

Optics inside routers

Nick McKeown

Computer Systems Laboratory, Stanford University, CA 94305-9030, USA.

Email: nickm@stanford.edu

Invited Paper

Abstract *In this paper we describe some recent developments in how optics can be used inside Internet routers to scale capacity and reduce power. At Stanford University we are currently designing a 100Tb/s Internet router with an optical switch fabric that guarantees 100% throughput for all traffic.*

Background

It has been speculated for some time that electronic packet-switched Internet routers will eventually be replaced by all-optical routers that process, buffer, route, and switch packets in the optical domain. There are a number of reasons why this won't happen anytime soon. First and foremost, an Internet router requires large random access buffers to hold packets during times of congestion. A widely used rule-of-thumb is that a router needs enough buffers to hold the data sent in an average round-trip-time (100-300ms).¹ When added to all the other features a router performs, a typical 10Gb/s linecard has 30 million gates, 300 Mbytes of packet buffers, stores a million addresses, and consumes 200W. Unless features are radically pruned, optical linecards won't be feasible any time soon.

But optical components are already being used inside routers - to interconnect a router's subsystems. To reduce power-density, there is a trend towards multi-rack routers [3,2,5], with typically 16-32 linecards sharing a rack, and connected together by multimode parallel optical links and a central switch fabric. This has the advantage of reducing power density by spreading the router over multiple racks, but the total power is *increased* because of the increased number of conversions between the optical and electrical domains. The typical processing sequence is that arriving packets are processed by the ingress linecard, and buffered until a central scheduler grants them access to the switch fabric; at which point they are converted into the optical domain, transmitted to the switch fabric, converted back into the electrical domain, switched by electronic crossbars, converted into the optical domain, transmitted to the egress linecard, converted back into the electrical domain, and then processed and scheduled for departure.

An obvious question to ask is: Can the number of conversions - and the total system power - be reduced by replacing the electronic switch fabric with

an optical switch fabric?

The conventional wisdom is that this probably doesn't make sense. The argument is two-fold and goes something like this: The performance of a crossbar switch is dictated by an electronic scheduler that picks a new permutation for each packet transfer; network operators would like schedulers that guarantee 100% throughput so as to maximize the utilization of their expensive long-haul links; but even heuristic schedulers are so complicated that they limit the router capacity (without guaranteeing good throughput). So why replace an electronic crossbar with an optical switch if its performance is limited by an inherently electronic scheduler? Second, the power of the system is dominated by the linecards, not the switch fabric. So why bother reducing the power of the switch fabric?

While these arguments have merit, the argument has been changed by the recent introduction of the *load-balanced switch* [8], which makes it possible to guarantee 100% throughput without a scheduler. In [11] we show how a 100Tb/s Internet router can be built from an optical load-balanced switch. The power of the switch fabric is so small that the entire 100Tb/s fabric can be placed in a single rack; compare this with the state-of-the-art (and power-limited) 2.5Tb/s electronic switch fabric described in [1].

The Load-Balanced Router Architecture

The basic load-balanced switch was first described by Chang in [8]. In [11] we showed how an optical load-balanced switch can be built from two uniform meshes of optical links (see

¹ Although rarely justified, this rule-of-thumb comes from studies of the particular congestion control algorithms used in TCP.

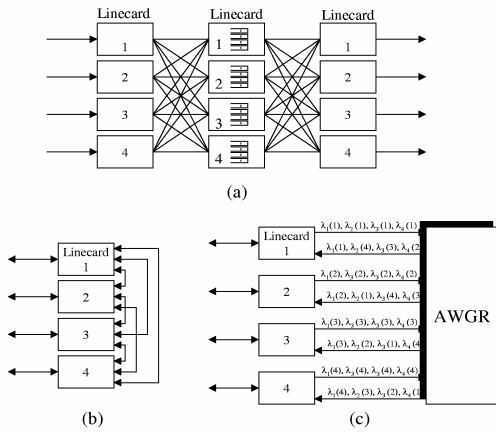


Figure 1a); or from a single mesh of N^2 links (

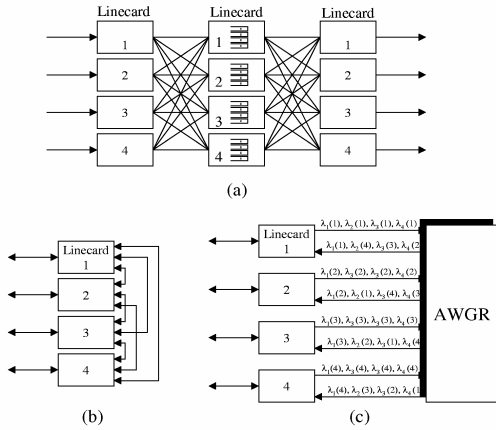


Figure 1b); or even from N links, an AWGR and N^2 WDM channels (

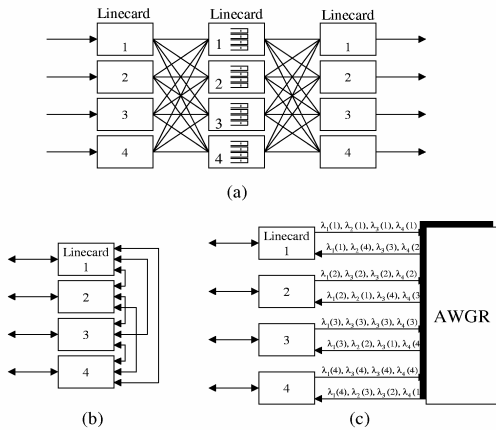


Figure 1c). Like most routers, the load-balanced architecture consists of a number of linecards connected by a switch fabric. But – as we will describe below – it differs in the path taken by packets, and by the operation of the switch fabric.

The router has N linecards operating at rate R . The switch fabric is a fully connected uniform mesh of DWDM channels that connect every pair of linecards at rate $2R/N$.

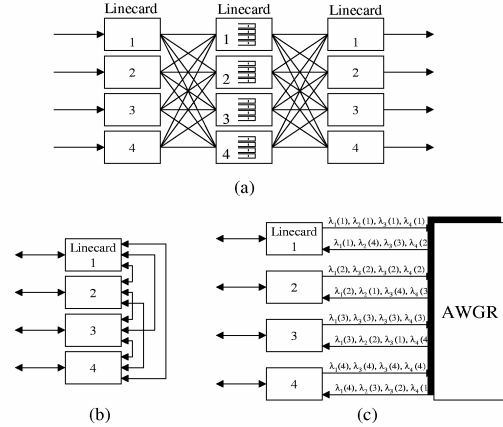


Figure 1 A load-balanced switch can be implemented by a uniform mesh. In (a) the logical two stages of the switch fabric are built from a uniform mesh of N^2 links operating at rate R/N , while in (b) the two meshes are replaced by a single fixed rate mesh with links operating at rate $2R/N$. In (c) the mesh has just N links carrying N^2 WDM channels interconnected by an AWGR.

It's quite different from a normal single-stage packet switch; instead of picking a switch configuration based on the occupancy of the queues, the mesh of channels is fixed, and independent of the state of the queues. When a packet arrives, the ingress linecard sends it to an intermediate linecard, which then sends it to the correct destination, i.e. every packet traverses the mesh twice, and the mesh's aggregate capacity is $N^2 \times (2R/N) = 2NR$. Arriving packets are spread uniformly over all the other linecards, regardless of their destination. The first packet is sent to linecard 1, the second to linecard 2, and so on. When a packet reaches its intermediate linecard, it is buffered in a virtual output queue (VOQ). The VOQ is served at rate R/N , regardless of its occupancy, and the packet is eventually delivered to the correct output linecard.

Although each packet passes through three linecards (ingress, intermediate, and egress), it is buffered only once in the intermediate linecard, making it an example of a Single Buffered Router [10]. Perhaps the most surprising aspect of the architecture is that although it is a packet-switch, and can support an arbitrary traffic matrix, there is no switch to be reconfigured; in fact, at first glance, it's not clear where the packet-switching actually takes place. The only place where a decision is made is in the intermediate linecard when it decides which VOQ to

put the packet in.

The intuition behind the architecture is that the first stage uniformly spreads packets over the intermediate linecards. This turns a non-uniform traffic matrix into a uniform matrix of arrivals to the second stage (first observed by Valiant in [9]). Uniform arrivals are much easier to schedule; so easy that if each VOQ is served at uniform rate R/N , then it leads to 100% throughput. This lead to the surprising result in [8] that the switch has 100% throughput for any stochastic, weakly mixing pattern of arrivals.

The basic architecture has some practical problems, such as: (1) Packets are mis-sequenced, (2) There are pathological deterministic arrival patterns that can reduce the router's capacity from NR to just R , and (3) Because the architecture relies on spreading traffic over all the linecards, it doesn't work when some linecards are missing. In [11] we solved these problems using a packet-scheduling algorithm called FOFF that prevents packet mis-sequencing, eliminates the pathological traffic patterns and gives 100% throughput for any pattern of arrivals. In fact, with FOFF, the expected packet delay is always within a fixed bound of an ideal output queued switch. We also described two architectures that support any number of linecards connected to arbitrary ports. Our hybrid electro-optical architecture is illustrated in Figure 2; its operation is described in detail in [11], and an algorithm for configuring the MEMS devices described in [3].

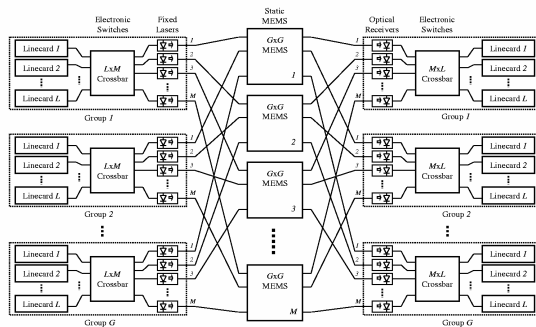


Figure 2 Hybrid electro-optical load-balanced switch. MEMS switches change configuration only when linecards are added and removed.

Stanford 100Tb/s Router Project

At Stanford University we have a collaborative project to design a 100Tb/s Internet router that conforms to Internet RFC 1812. The work is a collaboration between eight PhD students in the research groups of Professors David Miller, Olav Solgaard, Mark Horowitz and Nick McKeown. A capacity of 100Tb/s was picked because it is challenging but not impossible, is about two orders of magnitude beyond

what is commercially available today, and at current growth rates is about the size of routers expected to be available within eight years.

So far, the architectural work is complete; and our baseline design is the hybrid architecture show in Figure 2. The router is arranged as $G=40$ groups (racks) of $R=160Gb/s$ linecards, with $L=16$ linecards per rack. The racks are interconnected by $L+G-1=55$ MEMS switches that are reconfigured only when a linecard is added or removed. Essentially, the MEMS switches (together with the electronic crossbars in each rack) rearrange an arbitrary set of linecards so as to form a fully connected uniform mesh.

Designing this 100Tb/s creates some interesting research challenges. For example, how to design 160Gb/s electronic linecards to process and buffer packets. Our solution to building 160Gb/s packet buffers is described in [3]. Perhaps the most challenging part of the system is to build the electronic crossbar switches that connect the linecards inside a rack. The crossbars follow a fixed sequence of permutations so as to balance traffic across the MEMS switches. The capacity of one rack (2.5Tb/s) is too high for a single crossbar chip, so it needs to be decomposed into, say 32, bit-slices. Each bit-slice would connect via optical fibre to all 55 MEMS switches, and the data from all 32 bit-slices to one MEMS switch is multiplexed onto a single fibre. If the electronic crossbar slices use external optical modules, the size, power and cost per rack would probably be prohibitive. So instead, we are experimenting with arrays of optical modulators directly attached to the crossbar chip with a single external optical power source for all 32.

Conclusions

Today, most high-performance Internet routers use switches that reconfigure every packet transfer, and are limited by the scheduler that picks the configuration. We are finding that the load-balanced switch is a promising way to increase scalability by eliminating the need for a scheduler and enabling the use of a fixed mesh of wavelengths. In this architecture, the main source of complexity comes from the need to add and remove linecards at arbitrary locations. While our initial electro-optical solution appears feasible, we expect that future solutions, based perhaps on tuneable lasers and filters [11], will become cost-effective.

References

- 1 N. McKeown, C. Calamvokis, S.-T. Chuang, "A 2.5Tb/s LCS switch core," Hot Chips XIII, Aug. 2001.
- 2 Alcatel, "Alcatel's packet-based digital cross-

- connect switch application," July 2002, available at <http://www.alcatel.com>.
- 3 Juniper Networks, "The essential core: Juniper Networks T640 Internet routing node with matrix technology," April 2002, available at <http://www.juniper.net/solutions/literature/solutionbriefs/351006.pdf>.
 - 4 W. J. Dally, "Architecture of the Avici terabit switch router," Proc. Hot Interconnects XIII, Aug. 1998.
 - 5 Chiaro Networks. <http://www.chiaro.com>, May 2003.
 - 6 N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," IEEE Trans. on Communications, Vol.47, No.8, Aug. 1999.
 - 7 J.G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," Proc. of the IEEE INFOCOM, Vol. 2, pp. 556-564, Tel Aviv, Israel, March 2000.
 - 8 C.-S. Chang, D.-S. Lee and Y.-S. Jou, "Load balanced Birkhoff-von Neumann switches, Part I: one-stage buffering," Computer Communications, Vol. 25, pp. 611-622, 2002.
 - 9 L.G. Valiant and G.J. Brebner, "Universal schemes for parallel communication," Proc. of the 13th ACM Symposium on Theory of Computation, pp. 263-277, 1981.
 - 10 S. Iyer, R. Zhang, N. McKeown, "Routers with a single-stage of buffering," Proc. of ACM Sigcomm, Philadelphia, USA, Aug 2002.
 - 11 I. Keslassy, S-T. Chuang, K. Su, M. Horowitz, D. Miller, O. Solgaard, N. McKeown, "Scaling Routers Using Optics," Proc. of ACM Sigcomm, Karlsruhe, Germany, Aug 2003. Available at <http://www.stanford.edu/~nickm/papers>.
 - 12 I. Keslassy, S-T. Chuang, and N. McKeown, "Architectures and algorithms for a load-balanced switch," Stanford University HPNG Technical Report -TR03-HPNG-061501}, Stanford, CA, June 2003.
 - 13 S. Iyer, R. R. Kompella, and N. McKeown, "Designing buffers for router line cards," Stanford University HPNG Technical Report - TR02-HPNG-031001, Stanford, CA, Mar. 2002.