ELSEVIER

Computer Communications 25 (2002) 547–556

computer communications

www.elsevier.com/locate/comcom

Terabit switching: a survey of techniques and current products

Amit Singhal[a,*], Raj Jain[b]

[a]ATOGA Systems Inc., 49026 Milmont, Fremont, CA 94538, USA
[b]Nayna Networks Inc., 481 Sycamore Dr, Milpitas, CA 95035, USA

Received 8 October 2001; accepted 8 October 2001

Abstract

This survey paper explains the issues in designing terabit routers and the solutions for them. The discussion includes multi-layer switching, route caching, label switching, and efficient routing table lookups. Router architecture issues including queuing and differentiated service are also discussed. A comparison of features of leading products is included. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Routers; Computer Networking; High-Speed Routing; Routing Table Search Algorithms; Routing Products; Gigabit Routers; Terabit Routers

1. Introduction

In the present network infrastructure, world's communication service providers are laying fiber at very rapid rates. And most of the fiber connections are now being terminated using DWDM. The combination of fiber and DWDM has made raw bandwidth available in abundance. It is possible to transmit 64 wavelength at OC-192 on a fiber these days and OC-768 speeds will be available soon. A number of vendors including Siemens, Ciena, Alcatel, Nortel, and Sycamore have announced transport products capable of 160 OC-192 channel on a single fiber [1]. Terabit routing technologies are required to convert massive amounts of raw bandwidth into usable bandwidth. Present day network infrastructure is shown in Fig. 1. Raw bandwidth has increased by approximately four orders of magnitude whereas capacity of switches and routers have only increased by a factor of 10 approximately and, therefore, the bottleneck. Currently, Add/Drop Multiplexers are used for spreading a high-speed optical interface across multiple lower-capacity interfaces of traditional routers. But carriers require high-speed router interfaces that can directly connect to the high-speed DWDM equipment to ensure optical inter operability. This will also remove the overhead associated with the extra technologies to enable more economical and efficient wide area communications. As the number of channels transmitted on a single fiber increases with DWDM, routers must also scale port densities to handle all those channels. With increase in the speed of interfaces as well as the port density, next thing which routers need to improve on is the internal switching capacity. 64 wavelengths at OC-192 require over a terabit of switching capacity. Considering an example of Nexabit [2], a gigabit router with 40 Gbps switch capacity can support only a 4-channel OC-48 DWDM connection. Four of these will be required to support a 16-channel OC-48 DWDM connection. 16 of these are required to support 16-channel OC-192 DWDM connection with a layer of 16 4 :: 1 SONET Add/Drop Multiplexers in between. In comparison to that, a single router with terabit switching capacity can support 16-channel OC-192 DWDM connection. Current state-of-the art routers by leading vendors like Avici, Juniper, Pluris claim to support up to several OC-192 interfaces as well as multi-terabit switching capacities.

2. The architecture of Internet routers

This section gives a general introduction about the architecture of routers and the functions of its various components. This is helpful in understanding the bottlenecks in achieving high-speed routing and how these bottlenecks are avoided in the design of gigabit and terabit capacity routers available today in the market.

2.1. Router functions

Functions of a router can be broadly classified into two main categories [3]:

1. Data path functions: these functions are applied to every

* Corresponding author.
E-mail addresses: amits@atoga.com (A. Singhal), raj@nayna.com (R. Jain).

PII: S0140-3664(01)00423-6

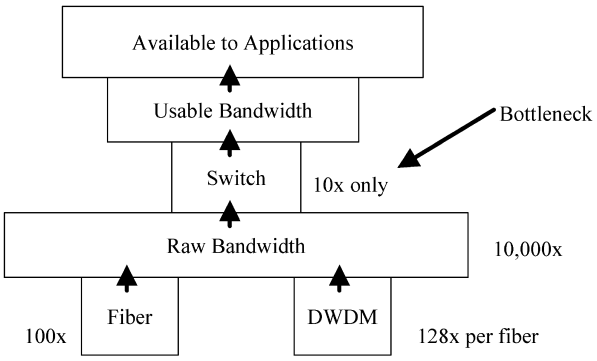

Fig. 1. Present day network infrastructure.

datagram that reaches the router and is successfully routed without being dropped at any stage. Main functions included in this category are the forwarding decision, forwarding through the backplane and output link scheduling.

2. Control functions: these functions include mainly system configuration, management and update of routing table information. These do not apply to every datagram and are, therefore, performed relatively infrequently.

The key goal in designing high-speed routers is to increase the rate at which datagrams are routed and, therefore, data path functions are the ones to be improved. The major data path functions are:

- *Forwarding decision*: routing table search is done for each arriving datagram and output port is determined based on the destination address. Also, a next-hop MAC address is appended to the front of the datagram, the time-to-live (TTL) field of the IP datagram header is decremented, and a new header checksum is calculated.
- *Forwarding through the backplane*: backplane refers to the physical path between the input port and the output port. Once the forwarding decision is made, the datagram is queued before it can be transferred to the output port across the backplane. If there is not enough space in the queue, then it might even be dropped.

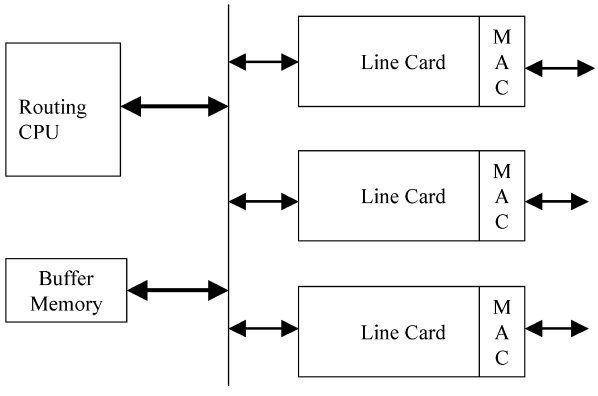

Fig. 2. Architecture of early routers.

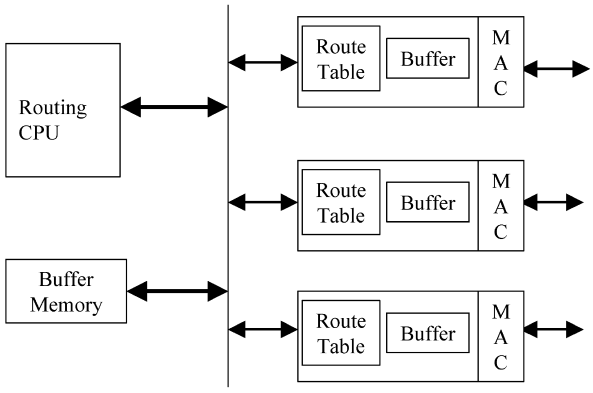

Fig. 3. Router architecture with intelligence on each line card.

- *Output link scheduling*: once a datagram reaches the output port, it is again queued before it can be transmitted on the output link. In most traditional routers, a single FIFO queue is maintained. But most advanced routers maintain separate queues for different flows, or priority classes and then carefully schedule the departure time of each datagram in order to meet various delay and throughput guarantees.

2.2. Evolution of present day routers

The architecture of early routers was based on that of a general-purpose computer as shown in Fig. 2 [4].

These routers had a shared central bus, central CPU, memory and the line cards for input and output ports. Line cards provide media access control (MAC) layer functionality and connect to the external links. Each incoming packet is transferred to the CPU across the shared bus. Forwarding decision is made there and the packet then traverses the shared bus again to the output port. Performance of these routers was limited mainly by two factors: processing power of the central CPU (since route table search is a highly time-consuming task) and the fact that every packet has to traverse twice through the shared bus.

To remove the first bottleneck, some router vendors introduced parallelism by having multiple CPUs. Each CPU now handles a portion of the incoming traffic. But each packet still has to traverse the shared bus twice. Soon, the design of router architecture advanced one step further as shown in Fig. 3. A route cache and processing power was provided at each line card and forwarding decisions were made locally. Each packet traverses the shared bus only once from input port to the output port. Even though each line card has the processing power, but all control functions are still handled by the central routing CPU. All control packets are, therefore, sent there for processing. After processing, control information like routing table update is propagated back to the line cards.

Even though CPU performance improved with time, it could not keep pace with the increase in line capacity of

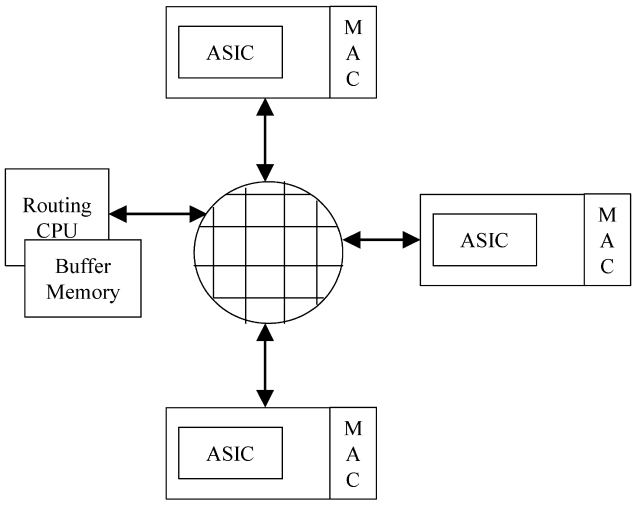

Fig. 4. Router architecture with switched backplane.

the physical links and it was not possible to make forwarding decisions for the millions of packets per second coming on each input link. Therefore, special purpose application specific integrated circuits (ASICs) are now placed on each line card, which outperform a general purpose CPU in making forwarding decisions, managing queues and arbitration access to the bus.

Use of the shared bus allowed only one packet at a time to move from input port to output port. This last architectural bottleneck was eliminated by replacing shared bus by a crossbar switch. This allows multiple line cards to communicate simultaneously with each other as shown in Fig. 4.

2.3. Assessing router performance

In this section, parameters that can be used to grade the performance of new generation router architectures are described [5]. These parameters reflect the exponential traffic growth and the convergence of voice, video and data.

- *High packet transfer rate*: increasing Internet traffic has made the packets per second capacity of a router as the single most important parameter for grading its performance. Further, considering the exponential growth of traffic, the capacity of routers must be scalable.
- *Multi-service support*: most of the network backbones support both ATM and IP traffic and may continue to do so, as both technologies have their advantages. Therefore, routers must support ATM cells, IP frames and other network traffic types in their native modes, delivering full efficiency of the corresponding network type [6].
- *Guaranteed short deterministic delay*: real-time voice and video traffic requires short and predictable delay through the system. Unpredictable delay results in a discontinuity, which is not acceptable for these applications.
- *Quality of service*: routers must be able to support service level agreements, guaranteed line-rate and differential quality of service to different applications or flows. This quality of service support must be configurable.
- *Multicast traffic*: Internet traffic is changing from predominantly point-to-point to multicast and, therefore, routers must support large number of multicast transmissions simultaneously.
- *High availability*: high-speed routers located in the backbones handle huge amounts of data and cannot be turned down for upgrades etc. Therefore, features such as hot-swappable software tasks allowing in-service software upgrades are required.

3. Switching vs. routing

The basic difference between switching and routing is that switching uses ‘indexing’ for determining the next hop for a packet in the address table whereas routing uses ‘searching’. Since indexing in general requires only one lookup, it is much faster than any search technique as discussed later in the paper. Because of this, many network managers started thinking about replacing routers with switches wherever possible and vendors flooded the market with several products under the name of switches. To differentiate their products, vendors gave different names to them like Layer 3 Switch, IP Switch, Layer 4 Switch, Tag Switch, etc. Regardless of what a product does, it is likely to be called a switch [7,8]. Therefore, it is important to understand the difference between all these different forms of switches.

3.1. Switching hubs

They operate at Layer 1 of the OSI networking model. Individual ports are assigned to different LAN segments. While they are useful for managing configuration changes, it must be noted that they still propagate contention among their ports and are, therefore, different from Layer 2 bridges.

3.2. Layer 2 switching

Layer 2 switch is just another name for multi-port bridges. As we know, bridges are used to extend the LANs without extending the contention domain. So Layer 2 switches have been used in some places to replace routers for connecting various LANs to produce one big flat network. But the problem with this approach is the broadcast traffic, which is propagated across all ports of a Layer 2 switch. To solve this problem, the concept of ‘Virtual LAN’ or VLAN was developed. Basic feature of VLAN is to divide one large LAN connected by Layer 2 switches into many independent and possibly overlapping LANs. This is done by limiting

the forwarding of multicast packets in these LANs. There are several ways of doing this:

- *Port-based grouping*: packets coming on a certain port may be forwarded only to a subset of ports.
- *Layer 2 address-based grouping*: the set of output ports is decided by looking at the Layer 2 address of the packet.
- *Layer 3 protocol based grouping*: bridges can also segregate traffic based on the Protocol Type field of the packet (2 bytes between the Layer 2 and Layer 3 address fields).
- *Layer 3 subnet based grouping*: for some Layer 3 protocols like IP, bridges may only forward traffic to other ports belonging to the same IP subnet. For this they have to look at Layer 3 address of the packet.

In brief, VLAN switches modify the forwarding of bridged traffic. Devices referred as Layer 3 VLAN switches, still operate at Layer 2 but they use some Layer 3 information.

Although ATM cell switching has Layer 3 and Layer 4 concepts, it is important to note that in most IP networks, ATM switches are used as Layer 2 products.

3.3. Layer 3 switching

There is no consistent definition of 'Layer 3 switches'. They refer to wide variety of products. The main requirement is that these devices use Layer 3 information to forward packets. Therefore, as discussed in Section 3.2, even Layer 2 VLAN switches with protocol/subnet awareness are sometimes referred as Layer 3 VLAN switches. Other products in this category are:

3.3.1. Layer 3 routing functionality in VLAN switches

Pattern of network traffic is changing and the 80–20 rule [7], which says that 80% of all network traffic is intra LAN, is no longer valid. More traffic is crossing the LAN boundaries these days. To forward this traffic, VLAN switches have to use Layer 3 routing functionality. Traditionally, VLAN switches forwarded such traffic to some route servers. However, as this type of traffic is increasing, it makes more sense to build this functionality within the switches. Many proprietary solutions are available for this.

3.3.2. Layer 2 ATM switching with IP routing

Most of the service providers have invested in ATM technology for their backbones. They need to map IP traffic on the backbone. There are several approaches for mapping Layer 3 traffic on to ATM circuits. Most of them aim at improving routing performance by separating the transmission of network control information from the normal data traffic. Control traffic passes through the routers and route servers while initiating call, while normal data traffic is switched through already established path. There are proprietary solutions for this like IP switching, and there are standard techniques like multi-protocol over ATM (MPOA) as well.

3.3.3. Label switching

Label switching techniques address various issues including WAN route scalability, adding more functionality and high performance. Routing decisions are performed once at the entry point to the WAN and a label is inserted in the packet. Remaining forwarding decisions within the WAN are based on label switching. Tag switching is one proprietary solution based on this approach and the IETF is developing a standard on multi-protocol label switching (MPLS).

3.3.4. Route caching

The number of Internet hosts is increasing at an exponential rate. It is not possible to have an entry for each host in the routing table. Therefore, routers combine many of these entries, which have the same next hop. But this worsens already complex task of route search. To improve route lookup time, many products keep a cache of frequently seen addresses. When an address is not found in the cache, it is searched in the full routing table. Cache sizes range from 2000 to 64,000. Most of these products have a processor based slow-path for looking up routes for cache misses. A few of the products take help of an external router to perform these functions. These are sometimes referred to as 'Layer 3 Learning Bridges'. Route caching technique scales poorly with routing table size, and may not be used for backbone routers that support large routing tables [9]. Frequent topology changes and random traffic pattern may also reduce benefits from the route cache. Worst-case performance is bounded by the speed of the slow full route table lookup.

3.3.5. Full routing

Some of the newer products in the market perform full routing at very high speeds. Instead of using a route cache, these products actually perform a complete routing table search for every packet. By eliminating the route cache, these products have a predictable performance for all traffic at all times even in most complex inter-networks. Unlike other forms of Layer 3 switches, these products improve all aspects of routing to gigabit speeds and not just a subset. These products are suited for deployment in large-scale carrier backbones. Some of the techniques used in these products to improve route lookup are discussed later in this paper.

3.4. Switching above Layer 3

Layer-less switching and Layer 4 switching are the new buzzwords in the industry. Again there is no consistent definition of what these terms mean. Vendors are adding the ability to look at Layer 4 header information into Layer 3 products and marketing them as Layer 4 or Layer-less switches. Products operating at Layers 2 and 3 handle each packet the same way whether it is part of a long flow between two hosts or one traveling alone. But at Layer 4 and higher, there is an awareness of the flows and the higher-level

applications to which this packet belongs. This information can be used to classify packets into different categories and can be used to provide differentiated services and implement service level agreements in the network.

4. Efficient routing table search

One of the major bottlenecks in backbone routers is the need to compute the longest prefix match for each incoming packet. Data links now operating at gigabits per second can generate nearly 1,500,000 packets per second at each interface. New protocols, such as RSVP, require route selection based on protocol number, source address, destination port and source port. Therefore, these protocols are even more time-consuming. The number of memory accesses and the speed of the memory determine the speed of a route lookup algorithm. Various techniques have been proposed to improve route lookup time [10]. They can be broadly classified into:

4.1. Tree-based algorithms

Each node in the tree, from root to leaf corresponds to an entry in the forwarding table and the longest prefix match is the longest path in the tree that matches the destination address of an incoming packet. In the worst case, it takes a time proportional to the length of the destination address to find the longest prefix match. The main idea in tree-based algorithms is that most nodes require storage for only a few children instead of all possible ones and, therefore, make frugal use of memory at cost of doing more memory lookups. But as the memory costs are dropping, these algorithms may not be the best ones to use. In this category, Patricia-tree algorithm is one of the most common. Main difference between a Patricia-tree and a regular binary tree is a skip field that allows prefixes of different lengths to be stored at the same level, thus eliminating the unoccupied intermediate nodes.

4.2. Hardware-oriented techniques

Some of these techniques are as simple as using more memory and have a separate entry for each Internet address. Longest prefix match is not required in this case and complexity of the search is reduced. Other techniques try to reduce the memory access time by combining logic and memory together in a single device. Ternary content addressable memories (TCAM) are suitable for lookup tables. Each bit in the CAM can have three values: 0, 1 or *x* ('don't care'). Multiple hits are allowed and a mechanism is used to resolve the priority among them. TCAMs are currently available from multiple vendors and can sustain 50–66 million lookups per second with predictable latency in the 80–160 ns range.

4.3. Table compaction techniques

These techniques make use of the fact that forwarding entries are sparsely distributed in the space of all Internet addresses. So they use some complicated compact data structure to store the forwarding table in the primary cache of a processor. This allows route lookup at terabit speeds. Degermark et al. [11] proposed multi-stage tables, and used the first stage to resolve the first 16 bits. But instead of direct indexing the table, they compacted it to store populated nodes into continuous space, and then used a clever mechanism to translate IP address into table index, thereby achieving smallest possible table size, while at the same time resolving the first 16 bits in one shot.

4.4. Hash based techniques

Hashing operates strictly on an exact-match basis and, therefore, longest prefix match limits the use of hashing for route lookup. The solution to this problem is to try different masks and choosing the one with the longest mask length. Choice of masks can be iterative or hierarchical. WASHU Algorithm [12] developed at Washington University, St Louis is a scalable algorithm that uses binary hashing. The algorithm computes a separate hash table for each possible prefix length and, therefore, maintains 33 hash tables in total. Instead of starting from the longest possible prefix, a binary search on the prefix lengths is performed. Search starts at table 16 and if there is a hit, looks for longer match, otherwise looks for shorter match. For each prefix P, markers are added in the tables corresponding to shorter lengths, which can be visited by binary search when looking for an entry whose best matching prefix is P. But this can lead to false hits and may require backtracking (searching for shorter length prefixes). To avoid backtracking, this algorithm pre-computes the best matching prefix for each marker and remembers that on a hit.

4.5. Stanford University's algorithm [13]

This algorithm makes use of the fact that most of the prefixes in route tables of the backbone routers are shorter than 24 bits. The basic scheme makes use of two tables, both stored in DRAM. The first table (TBL24) stores all possible route prefixes that are up to, and including, 24 bits long. Prefixes shorter than 24 bits are expanded and multiple 24 bit entries are kept for them. Second table (TBLLong) stores all route prefixes in the routing table that are longer than 24 bits. Each entry in TBLLong corresponds to one of the 256 possible longer prefixes that share the single 24 bit prefix in TBL24. The first 24 bits of the address are used as an index into the first table TBL24 and a single memory read is performed, yielding 2 bytes. If the first bit equals zero, then the remaining 15 bits describe the next hop. Otherwise, the remaining 15 bits are multiplied by 256, and the product is added to the last 8 bits of the original destination address, and this value is used as a direct index into TBLLong, which

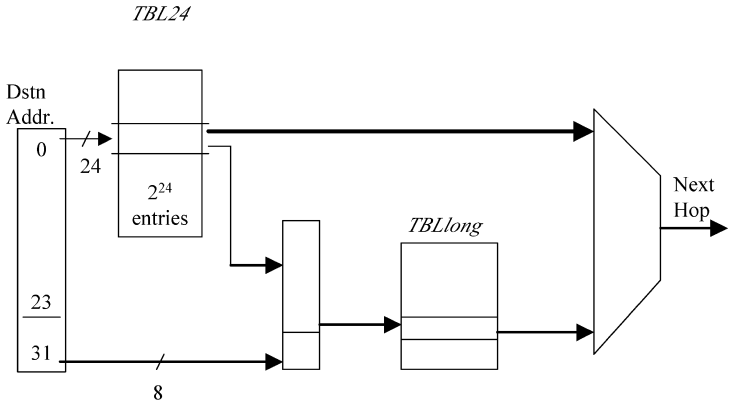

Fig. 5. Stanford University's algorithm for efficient route lookup.

contains the next hop. Two memory accesses in different tables can be pipelined and the algorithm allows 20 million packets per second to be processed. Fig. 5 shows how the two tables are accessed to find the next hop.

Vendors like Pluris, Nexabit (now Lucent), Avici also have their own solutions to route lookup problem. But details have not been disclosed. Route lookup is the single most important thing in the design of high-speed routers and no vendor wants to share its ideas with anyone else.

5. Router architecture for the differentiated services

Providing any form of differentiated services require the network to keep some state information. The majority of the installed routers use architectures that will experience a degraded performance if they are configured to provide complicated QOS mechanisms. Therefore, the traditional approach was that all the sophisticated techniques should be in the end systems and network should be kept as simple as possible. But recent research and advances in hardware capabilities have made it possible to make networks more intelligent [10,14,15].

5.1. Components of differentiated services

Following operations need to be performed at a high speed in the router to provide differentiated services:

Packet classification, which can distinguish packets and group them according to different requirements.

Buffer management, which determines how much buffer space should be allocated for different classes of network traffic and in case of congestion, which packets should be dropped.

Packet scheduling, which decides the order in which the packets are serviced to meet different delay and throughput guarantees.

5.2. No queuing before header processing

The first requirement for differentiated services is that the maximum delay for header processing must be no larger than the delay, a packet from the service class with the least delay can experience. Without this constraint, violation of service assurances can be done even before header processing and that is not allowed. Therefore, packet header processing must be done at wire speeds and not be traffic-dependent. The implication of this is on the design of forwarding engines. It is the worst-case performance of the forwarding engine, which determines the packet processing rate, and not the average case performance. If average case performance is used to determine supported packet processing speeds, buffering will be required before processing.

5.3. Queuing

Once the packet header is processed and next-hop information is known, packet is queued before being transmitted on the output link. Switches can either be input or output queued. Output queued switches require the switch fabric to run at a speed greater than the sum of the speeds of the incoming links and the output queues themselves must run at a speed much faster than the input links. This is often difficult to implement with increasing link speeds. Therefore, most of the switch designs are input queued but it suffers from the head-of-line blocking problem, which means that a packet at the head of the input queue, while waiting for its turn to be transmitted to a busy output port, can block packets behind it which are destined for an idle output port. This problem is solved by maintaining per-output queues, which is also known as virtual output queuing. A centralized scheduling algorithm then examines the contents of all the input queues, and finds a conflict-free match between inputs and outputs. But input queuing poses another challenge for the scheduling. Most of the packets scheduling algorithms are specified in terms of output queues and this is a non-trivial problem to modify these algorithms based on input queuing.

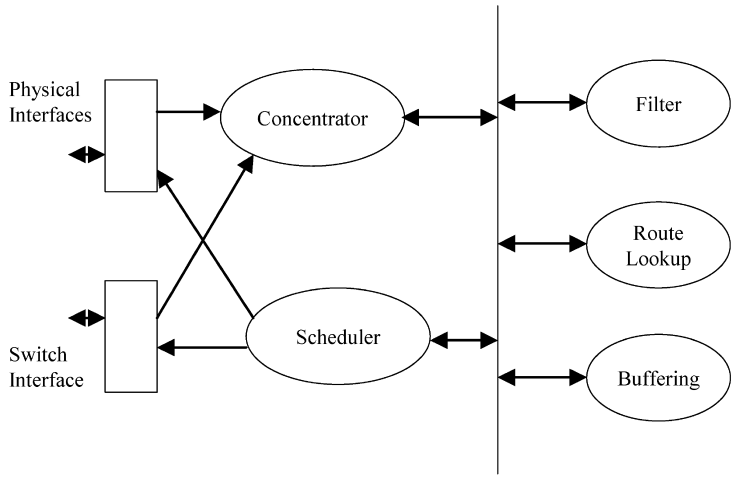

Fig. 6. Router architecture with partitioned forwarding engine.

5.4. Optimized packet processing

Increasing link capacities and the need for differentiated services stretch processor based architecture to the limit. Therefore, multiprocessor architectures with several forwarding engines are designed. Another efficient solution [14] is described here which is based on functional partitioning of packet processing as done below:

- buffer and forward packets through some switching fabric
- apply filtering and packet classification
- determine the next hop of the packet
- queue the packet in an appropriate queue based on both the classification decisions and the route table lookup
- schedule packet transmission on outgoing links to meet QoS requirements.

Processing elements optimized for each task are used in sequence and pipelining is done between these stages. Fig. 6 shows the router architecture based on this approach. Further, combination of this design with the multiple shared processor architecture is also possible to provide very high packet forwarding rates.

6. Survey of products

This section provides a survey of the terabit and gigabit capacity routers available in the market. Comparative analysis of all the major products classifies them into various categories based on architecture design as well as performance. Later, a detailed description of state-of-the-art terabit switch from Stanford University is also given. This competitive study identifies the key router vendors and maps each of them into the landscape of edge, mid-core, and core routing requirements. In addition, the study provides an overview of the critical line capacity and total switching capacity requirements for edge and core environments and compares the various architectural approaches being used to address these performance needs. Many of the data and ideas in this section are borrowed from a white paper at the site of Pluris corporation [16–18].

6.1. Line capacity and total switching capacity

To get into more detailed architectural comparisons, it is important to further define the differences between line capacity and total switching capacity and to know what these values are for various types of scalable gigabit and terabit systems available in the market.

Line capacity:line capacity refers to the effective input/output bandwidth that is available to a subscriber via the line-card ports. For example, a line card that has four OC-48 ports at 2.5 Gbps each would deliver 10 Gbps of line capacity. Invariably, line capacity represents only a percentage of overall switching capacity. Gigabit routing devices typically can provide total line capacity of up to tens of Gbps, and are able to support multiple port interface speeds up to OC-48 (2.5 Gbps) or OC-192 (10 Gbps). Gigabit routing vendors include Cisco [19], Lucent/Ascend, Argon/Siemens, NetCore/Tellabs, Juniper, Nexabit/Lucent, and Torrent/Ericsson. Terabit routing devices are designed with the aggregate line capacity to handle thousands of Gbps and to provide ultra-scalable performance and high port density. These routers can support port interface speeds as high as OC-192 (10 Gbps) and beyond.

Switching capacity: the switching capacity of a system consists of the total bandwidth for all line-card connections and internal switching connections throughout the system. The switching capacity should be substantially higher than the line capacity to ensure non-blocking switching between any two ports. Additional switching capacity is also needed to provide active redundancy and a higher level of fault-tolerance. Therefore, switching capacity includes: bandwidth used for line card connections, bandwidth available to modular expansion of line card connections, bandwidth

Table 1
Single chassis configurations

Product	Capacity in Gbps		Number of line cards	Wan interface support	Number of OC-48 ports	Line card performance (million PPS)
	Switch fabric	Line card				
Single box edge to mid-core devices						
Cisco 12012	60	27	11	OC-3/12/48	8	1
Juniper M40	40	20	8	OC-3/12/48	8	2.5
Lucent Packetstar 6416	60	40	16	OC-3/12/48	16	NA
Torrent IP9000	20	20	16	OC-3/12	NA	NA
Single box mid-core to core devices						
Nexabit NX64000	6400	160	16	OC-3/12/48/192	64	NA
Integrated multi-chassis edge to mid-core devices						
Argon GPN	40	20	8	OC-3/12/48	8	NA
Netcore Everest	20	10	4	OC-3/12/48	4	NA
Integrated multi-chassis mid-core to core devices						
Avici systems TSR	640	100	10	OC-3/12/48/192	10	7
Pluris TNR	1440	150	15	OC-3/12/48/192	60	33

for non-blocking intra-chassis switching, bandwidth for non-blocking inter-chassis switching and for modular multi-chassis expansion, aggregate bandwidth needed to support redundancy and fault tolerance.

6.2. Single box vs. multi-chassis architectures

Architectures from leading vendors can be divided into two broad categories based on how they scale to the increasing network demands:

Single box architectures: traditional single-box designs have high-capacity switching fabrics but they use blocking LAN interconnects to link multiple boxes to increase the overall network switching capacity. Because of the inherent limitations of using external line cards to handle the LAN interconnects, such single-box architectures cannot seamlessly grow their capacities to meet ever-higher traffic requirements. Currently, the router vendors offering high-end, single-box solutions include Lucent/Ascend, Cisco, Juniper, NEO Networks, Nexabit/Lucent, and Torrent/Ericsson. These routers tend to be most appropriate for edge to mid-core and core deployments with maximum line capacities between 25 and 160 Gbps.

Multi-chassis integrated architectures: distributed multi-chassis designs make use of an integrated, expandable switching fabric to provide non-blocking interconnection between multiple expansion chassis. By delivering seamless non-blocking connections between all elements of the system, these integrated architectures can provide smooth scaling to terabit levels and beyond. Most of the integrated multi-chassis solutions range from edge to core applications with maximum line capacities topping out at 160 Gbps for

Table 2
Fully expanded configurations

Product	Capacity in Gbps		Number of line cards	Wan interface support	Number of OC-48 ports	Line card performance (million PPS)
	Switch fabric	Line card				
Single box edge to mid-core devices						
Cisco 12012	60	27	11	OC-3/12/48	8	1
Juniper M40	40	20	8	OC-3/12/48	8	2.5
Lucent Packetstar 6416	60	40	16	OC-3/12/48	16	NA
Torrent IP9000	20	20	16	OC-3/12	NA	NA
Single box mid-core to core devices						
Nexabit NX64000	6400	160	16	OC-3/12/48/192	64	NA
Integrated multi-chassis edge to mid-core devices						
Argon GPN	320	160	64	OC-3/12/48	64	NA
Netcore Everest	1200	640	256	OC-3/12/48	256	NA
Integrated multi-chassis mid-core to core devices						
Avici Systems TSR	36,000	1400	560	OC-3/12/48/192	560	7
Pluris TNR	184,000	19,200	1920	OC-3/12/48/192	7680	33

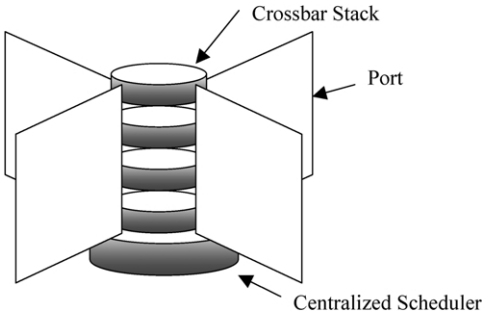

Fig. 7. Architecture of Tiny Tera.

NetCore/Tellabs and Argon/Siemens, 1.4 Tbps for Avici and as high as 19.2 Tbps for Pluris. However, specific architectural and implementation choices can dramatically impact the overall scalability and deployment flexibility of such multi-chassis systems. The current multi-chassis architectures fall into the following categories:

- Star architecture: such architectures expand by using a central switch to aggregate multiple smaller leaf nodes. Examples for this architecture are NetCore/Tellabs and Argon/Siemens. The star architecture has limited scalability and reliability since it relies on a centralized bottleneck and a single point of failure.
- Matrix architecture: such architectures expand by building a scalable matrix of switching elements. Examples for this architecture are Avici [20], with a three-dimensional switching matrix that expands electrically using copper and Pluris, with a multi-dimensional switching matrix that expands optically via fiber-optic interconnects.

6.3. Comparative product positioning

Table 1 shows various single-box and multi-chassis architectures. It compares only 'single-chassis' versions of the multi-chassis systems to better illustrate relative throughputs for their basic configurations. Key factors to consider when comparing single-box and multi-chassis systems are the switch fabric capacity, line card capacity, number of cards supported, WAN interfaces supported (e.g. OC-3, OC-12, OC-48, OC-192), and line card performance in packets per second.

Table 2 provides a relative comparison of fully expanded systems to show the maximum scalability of each type of architecture. It illustrates that single-box systems cannot expand beyond their previous capacities, whereas the multi-chassis architectures are able to deliver significantly more performance than in their single-chassis versions. Among the multi-chassis architectures, systems from vendors such as Argon/Siemens and NetCore/Tellabs provide switching capacities in the 320 Gbps–1.2 Tbps range with line capacities of 160–640 Gbps, which can provide adequate performance to address mid-core routing environments. Systems from Avici and Pluris sit at the next performance level, delivering the terabit and greater switching capacities required for core routing requirements.

6.4. The Tiny Tera

Tiny Tera [21] is a Stanford University research project, the goal of which is to design a small, 1 Tbps packet switch using normal CMOS technology. The system is suited for an ATM switch or Internet core router. It efficiently routes both unicast and multicast traffic. The current version has 32 ports each operating at a 10 Gbps (Sonet OC-192 rate) speed. The switch is a small stack composed of a pile of round shaped crossbar slices and a scheduler. See Fig. 7. Each slice (6 cm diameter) contains a single 32×32 1 bit crossbar chip. A port is connected to the slices radially. The port design is scalable in data rate and packet size. The basic switching unit is 64 bits, called a chunk.

Unicast traffic use a buffering scheme called 'Virtual Output Queuing' described earlier. When the 64 bit data chunks are transferred over the 32×32 switch, the scheduler uses a heuristic algorithm called iSLIP. It achieves fairness using independent round-robin arbiters at each input and output. If the iSLIP algorithm is implemented in hardware, it can make decision in less than 40 ns. The switch also has special input queues for multicast. A multicast input can deliver simultaneously to many outputs. The switch uses fan-out splitting, which means that the crossbar may deliver packets to the output over a number of transfer slots. Developing good multicast scheduling algorithms was an important part of the Tiny Tera project.

7. Summary

It is very clear now that with deployment of more and more fiber and improvements in DWDM technology, terabit capacity routers are required to convert the abundant raw bandwidth into useful bandwidth [22,23]. These routers require fast-switched backplane and multiple forwarding engines to eliminate the bottlenecks provided in traditional routers. Ability to efficiently support differentiated services is another feature, which will be used along with total switching capacity to evaluate these routers. Switching is faster than routing, and many products in the market combine some sort of switching with routing functionality to improve the performance and it is important to understand what the product actually does. But the products that scale up all aspects of routing rather than the subset of them are bound to perform better with arbitrary traffic patterns. Route lookup is the major bottleneck in the performance of routers and many efficient solutions are being proposed to improve it. Supporting differentiated services at such high interface speeds poses some new challenges for the design of router architecture.

References

[1] S. Biagi, Ultra Everything, Telephony, October 16, 2000, http://industryclick.com/magazinearticle.asp?releaseid = 2357&magazinearticleid = 4786&siteid = 3&magazineid = 7.

[2] Nexabit, The New Network Infrastructure: The Need For Terabit Switch/Routers, 1999, 11 p. http://www.nexabit.com/need.html.

[3] N. McKeown, A fast switched backplane for a gigabit switched router, Business Commun. Rev. 27 (12) (1997) http://www.bcr.com/bcrmag/12/mckeown.htm.

[4] N. McKeown, High performance routing and switching, Stanford University Telecom Center Workshop on Routing and Switching, September 1997, http://tiny-tera.stanford.edu/~nickm/talks/Telecom_Center_Workshop_Sept1997.pdf.

[5] Nexabit, Will The New Super Routers Have What it Takes, 1999, 12 p. http://www.nexabit.com/architecture.pdf.

[6] K.Y. Yun, A terabit multiservice switch, IEEE Micro 21 (1) (2001) 58–70.

[7] Decisys, Route Once, Switch Many, July 1997, 23 p. http://www.netreference.com/public/wpapers.htm.

[8] Decisys, The Evolution Of Routing, September 1996, 6 p. http://www.netreference.com/public/wpapers.htm.

[9] C. Partridge, Designing and Building Gigabit and Terabit Internet Routers, Networld + Interop, May1999.

[10] S. Keshav, R. Sharma, Issues and trends in router design, IEEE Commun. Mag. May (1998) 144–151 http://www.cs.cornell.edu/skeshav/papers/routertrends.pdf.

[11] M. Degermark, et al., Small forwarding tables for fast routing lookups, Comput. Commun. Rev. (1997).

[12] M. Waldvogel, G. Varghese, J. Turner, B. Plattner, Scalable high speed IP routing lookups, Proc. ACM Sigcomm, September 1997, http://www-cse.ucsd.edu/users/varghese/publications.html.

[13] P. Gupta, S. Lin, N. McKeown, Routing lookups in hardware at memory access speeds, IEEE Infocom April (1998) http://tiny-tera.stanford.edu/~nickm/papers/Infocom98_lookup.pdf.

[14] V. Kumar, T. Lakshman, D. Stiliadis, Beyond best effort: router architectures for the differentiated services of tomorrow's Internet, IEEE Commun. Mag. May (1998) 152–163 http://www.bell-labs.com/~stiliadi/router/router.html.

[15] C. Partridge, P.P. Carvey, E. Burgess, I. Castineyra, T. Clarke, L. Graham, M. Hathaway, P. Herman, A. King, S. Kohalmi, T. Ma, J. Mcallen, T. Mendez, W. Milliken, R. Pettyjohn, J. Rokosz, J. Seeger, M. Sollins, S. Storch, B. Tober, G. Troxel, D. Waitzman, S. Winterble, A 50-Gb/s IP router, IEEE/ACM Trans. Networking 6 (3) (1998) 237–248.

[16] Pluris, Competitive study, April 1999, 10 p. http://www.pluris.com/html/coretech/whitepaper4.htm.

[17] Pluris, Practical implementation of terabit routing scenarios, April 1999, 14 p. http://www.pluris.com/pdfs/Pluris_Practical.pdf.

[18] Nexabit, NX64000 Multi-Terabit Switch/Router Product Description, 1999, 18 p. http://www.nexabit.com/proddescr.html.

[19] Cisco, Optical Internetworking: A Roadmap to the Data Network of the Future, 10 p. http://www.cisco.com/warp/public/cc/cisco/mkt/servprod/opt/tech/coint_wp.htm.

[20] Avici, The World of Terabit Switch/Router Technology, 1999, 3 p. http://www.avici.com/whitepapers/a_new_world_1.html.

[21] N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick, M. Horowitz, The Tiny Tera: a packet switch core, Hot Interconnects August (1996) http://tiny-tera.stanford.edu/~nickm/papers/HOTI_96.pdf.

[22] K. Lindberg, Multi-Gigabit Routers, May 1998, http://www.csc.fi/lindberg/tik/paper.html.

[23] D. Allen, Terabit routing: simplifying the core, Telecommun. Online May (1999) http://www.telecoms-mag.com/issues/199905/tcs/terabit.html.