RC21602 (97408) November 18, 1999
Engineering and Technology

IBM Research Report

An Optical IP Switch

Rick Boivie
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

Research Division
Almaden – Austin – Beijing – Brazil – Cambridge – Dublin – Haifa – India – Kenya – Melbourne – T.J. Watson – Tokyo – Zurich

An Optical IP Switch

Rick Boivie

IBM T. J. Watson Research Center
Hawthorne, New York

Abstract
This paper discusses the design of an "Optical IP Switch". The Optical IP Switch combines Multiprotocol Label Switching[1][2][3] with optical switch technology in an innovative design that allows the Optical IP Switch to switch traffic in an IP network without processing the individual IP packets in "electronics". By eliminating the electronics from the data path through the switch, the Optical IP Switch can handle very high traffic volumes. This is likely to become important in the future since Internet traffic is expected to continue grow at a high rate for the foreseeable future.

Introduction

The 1990's have been a time of explosive growth for the Internet and by all estimates traffic on the Internet will continue to grow rapidly for some years to come. The US Commerce Department says that traffic on the Internet is doubling every 100 days [4] and John Sidgemore, CEO of UUnet (the largest US Internet Service Provider) predicts that this doubling of traffic every 100 days will continue at least through the year 2003[5][1].

This exponential growth will, if current trends continue, make it difficult for silicon-based routers and switches to keep up with the traffic. Note that Moore's law says that silicon processing power doubles in approximately 18 months but Internet traffic doubles in a little over 3 months[2].

One way to get around this "electronic bottleneck" is to eliminate the electronics from the data path through the router. The Optical IP Switch described in this paper eliminates this "electronic bottleneck" through a combination of Multiprotocol Label Switching and optical switching techniques.

Multiprotocol Label Switching or MPLS [1][2][3] is an IETF effort that combines "label swapping" based forwarding (e.g. the label swapping used in ATM[7][8] switches) with network layer routing. MPLS can improve the performance of a router by eliminating the IP processing on the data path through the router. This process is described in detail in references 2 and 6 but the basic idea is to setup switched paths through a network that correspond to the paths computed by

---

[1] Note that this high growth rate is not a recent development but has been a long-term trend. When we (IBM, Merit & MCI) built the NSFnet backbone (which was the principal backbone of the Internet between 1988 and 1995), traffic on the backbone increased by a factor of 10 in the first year of operation (from approximately 100 million packets per month in June of 1988 to about 1.05 billion packets per month in June of 1989).

[2] Note that the capacity of optical fiber is not a problem and won't be for some time. In his book, *Fiber Optic Networks* (Prentice Hall, 1993), Paul Green points out that there is 75,000 Ghz of useful capacity in a single strand of optical fiber.

the routing protocols so that IP processing can be avoided on most of the routers on the paths through the network .

The figures below give a simple example.

Figure 1 shows a pair of adjacent "Label Switch Routers".

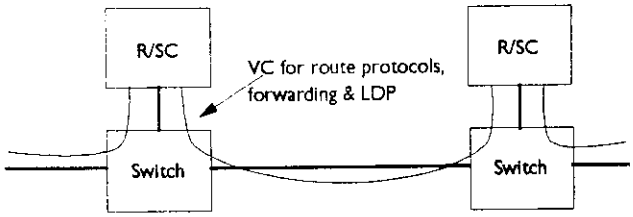

Figure 1

A "Label Switch Router" (or LSR) consists of a label swapping switch (such as an ATM switch) and a "Router/Switch Controller". The Router/Switch Controller (or RSC) exchanges IP routing information with other routers (some of which may be "Label Switch Routers") via the standard routing protocols (e.g. OSPF, BGP) and forwards IP packets based upon the routing information that it acquires. The RSC also *controls* its associated label swapping switch and sets up switched connections through the switch. The purpose of these switched connections and the mechanisms for setting up these switched connections will be described below.

The links between the adjacent LSR's in Figure 1 are ATM links and the traffic on these links is carried in ATM cells. An ATM cell contains an ATM header which contains some information that "labels" the cell as belonging to a particular virtual circuit (or VC). ATM switches use this label information in deciding how a cell should be routed[3]. (See references 7 & 8 for a more complete description of the ATM architecture.)

In an MPLS network, one of the VC's, the "default VC", is used for exchanging routing information and for forwarding IP datagrams between adjacent LSR's as shown in Figure 1. LSR's also use this default VC for a Label Distribution Protocol (or LDP) which is used to associate labels (ie VC's) with route table entries. In figure 2, for example, each of the LSR's uses the LDP to inform a neighbor of the association that it would like to establish between a VC and a route table entry, in this case network 9.0.0.0 (i.e. IBM).

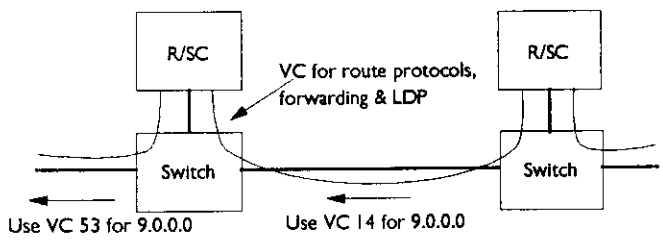

Figure 2

If an LSR receives an IP datagram on the default VC, the switch will route the individual cells to the RSC where they will be assembled and processed. The processing on the RSC includes the usual IP header level processing that routers typically do including checking the IP header checksum, processing any IP options, decrementing the value of the TTL field in the IP header (and discarding

[3] The labels used in ATM switches are "small" integers which lend themselves to fast routing.

the packet if the resulting value of the TTL field is 0), recomputing the header checksum, determining the next hop in the IP forwarding table, breaking the IP datagram up into ATM cells and sending those cells out on the appropriate link.

On the other hand, much of this processing can be avoided if adjacent LSR's have established appropriate mappings between VC's and route table entries. For example, if adjacent LSR's have agreed on the mappings shown in figure 2 and if the righthand LSR in the figure is the nexthop router for the lefthand LSR for traffic going to network 9.0.0.0, then the RSC in the lefthand LSR can instruct its switch to "splice" VC 53 from its lefthand link to VC 14 on its righthand link. As a result, any traffic that arrives on VC 53 (i.e. traffic addressed to network 9.0.0.0) will go out on VC 14 without any of the RSC processing described in the previous paragraph. Of course in a real network, this splicing would be done for most or all of the destinations in the route table[4]. This offloads the LSR's "in the middle" of a network which makes it easier for the nodes in the middle to handle the aggregated traffic of all the "edges" of the network[5].

Optical IP Switching

The previous section described how MPLS can used to eliminate much of the processing on the data path through a router. An extension of this idea is to eliminate all of the "electronics" on the data path so that we can avoid the "electronic bottleneck" and exploit the enormous capacity of optical fiber.

The Optical IP Switch discussed here is similar in concept to the LSR described in the previous section but with a couple of key differences. Like the LSR, the Optical IP Switch consists of a switch and an RSC. And like the RSC in the LSR, the RSC in the Optical IP Switch:

- implements routing protocols and exchanges routing information with other nodes
- implements the IP protocol and forwards IP datagrams to next hops
- implements an LDP to associate "labels" with route table destinations
- controls a switch to connect an input "VC" to an output "VC"

But where the LSR uses ATM VC's, the Optical IP Switch uses wavelengths, that is different destinations in the route table will be associated with different "colors" of light. And where the LSR uses an ATM switch and label swapping to splice an input VC to an output VC, the Optical IP Switch uses optical technology to route the different wavelengths in different directions. The Optical IP Switch also differs from the LSR in another way. When the LSR splices an input VC to an output VC, the 2 VC's can be different as in the example above in which the LSR splices VC 53 to VC 14. But when the Optical IP Switch splices an input "VC" to an output "VC" it will use the same wavelength for both "VC"s. This simplifies the design of the optical switch since it eliminates the need for wavelength conversion but it does limit flexibility in assigning "VC"s. But as we will see below, this is not a serious problem.

---

[4] ARIS and MPLS also include provisions for using a single VC for all the destinations that are "behind" a given "egress" point. See references 2 and 6 for details.

[5] This is important. From the early days of the NSFnet, it's been a challenge making the "middle of the network" run fast enough to handle the traffic from all of the edges. But by offloading the "middle", MPLS helps the middle keep up with the edges even as the edges run faster and faster.

In a network of Optical IP Switches (or OIP's), the OIP's will run routing protocols, exchange routing information and determine the next hop for various destinations -- just like other routers. And as in a network of ordinary routers, the routes for a given destination will form a tree like the one shown in Figure 3.

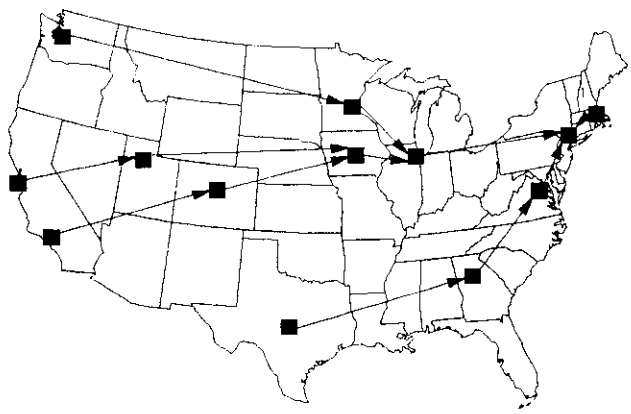

Figure 3

But in a network of OIP's, the OIP's will also use an LDP to establish switched paths to each "egress" where an egress is the point at which traffic *leaves* a network of OIP's. The switched paths to an egress will be used for all the destinations that are "behind" that egress.

The switched paths will be set up as follows. From each egress, the OIP's will grow a switched path tree that is rooted at the egress. The tree will grow upwards from the root to the leaves as described in reference 6. A given tree will use a single wavelength that is passed upwards from the root as branches are added to the tree. The tree shown in Figure 3, which delivers traffic to Westchester County, New York might use a "blue" wavelength, for example. The tree in Figure 4, which delivers traffic to the Chicago area might use a "red" wavelength. (Because a tree grows upwards from the root, it's a simple matter to assign a single wavelength to all of the branches in a given tree.)

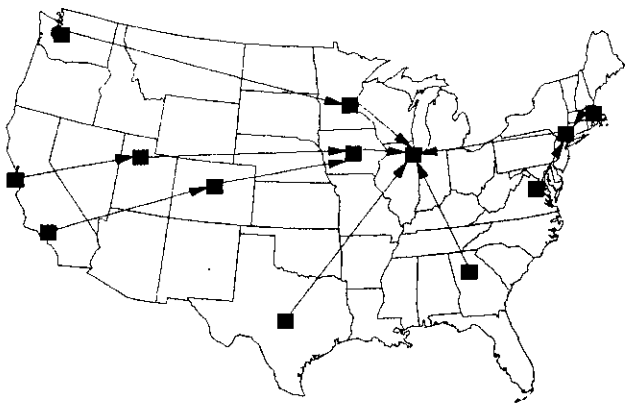

Figure 4

"Merging" is also an issue in setting up switched path trees[6]. At a merge point, like the one in Iowa in the switched path trees shown in Figures 3 and 4, care must be taken so that merging traffic doesn't result in garbled data. In an MPLS-over-ATM network, special hardware is used in ATM switches to avoid what is called the "cell interleaving problem" in which cells from different packets become interleaved at a merge point. The merging problem also needs to be addressed in a network of Optical IP Switches. One solution is to use different "shades" of color at a merge point. For example, the transmission from Salt Lake City to Ames, Iowa in Figure 4 might use a "light blue" wavelength and the transmission from Denver to Ames, Iowa might use a "dark blue" wavelength. Another possibility is to use multi-fiber bundles on the links between OIP's and then arrange things so that the "blue" wavelengths (say) from different sources are carried on different fibers in a bundle. Both of these techniques will be illustrated in the examples below.

---

[6] See references 1, 2, 3 and 6 for details.

A Simple Optical IP Switch

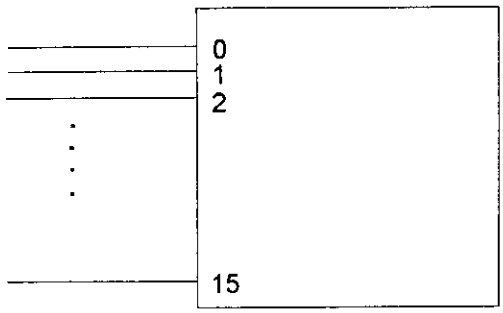

Figure 5

A simple Optical IP switch is shown in Figure 5. The switch connects to 16 optical ports[7] and each port connects to 2 "bundles" of optical fibers. A port receives traffic on one of these fiber bundles and transmits on the other. Each fiber bundle contains 100 hair-thin optical fibers and each fiber can carry up to 100 wavelengths of light. This choice of 100 fibers in a bundle and 100 wavelengths in a fiber will allow us to build a high-speed backbone network of up to 100 nodes. (We could also use k fibers in a bundle and k wavelengths on a fiber to support networks of up to k nodes.)

In a network of these switches, a wavelength will be associated with each egress, e.g. "blue" light for traffic going to Westchester and "red" light for traffic going to Chicago. And a particular fiber number will be associated with each "ingress". (In ARIS and MPLS, a traffic "source" is also known as an "ingress". An "egress", of course, is a traffic "sink".) So fiber i in a bundle of fibers will be used for traffic that enters the network at node i. Since nodes in real networks can be sinks as well as sources, we will assign a unique number between 0 and 99 to each node and we'll use the ith color for traffic that leaves the network at node i and the ith fiber for traffic that enters the network at node i.

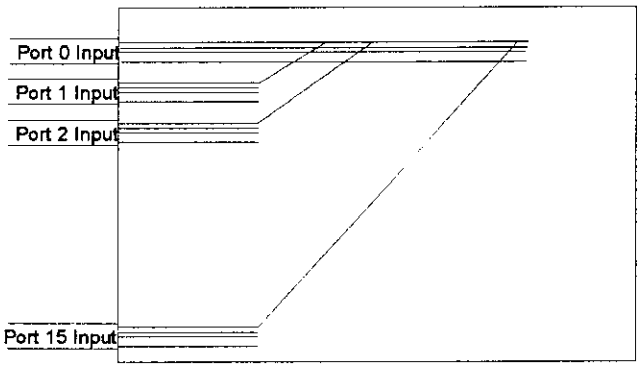

Figure 6

The internals of the optical switch are shown in Figures 6 - 9.

First, the fiber bundles from the 16 input ports are "merged together". By this we mean that fiber-0 from each of the 16 input ports are merged in such a way that light from any of the 16 fiber-0's is passed on to a single "merged" fiber-0. Similarly, fiber-1 from each of the 16 inputs are merged together, and so on for each of the fibers between 0 and 99. This merging allows us to reduce by a factor of 16 the number of fibers that need to be dealt with in the remaining "stages" of the switch. Note that this merging will not result in any garbling problems as long as our routing protocol computes a single route between any 2 points[8]. Since fiber i, say, is only used for traffic originating at node i and since node i's

---

[7] It could just as easily connect to 8 ports or 32 ports or k ports for any "reasonable" value of k (e.g. k <= 100).

[8] This feature is supported by modern routing protocols. Of course, the very high speed that the Optical IP Switch will be able to achieve will also make the computation of alternate routes between a pair of points "unnecessary overhead".

traffic will be arriving at any node j on a single port, no merging problems will occur on fiber i -- or on any of the other fibers for that matter.

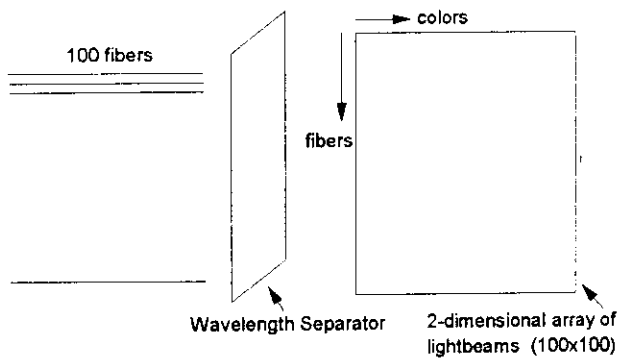

Figure 7

After this first stage of processing, we have a single 100 fiber bundle to deal with. The next step is to separate the wavelengths in the 100 fibers. A wavelength separator of some kind (e.g. a prism or a grating) is used to separate the wavelengths in each of the fibers. This results in a 2-dimensional array of lightbeams in which one dimension corresponds to color and the other dimension corresponds to fiber number as shown in Figure 7.

Since, as we said earlier, the ith color is used for traffic that leaves the network at node i, we want to direct all of the light beams of color i out the appropriate output port so that they can get to node i.

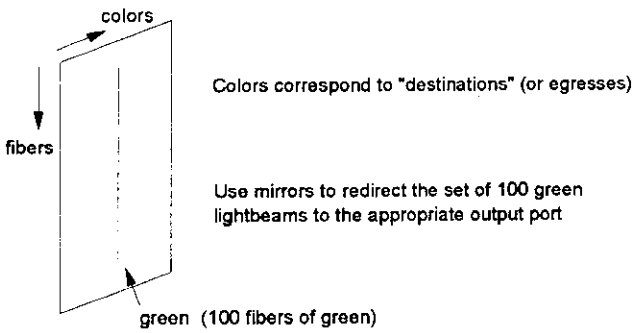

Figure 8

Figure 8 provides another view of this 2-dimensional array of lightbeams.

Since a color, such as "green", corresponds to a given egress and since all the green lightbeams are neatly arranged in a nice column, we can use a mirror to steer this column of green lightbeams to the appropriate output port.

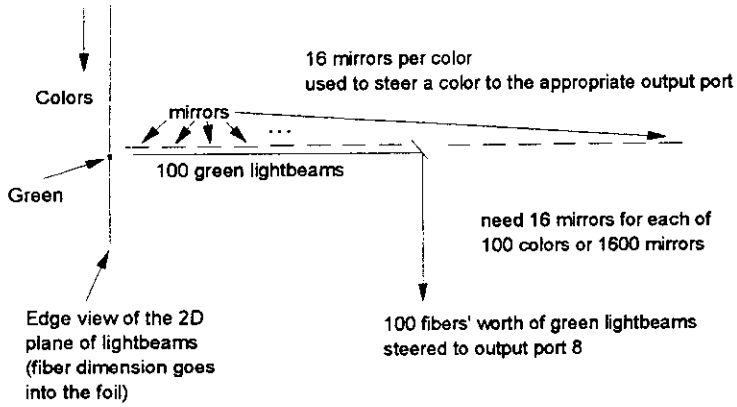

Figure 9

This can be accomplished as shown in Figure 9 which shows another view of the 2-dimensional array of light beams. In this case, we have an edge view of the 2-dimensional plane of lightbeams in which the color dimension goes from the top of the diagram to the bottom and the fiber dimension is perpendicular to the plane of the diagram.

As Figure 9 shows, mirrors are used to steer the 100 green lightbeams to the fiber bundle on the appropriate output port. There is one mirror for each of the 16 output ports and directing the 100 green lightbeams to the appropriate output port is simply a matter of "flipping" the appropriate mirror. There will be a similar array of 16 mirrors for each of the 100 colors of light. Thus a total

of 1600 mirrors is sufficient to steer all of the lightbeams in the 2-dimensional array of lightbeams to the appropriate output ports.

So the process of "splicing" a set of "upstream VC's" to a "downstream VC" as discussed in the ARIS and MPLS references is just a matter of flipping the appropriate mirrors. Note that there are no "merging" or "intermingling" problems when the lightbeams of a given color arrive at an appropriate egress since the lightbeams from the various ingresses arrive on distinct fibers in the fiber bundle connected to the egress.

Note that this Optical IP Switch has some nice properties:

- minimal delay and zero jitter. There are no delays on the path through the switch to buffer data or to convert between optics and electronics. Traffic passes through the switch, literally, at the speed of light.
- large capacity. Consider the case of a 100 node network in which each of the 100 nodes is simultaneously transmitting to each of the other nodes at OC768, say (~40 Gbits/sec). If, as a result of network problems (e.g. power problems, construction workers accidentally cutting a fiber), all the traffic ends up going through a single Optical IP Switch in Chicago, say, the total traffic going to that switch would be on the order of

  40 Gbits/sec * 100 * 100 = **400 Tbits/sec**.

  Although this is quite a bit more capacity than can be handled by today's routers and switches, this would not be a problem for the Optical IP Switch. The RSC in the Optical IP Switch would determine the appropriate routes, build the appropriate switched path trees and flip the appropriate mirrors to steer the various components of traffic in the appropriate directions. But 400 Tbits/sec is not the capacity of the Optical IP Switch. If the traffic sources ran 10 times faster or 1000 times faster, the Optical IP Switch would just handle that much more traffic. Thus the Optical IP Switch solves a fundamental problem that we've had from the early days of the Internet in that it allows the backbone to run fast enough to handle the aggregated traffic of all the edges of the network *no matter fast the edges are*.
- doesn't depend on high-speed electronics. Although the Optical IP Switch can handle very high data rates, it doesn't require very high speed electronics since there are no electronics on the data path through the switch. The Optical IP Switch doesn't use electronics to switch or process individual IP packets and the electronics in the Optical IP Switch don't have to do anything else at pico-second rates even though the switch handles very large traffic volumes. The electronics in the Optical IP Switch are just used to set up "relatively static" optically switched paths that change only when routing topology changes.

Variations on the Design

In the previous section, we discussed an Optical IP Switch that uses 100 fibers in a fiber bundle and 100 wavelengths in an individual fiber (or, more generally, k fibers in a fiber bundle and k wavelengths in a fiber) but other designs are also possible. As we shall see below, there are other designs that require far fewer fibers in a bundle.

Instead of k fibers with k wavelengths, we could also build an Optical IP Switch that could support networks of up to k nodes with single fiber "bundles" that support k^2 wavelengths per fiber. For example, we might use a single fiber between Optical IP Switches with 400

wavelengths per fiber to build a high-speed backbone network of up to 20 nodes. In this case, we could divide the 400 wavelengths into "virtual fibers" and "virtual colors" and use these "virtual fibers" and "virtual colors" in pretty much the same way as we used the physical fibers and physical colors in the previous design.

For example, we could

- divide the 400 wavelengths up into groups of 20 such that the ith block of 20 colors corresponds to the ith "virtual color" (e.g. the various shades of "blue", from "sky blue" to "deep blue" to "navy blue" might correspond to a single virtual color which we might simply call "blue")
- define the jth "virtual fiber" to consist of the jth element from each of the 20 blocks of colors (e.g. virtual fiber 3 would consist of colors, 3, 23, 43, ..., 383)
- use the ith "virtual color" for traffic that leaves the network at node i
- use the ith "virtual fiber" for traffic that enters the network at node i
- use the design described in the previous section pretty much "as is"

In this case the Optical IP Switch would perform the usual tasks of

- merging the fibers from the input ports
- separating the wavelengths in the resulting fiber "bundle" and
- steering the individual colors to the appropriate output port

But in this case, some of the tasks would be a bit simpler than in the previous design.

In the case of merging, we would just need to merge the single fiber from each input port into a single merged fiber. In the wavelength separation, we would only need to separate the wavelengths from a single fiber (rather than k fibers) and we would only need to perform a fairly "coarse" separation of the wavelengths into the "virtual colors" described above. Since the various shades of blue correspond to a single "virtual color" and since all of the shades of that "virtual color" are supposed to go to the same node, it's sufficient to separate the wavelengths into the coarse-grained "virtual colors". The steering of colors to appropriate output ports is also simplified since we only need to deal with the coarse-grained "virtual colors". A 16-port switch would require 16 mirrors to steer one of the "virtual colors" to an appropriate output port and a total of 20 * 16 = 320 mirrors to direct all of the "virtual colors" to appropriate output ports.

Additional Variations

Other variations are also possible. For example, we could use 1000 wavelengths on each of the fibers in a 10-fiber bundle in building an Optical IP Switch that could support a high-speed backbone network of up to 100 nodes.

In this case, we might use the wavelengths as follows:

- Divide the wavelengths in each fiber into blocks of 10
- Use block i on each fiber for traffic that leaves the network at node i. Thus there will be a total of 100 wavelengths available (10 wavelengths on each of 10 fibers) for traffic leaving the network at node i.

- For traffic entering the network at node j that is going to node i, use the jth wavelength of the wavelengths available for node i where the jth wavelength is determined by counting wavelengths starting from block i on fiber 0 and continuing through block i on fiber 1, fiber 2 and so on through block i on fiber 9. In other words, for traffic that enters the network at node j that is going to node i, we use wavelength w on fiber f where
  f = the integer part of j / 10, and
  w = (i * 10) + j % 10 (where j % 10 is the "remainder" when j is divided by 10)

The operation of the switch is as follows:

- The switch merges the fibers from the input ports, merging fiber-0 from each of the input ports into a single merged fiber-0 and doing the same for each of the other fibers from fiber-1 through fiber-9.
- The wavelengths in the resulting merged fibers are separated into the "coarse-grained" blocks of 10 wavelengths each defined above. This results in a 2-dimensional array of lightbeams where one of the dimensions is fiber number, from 0 to 9, and the other dimension is "coarse-grained color" from 0 to 99.
- Mirrors are used to steer the various colors from the merged fibers to the appropriate output ports (after the route protocols and the LDP have determined the proper positions for the various mirrors). As in the first example, the switch would have 16 mirrors for each of the 100 colors that need to be switched.

Of course many other designs are possible. In the last example, we could have divided the 1000 wavelengths into blocks of 100 and used the ith block for traffic leaving the network at node i and the ith wavelength within a block for traffic entering the network at node i. We could also vary the number of fibers and the number of wavelengths per fiber as long as the product of these 2 numbers is greater than or equal to k^2 where k is the maximum network size that needs to be supported. We should also point that although several of the previous examples assumed a network of 100 nodes, most backbone networks today have far fewer than 100 nodes and thus the number of fibers and the number of wavelengths in a real Optical IP Switch could be smaller than in the examples shown here.

A "Solid-State" Optical IP Switch

The Optical IP Switches described in the previous examples used movable mirrors to steer lightbeams to appropriate output ports. But it may also be possible to build Optical IP Switches that don't have *any* moving parts. The advantage of eliminating moving parts, of course, is improved reliability. Figure 10 shows how the lightbeams might be steered in an Optical IP Switch via non-movable mirrors made out of liquid crystals. Figure 10 is basically the same as Figure 9 but the movable mirrors of Figure 9 have been replaced by liquid

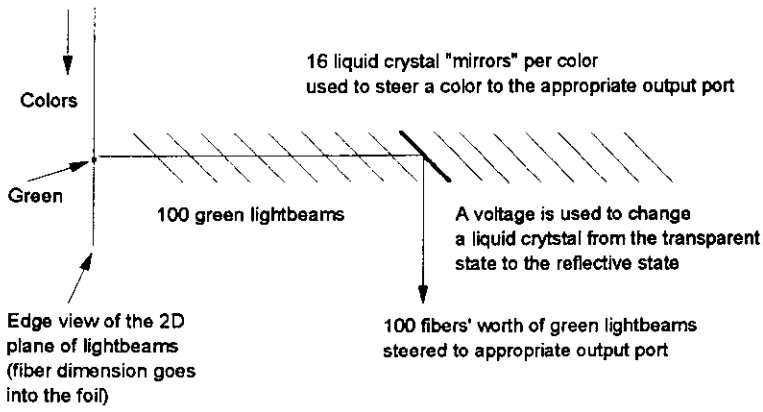

Figure 10

crystals that are installed in fixed positions. The liquid crystals have 2 states. In one state, they are transparent and light passes through them. In the other state they reflect light. As Figure 10 indicates, a voltage can be applied to cause a liquid crystal to change from the transparent state to the reflective state -- and if the liquid crystals are installed as shown in the figure, it may be possible to steer a lightbeam or an array of lightbeams to a particular output port by applying a voltage to the appropriate liquid crystal.

Summary

In this paper we have introduced the concept of Optical IP Switching and examined a number of possible designs for a practical Optical IP Switch. The Optical IP Switch can switch IP traffic at rates that are much higher than those of today's routers and switches since the Optical IP Switch eliminates the electronics and thus "the electronic bottleneck" on the data path through the switch. The very high capacity of the Optical IP Switch may prove to be important if Internet traffic continues to grow at its current high exponential rate. The Optical IP Switch does use more wavelengths than conventional routers and switches but this appears to be a manageable problem and this may be a reasonable price to pay for a device that can switch IP packets, literally, ***at the speed of light***.

References

[1] A Framework for MPLS, MPLS Working Group Internet draft, (in progress)

[2] Multiprotocol Label Switching Architecture, MPLS Working Group Internet draft (in progress)

[3] MPLS using LDP and ATM VC Switching, MPLS Working Group Internet draft (in progress)

[4] US Department of Commerce Report, "*The Emerging Digital Economy*", April, 1998

[5] John Landry, private communication *("The Thunderbolt Proposal", < can we find the original source? >)*

[6] IBM Technical Report, Aggregate Route-Based IP Switching (ARIS), Nancy Feldman, Arun Viswanathan, Rick Boivie, TR 29.2353, February 1998

[7] U. Black, *ATM: Foundation for Broadband Networks*, Prentice Hall 1995

[8] H. Dutton and P. Lenhard, *Asynchronous Transfer Mode (ATM) Technical Overview*, Prentice Hall 1995