm implemented in each RU dictates the GMR behavior. It delivers packets to their destinations through *shortest paths* without causing any deadlock. As only shortest paths are considered, routing in the GMR fabrics is always *minimal*. It orders fabric resources (i.e., directions) and requires packets to use those resources in a strictly monotonic order, like the XY routing algorithm. A packet is sent along an x-directional CC first before a y-directional CC (when both directions have to be taken) to avoid deadlocks. Alternatively, a packet may be delivered along a y-directional CC first prior to an x-directional CC as well. In either case, an RU makes its routing decision independent of traffic over the two directions, referred to as deterministic routing. When the packet is to be sent over an x-directional (or y-directional) CC, the RU then selects an arbitrary copy (among those multiple CC's, if existing) for delivery [7]. Traffic is distributed over those multiple copies uniformly. This deterministic routing is called the *XY routing* algorithm, based on which our analysis will be conducted subsequently.

3. Performance Analysis

GMR under analysis is bidirectional, with one queue equipped at each output port of constituent RU's and CC's to hold multiple packets heading for the same output in a cycle. Like earlier work [1], this analysis assumes the traffic distribution to be *uniform* and the queue size to be *infinite*. Queues with infinite capacity do not drop any packet and are a good approximation for queues with finite length, since a queue of length no more than four behaves very closely to an infinite queue [3]. The performance measure of interest is the mean latency (in cycles), which reflects the average number of cycles for a packet (of fixed length) to travel from its arrival port to its departure port over the route dictated by XY routing. If a packet has to traverse one row (along the x direction) plus one column, it passes through the x output queue (namely, the queue at the output port connecting to an x directional CC) of the arrival RU and then an output queue of the CC, followed in sequence by an output queue of another RU and an output queue of the CC which connects to the destined RU (where its departure port lies). If a packet travels only along one row or column (exclusively) to reach its destination, the packet passes through an output queue of the arrival RU and then one output queue of the CC which connects to the destined RU. Our analysis is performed under simplified operational assumptions for various sizes N ($= 2 \cdot N_x \times N_y$, where N_x and N_y are the numbers of RU's in each row and column, respectively):

1. Packets arrive at input ports independently following Bernoulli trials with a specified arrival rate λ.
2. Each output queue at RU's and CC's can take multiple competing packets from different input ports during a single cycle, depending on the *speedup* of the queue.

3. Packets in an output queue have a deterministic service process with mean μ (i.e., an M/D/1 queue).
4. Each CC is modeled as a single stage crossbar.

The first two assumptions are commonly employed to derive the queue waiting time [1]. Our GMR has a range of structural parameters, including the number of CC copies along both the *x*-direction and the *y*-direction, speedups at RU queues and CC queues, and the ratio of the CC speed to the RU speed [7], the service rate at each output queue is no longer a constant of 1 and is determined by the structural parameters. Assumption 3 signifies that each output queue is an M/D/1 system, whose solution is known [5]. As each output queue at RU's and CC's takes packets from multiple input ports in one cycle, with each port possibly supplying multiple packets, the mean service rate μ should be no smaller than the rates from all the input ports combined. The last assumption renders internal contention in CC's to be ignored and the queuing effects are captured only at the last stage of CC's. (Note that when this last assumption is removed in realistic situations, as investigated using simulation in Section 4, the packet mean latency is always higher under any load since traversing CC's takes longer when they are in the multistage form than in the single crossbar form.) The queue speedup dictates how many competing packets in total can be transferred to the queue from different inputs in one cycle, with each input supplying up to μ packets. If the speedups chosen are large enough to let all competing packets accepted in every cycle, the arrival rates of all queues can be derived precisely according to the routing probabilities of traffic flows in RU's and CC's. In the subsequent analysis, the queue speedups are assumed *infinite*, leading to the lowest mean latency possible.

The mean latency of a packet in GMR can be obtained from the expected waiting time in each queue and the average number of queues along the route where a typical packet travels. Let $E[r]$ and $E[s]$ be the mean response time and the mean service time per packet, respectively, we have the expected waiting time given by

$$E[w] = E[r] - E[s]. \quad (1)$$

The expected response time $E[r]$ for an M/D/1 queue, according to [5], is expressed as $E[s] + \rho E[s]/\{2(1 - \rho)\}$, where ρ is traffic density, which equals $\lambda E[s]$. From Eq. (1), the expected waiting time for a queue becomes

$$E[w] = \lambda E[s]^2/\{2(1 - \lambda E[s])\}, \quad (2)$$

which is dictated by λ (the queue arrival rate) and the mean service time $E[s]$ (which equals $1/\mu$). As there are different types of output queues in GMR, we derive λ for those queues under a uniform traffic distribution below.

A. Derivation of λ

Packets are generated at incoming ports connected to RU's (see Fig. 1), and they travel along either one and only one direction or along both directions (i.e., *x*- followed by *y*-direction) based on XY routing, provided that their destination RU's are different from the arrival RU's. As packets are generated randomly with a uniform distribution, every packet has an equal probability to destine for any port (other than the arrival one) and the route to its destination is shortest and uniquely defined. Since all competing packets at any queue in GMR are accepted by the queue in one cycle (under the assumption of infinite queue speedups), our analysis considers GMR whose constituent RU's are connected to only single copies of CC's along either direction, as depicted in Fig. 2. As GMR is bidirectional, every link is represented by two unidirectional links in the figure. The RU output queue which leads to an *x*-directional CC (or a *y*-directional CC) is denoted by $Q_{RU,x}$ (or $Q_{RU,y}$). The mean arrival rate (λ) of $Q_{RU,x}$ comprises two components: $g^*p_{R,x}$ and $g^*p_{R,x}$, one contributed by each external port with the packet generation rate of g, where $p_{R,x}$ is the probability for a generated packet to head for the queue (and then terminate at a remote port, referring to as remote traffic), as shown in Fig. 2. Since packets are uniformly distributed, we have the expression of $p_{R,x} = (N - 2N_y)/(N - 1)$ for GMR with size N ($= 2 \cdot N_x \times N_y$). This makes λ of $Q_{RU,x}$ equal

$$2g^*(N - 2N_y)/(N - 1). \quad (3)$$

The mean arrival rate to $Q_{RU,y}$ has three components: $g^*p_{R,y}$, $g^*p_{R,y}$, and g_{x_CC}, where g_{x_CC} is the incoming rate from the output queue $Q_{x\_CC,I}$ in *x_CC* which connects $Q_{RU,y}$, as illustrated in Fig. 2. Each output port in *x_CC* is equipped with *two separate queues*, one (i.e., $Q_{x\_CC,I}$) for packets which have to pass through *y*_CC's before reaching their destination RU's (called an intermediate queue) and the other (i.e., $Q_{x\_CC,T}$) for packets which are to terminate at the RU (called a terminating queue). This is to prevent pass-through packets from being blocked by terminating packets and is easily realizable by checking a *y*-traversal (single-bit) indicator carried in the header of a packet; the indicator is set at the arrival port when a *y*_CC traversal is needed.

In a state of equilibrium, the departure rate and the arrival rate of $Q_{RU,x}$ are identical. The departure rate of $Q_{RU,x}$ is thus given by Eq. (3), which is the rate for packets to arrive at each *x_CC* incoming port (under the uniform and independent packet generation assumptions). The rate of packets routed to a given output port is equal to that expressed by Eq. (3), because (i) packets arriving at an *x_CC* via any given port are equally distributed to all but one (i.e., N_x - 1) output ports (note that *x_CC* has N_x bidirectional links each associated with a pair of input and output ports, as it is assumed to be a single crossbar in this analysis) and (ii) a given output port in *x_CC* receives arrival rate contribution from N_x - 1 (i.e., all but one) input ports. Let p_I be the probability that a request routed to a given output port is put in $Q_{x\_CC,I}$, then p_I can be obtained by considering N random packets, one generated by each external port. Among those N packets, $N - 2N_y$ of them must pass through *x_CC*'s to reach their destinations, resulting in $p_I = (N - 2N_y - 2(N_x - 1))/(N - 2N_y)$, where the numerator excludes those packets whose destinations are on the same row as their originators.

At equilibrium, $g_{x\_CC,I}$ equals the arrival rate of output queue $Q_{x\_CC,I}$ in *x_CC*, which amounts to Eq. (3) multiplied by p_I, namely,

$$2g(N - 2N_y - 2(N_x - 1))/(N - 1). \quad (4)$$

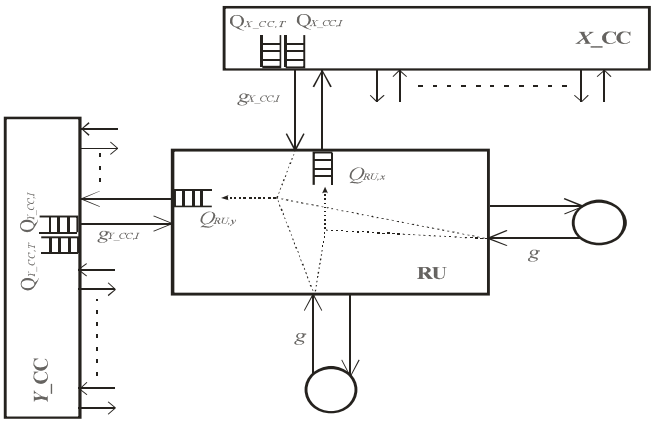

Figure 2. Waiting time derivation for all output queues.

As stated earlier, the arrival rate of $Q_{RU,y}$ equals $g^*p_{R,y} + g^*p_{R,y} + g_{x\_CC}$, which is given by

$$2g(p_{R,y} + (N - 2N_y - 2N_x + 2)/(N - 1)), \quad (5)$$

where $p_{R,y}$ denotes the probability of a generated packet contributing to remote traffic and passing through $Q_{RU,y}$, and it equals $2(N_y - 1)/(N - 1)$. Unlike $Q_{x\_CC,I}$ in *x_CC*, output queue $Q_{y\_CC,I}$ in *y_CC* has a nil arrival rate since no packets following XY routing are routed over *y_CC* and then over *x_CC*. It should be noted that for other routing algorithms [7], the arrival rate to $Q_{y\_CC,I}$ is not zero in general and can be derived accordingly.

B. Mean Latency

The mean latency in GMR can be obtained by summing up latencies experienced by the packet at all queues along the route taken. If a packet is destined for the port connected to the same RU as its originator (with probability $p_0 = 1/(N-1)$), the packet always takes one cycle to reach its destination; otherwise, it may require one- or two-dimensional correction before reaching its destination. The probability of one-dimensional correction equals $p_1 = 2(N_x+N_y-2)/(N-1)$, and that of two-dimensional correction is $p_2 = 1-(p_0+p_1)$. For a packet with one-dimensional correction, its route includes either (i) a $Q_{RU,x}$ plus a $Q_{x\_CC,T}$ or (ii) a $Q_{RU,y}$ plus a $Q_{y\_CC,T}$. The probability of route case (i) is $(N_x-1)/(N_x+N_y-2)$, whereas that of route case (ii) is $(N_y-1)/(N_x+N_y-2)$. The latency for a packet to pass through a $Q_{RU,x}$ (or $Q_{RU,y}$) equals 1 plus its waiting time, which according to Eq. (2), depends on the arrival rate, given by Eq. (3) (or Eq. (5)), and the mean service time, $E[s]$ (= $1/\mu$). Our analysis assumes the service rates of $Q_{x\_CC,T}$ in *x_CC* and of $Q_{y\_CC,T}$ in *y_CC* to be large enough so that when packets are put on those output queues, they are *all* moved to their respective destinations in the subsequent cycle. This assumption renders the latency to pass through a $Q_{x\_CC,T}$ (or $Q_{y\_CC,T}$) equal to 1 always. The mean latency of one-dimensional correction, L_1, is thus derived. For a packet with two-dimensional correction, its route contains a $Q_{RU,x}$, a $Q_{x\_CC,I}$, a $Q_{RU,y}$ and a $Q_{y\_CC,T}$. Like a $Q_{RU,x}$ or $Q_{RU,y}$, $Q_{x\_CC,I}$ exhibits the mean service time of $E[s]$. The latency of traversing a $Q_{x\_CC,I}$ amounts to 1 plus its waiting time, which is governed by Eq. (4) and $E[s]$. The mean latency of two-dimensional correction, L_2, equals the sum of latencies over those four queues. From L_1 and L_2, the packet mean latency is expressed as

$$p_0 + p_1*L_1 + p_2*L_2. \quad (6)$$

C. Analytic Results and Comparison

Analytic mean latency results as a function of offered load (in packet/cycle per port) for GMR with different sizes are illustrated in Fig. 3. RU's and CC's are assumed to operate at the same clock rate, given that a CC is considered to be a single crossbar in our analysis (rather than a multistage network composed of smaller building blocks as depicted in [7]). Each visit to an RU or a CC thus takes *one cycle* plus its waiting time in a corresponding output queue, as analyzed in the preceding subsection. As can be seen in the figure, when the offered load goes beyond 0.6, waiting times start to grow noticeably, despite that every output queue can accept all competing packets from every input in RU's or CC's during each cycle and that the queue capacity is unbounded. The waiting times are due solely to a bounded mean service rate (μ) of 2.5 or 3.0 (note that $\mu = 2$ gives rise to exceedingly large waiting times when the offered load is close to 1.0 because the arrival rate of $Q_{RU,x}$ then approaches 2.0). To validate the correctness of our analytic model, a simulator has been developed to evaluate the mean latency of GMR under the same set of (simplified) assumptions.

The simulation results have been gathered and compared with the analytic ones for various sizes and average service rates ($\mu > 2$). While the gaps between the analytic and simulation results expand as the offered load grows for a given size and mean service rate, they are found to be within 3% for all the examined sizes (up to 1024) and service rates. In Fig. 3, the simulation results (shown by dashed curves) are illustrated for $N = 32$ and 256 and for $\mu = 2.5$ and 3.0, and they all are seen to be no more than 2% away from their corresponding analytic results. This indicates that not only our analytic expressions for mean latency are correct but also our simulator produces results of high confidence. In the next section, performance of GMR gathered using our simulator under realistic assumptions with different structural parameters is presented.

4. Performance under Realistic Assumptions

GMR under realistic assumptions, unlike simplified ones above, has limited speedups for queues in RU's and in CC's, a bounded queue capacity (of a small constant), and multiple CC's connected to an RU along either direction. A CC is composed of small 4×4 crossbars [7], which are simpler than an RU and operate at a higher clock rate. Let the ratio of the CC speed to the RU speed be represented by γ. Each packet can be moved from an RU to a 4×4 crossbar within a CC in one cycle, while it can be moved from one 4×4 crossbar to another within a CC in $1/\gamma$ cycle.

The performance measures of interest are offered load and mean latency, where the former indicates the number of packets delivered to GMR from each external input per cycle, and the latter reflects the average time for a packet to travel from its arrival (external) port to its outgoing port.

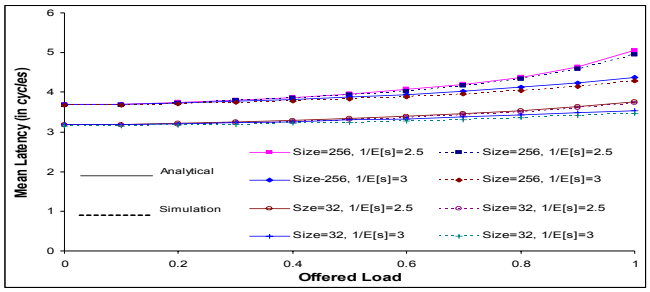

Figure 3. Mean latency versus offered load.

Each data value presented is the result after 100,000 simulation clock cycles. This number of cycles has been observed to produce stable results for all evaluated cases. In our simulation, each external port generates a fixed-length packet following an independent Bernoulli process with various generation rates (in packet/cycle). The packets are assumed to be uniformly distributed, and queues are introduced only to the output ports of RU's and CC's (i.e., output queuing design), with the queue capacity fixed to 4.

A. Uniform Traffic

We have simulated GMR extensively to gather its performance outcomes under XY routing over a wide range of structural parameters, including the size (N), the speedup of RU's (ζ_{RU}), the speedup of constituent crossbars in CC's (ζ_{CC}), the speed ratio (γ), and the number of CC's connected to an RU along each direction (Γ). Each GMR structure is simulated for sizes up to 1024 and the packet generation rate ranging from 0 to 1.0. Here, we demonstrate the simulation results only for the case of $\zeta_{RU} = 3$, $\zeta_{CC} = 2$, $\gamma = 3$, and $\Gamma = 3$ in Fig. 4, where the mean latency is given in cycles. As can be found in this figure, the mean latency for $N = 32$ remains essentially flat for the whole range of offered load, indicating that GMR under the given set of structural parameters experiences a negligible waiting time even under extremely heavy load. This fact holds for other cases where the parameter values (ζ_{RU}, ζ_{CC}, γ, Γ) are smaller as well, according to our simulation. GMR with $N = 256$ exhibits relatively larger growth in mean latency for offered load exceeding 0.9; its mean latency climbs beyond 5 cycles when the offered load approaches 1.0. For light offered load, mean latency reflects the average number of queues visited by a typical packet on the way to its destination, and packets in a larger sized GMR naturally require to traverse more queues before reaching their destinations.

When compared with the corresponding curves (for N = 32 and 256) depicted in Fig. 3, the mean latency results under realistic assumptions are always larger for any offered load when N is 256, while they are roughly the same as those under $\mu = 3.0$ for $N = 32$. This is because the simplified assumptions consider CC's to be single crossbars (rather than multistage interconnects comprising 4×4 building blocks); for $N = 32$, a CC happens to be one building block but a CC in GMR with $N = 256$ is a two-stage interconnect [7], which introduces extra latency (a fraction of one cycle determined by γ). The curve for $N = 256$ in Fig. 4 seems to better match the trend of the corresponding curve with μ = 2.5 shown in Fig. 3, implying that the average service rate under this set of structural parameters is closer to 2.5. Under the same set of parameters, GMR with $N = 32$ presents an average service rate μ approximating 3.0. Packets in GMR with $N = 1024$ start to encounter sizable waiting times when offered load is more than 0.8. Unlike its smaller sized counterparts, this GMR fails to reach the maximum offered load of 1.0, signifying that certain packets generated at a rate of 1.0 are dropped. Fortunately, not all input ports to switching fabrics of routers in practical settings are likely to have generation rates of 1.0, and GMR performance under those settings is illustrated next.

Routers employ switching fabrics to interconnect key components, like routing engines and line cards. The highest link speed for commercially available routers is 10 Gbps and will soon be elevated to 40 Gbps. Consider IP traffic, which accounts for the vast majority of current Internet traffic and whose packet length ranges from 40 bytes up to 64K bytes. Packets are fragmented into data units of fixed length in switching fabrics for delivery. The clock rate is chosen such that one data unit can be moved from its arrival port to an RU output queue (or from an RU queue to a CC queue) in one cycle time. Let the data unit selected for GMR be **40 bytes** and the clock time be **10 *ns*** (i.e., operating at 100 MHz like the core of the Spider chip [4]). In practice, the data rates of external router links very, depending upon the port interfaces employed. Let's consider GMR with links of 10 Gbps and 40 Gbps only (since the cases with lower link rates are easier to be accommodated). Mean latency versus ψ for GMR with $N = 256$ and under the same structural parameters as listed in Fig. 4 is shown (by the dotted curve) in Fig. 5, where ψ denotes the percentage of links being 40 Gbps. The average input rate (in terms of data unit/cycle) across the entire system is about 0.3 (= 10*10/(40*8)) for ψ = 0 and close to 1 for ψ = 70%. It is clear that GMR can deliver packets to their destinations swiftly without incurring long latencies under uniform traffic distributions even when most ports are with the rate of 40 Gbps and all ports are transferring packets at their full data rates consistently.

B. Non-Uniform Traffic

Packets at a router decide their outgoing ports by looking up the routing table contained in the router, with packet destinations used as keys for table lookups. A typical router often designates one or a few ports as the *default ports*, which are meant to take those packets whose destinations cannot match any table entries. In other words, a packet which fails in the table lookup is forwarded to a default port (which often connects to a more powerful router with a larger and more complete routing table). This implies that traffic in switching fabrics is often non-uniform, with one or a few ports acting as the *hot* port(s) which receive an un-proportionally big traffic rate. GMR under non-uniform traffic distributions is evaluated using simulation, with traffic distributions characterized by the hot spot rate η ($0 \le \eta \le 100$). Given η for GMR with size N, it means that η% traffic from every input port is destined for the specific hot port(s), with the rest evenly distributed to all N-1 output ports.

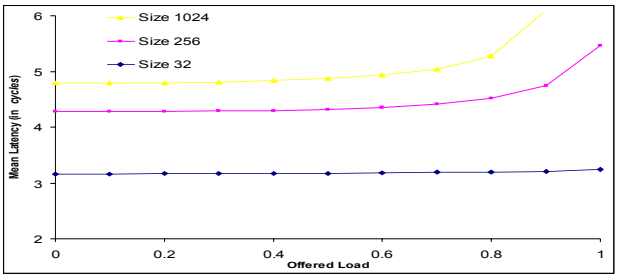

Figure 4. Mean latency versus offered load for $\zeta_{CC} = 2$ and $\zeta_{RU} = \gamma = \Gamma = 3$ under uniform traffic.

In the case of one single hot port, the performance result for a given N is found to be independent of the hot port position, whether it is on a corner or at the middle of the grid. The results under a single hot port with η = 1%, 5%, 10% are shown (by dashed curves) in Fig. 5, where η = 1% is likely to reflect a backbone router (whose routing table is often large enough to yield matching for almost all packet destinations) and η = 10% may better capture the behavior of an edge router. A backbone router is expected to have more high-speed connections (of 40 Gbps) whereas an edge router normally possesses few such connections, if any at all. For a backbone router with η = 1%, GMR behaves similarly to the uniform traffic situation, permitting the majority (more than 70%) of the ports to be at 40 Gbps. For η = 10%, however, only a very small fraction (say, $\leq$ 3%) of ports may operate at the high-speed rate with the remaining ones at 10 Gbps (or lower), but this is expected to fit the need of edge routers. Our GMR thus offers good performance for all types of routers in practice, be backbone or edge ones. If multiple hot ports (which together take η% total traffic additionally) exist, the behavior of GMR changes. GMR performance is found insensitive to the positions of the hot ports, and we include in Fig. 5 only the results of two hot ports which are located at the opposite corners of GMR. When η is 5% or more, two hot ports tend to support a bigger percentage of 40-Gbps ports than their one hot-port counterparts, because they delay *traffic saturation* at hot ports until a larger ψ. Once saturation is developed, however, the case of multiple hot ports exhibits rapid growth in mean latency as those multiple saturation "regions" (one around each hot port) seem to exuberate the degree of contention. For example, the saturation point for η = 10% is pushed beyond ψ=30%, in comparison to ψ=2% (or so) for the case with a single hot port. A similar phenomenon is observed for η = 5% illustrated in the figure, where the saturation point for two hot ports is seen at ψ=50%, as opposed to ψ=30% for one hot port. In a backbone router application (with η = 1% or less), GMR can handle an identical number of 40-Gbps ports with a single hot port as with two hot ports while maintaining a performance level close to what uniform traffic presents.

5. Conclusion

This paper has evaluated switching fabrics proposed recently for scalable routers [7], analytically and by simulation. GMR is analyzed to arrive at a key performance measure, the mean latency. The analytic results are validated using our simulation under the same assumptions, and they are observed to be usually within 3% of simulated outcomes for the full range of input load and for various sizes (up to 1024), suggesting that our simulator provides results of high confidence. The simulator is then employed to evaluate GMR under realistic assumptions and with varying structural parameters (such as the speedups of RU and CC queues, the CC to RU clock rate ratio, and the number of CC copies, which dictate GMR performance). GMR employed in practical settings where the external links may carry different data rates (ranging typically from 100 Mbps up to 40 Gbps) and traffic could be non-uniformly distributed, is also evaluated by simulation, with the results illustrated and discussed. For N = 256, GMR is shown to permit many ports operating at 40 Gbps, depending on the degree of non-uniformity (which tends to be lower for backbone router applications). Under non-uniform traffic, GMR is possible to offer a performance level close to what is delivered under its uniform traffic counterpart. As it requires only two types of chips for any sized construction with a low cost [7] and can handle uniform and non-uniform traffic well with good scalability, GMR is ideally suitable for future routers with large numbers of ports operating at various data rates.

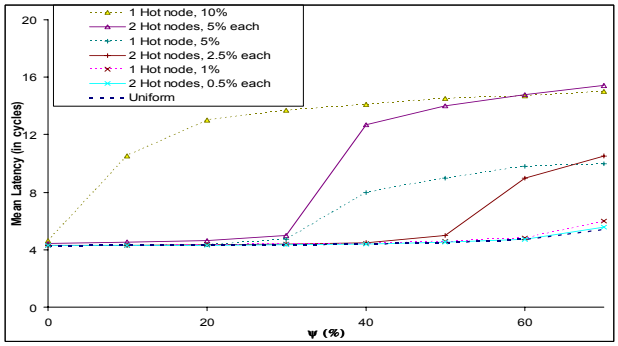

Figure 5. Mean latency versus ψ for GMR under non-uniform traffic. ($N = 256$, $\zeta_{CC} = 2$, $\zeta_{RU} = \gamma = \Gamma = 3$)

6. References

[1] J. Anderson and S. Abraham, "Performance-Based Constraints for Multidimensional Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 11, pp. 21-35, Jan. 2000.

[2] W. Dally, "Scalable Switching Fabrics for Internet Routers," White Paper for Avici Systems Inc., URL – http://www.avici.com/technology/whitepapers/TSRfabric-WhitePaper.pdf, 2002.

[3] D. Dias and J. Jump, "Packet Switching Interconnection Networks for Modular Systems," *Computers*, vol. 14, pp. 43-53, Dec. 1981.

[4] M. Galles, "Spider: A High-Speed Network Interconnect," *IEEE Micro*, vol. 17, pp. 34-39, Jan./Feb. 1997.

[5] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, New York, 1991, Ch. 31.5, pp. 540-544.

[6] Pluris, Inc., "Tech Briefs: Multistage Routing," URL – http://www.pluris.com/terabit/ techbriefs/, 2001.

[7] N. Tzeng and M. Mandviwalla, "Cost-Effective Switching Fabrics with Distributed Control for Scalable Routers," *Proc. 22nd IEEE Int'l Conf. on Distributed Computing Systems*, July 2002, pp.65-73.

[8] K. Yun, "A Terabit Multi-Service Switch," *IEEE Micro*, vol. 21, pp. 58-70, Jan./Feb. 2001.