Design of A Scalable IP Router

Vibhavasu Vuppala and Lionel M. Ni

Department of Computer Science
Michigan State University
East Lansing, MI 48824-1226
{vuppala,ni}@cps.msu.edu

July 22, 1997

Abstract: *Conventional routers are unable to scale to accommodate the explosive growth of the Internet traffic due to their architectural constraints. Their architecture does not allow for scalability by the addition of more processing nodes or faster bus technology. There has been an emergence of very high speed, economical and scalable SAN/LAN interconnects providing bandwidths of gigabits per second. Design of an IP router composed of off-the-shelf processors interconnected by such SAN/LAN switches is proposed. Implementation of a prototype and its performance are described.*

1. Introduction

The exponential growth of the Internet traffic has long become a cliché. This is not only due to the rapidly growing size of the Internet but also due to the new multimedia rich applications, whose demand for higher bandwidths is never satisfied. This demand for higher bandwidth has been met by the communications industry; technologies like ATM are making gigabit networks feasible. But all these factors are demanding higher performance routers on the Internet for both multicast and unicast traffic.

Today's conventional routers are unable to scale to accommodate the explosive growth of the Internet traffic. This is mainly due to their architectural constraints and the complexity of IP routing. IP routing is a complicated process. It involves sifting through large routing tables, processing various IP options and handling multicast traffic. The architecture of today's routers does not allow for scalability, e.g., by the addition of more processing nodes or faster bus technology. At the same time there has been an emergence of very high speed, economical and scalable SAN/LAN interconnects, such as Myrinet [2] and ServerNet [1]. These SAN/LAN interconnects provide bandwidths of gigabits per second. For example, a Myrinet link composed of a full duplex pair of channels provides a bandwidth of 1.28 Gbps. These switches can be incrementally expanded to provide higher aggregate throughput and thus scalability. A Myrinet multi-port switch can be connected to host computers or other Myrinet switches, resulting in a larger configuration.

This motivates us to combine the SAN/LAN switches with off-the-shelf processors to architect a highly scalable, high-throughput IP router. Figure 1 shows the architecture of the proposed scalable router. It comprises of a set of autonomous *routing nodes* connected by one or more high-speed SAN/LAN switches. The routing nodes are general-purpose computers. Each node has one or more of internal and external network interfaces. The *internal network interfaces* attach to the switching fabric, and the *external network interfaces* attach to LAN or WAN networks. The external network interfaces of different routing nodes may be attached to different networks or to the same network. The architecture provides for high degree of scalability. If more ports are required, more routing nodes or external network interfaces can be added. More internal network interfaces or switches can be added to handle higher aggregate bandwidth. High-end multiprocessors can be used as routing nodes to accommodate larger bandwidth, such as OC-48 and above, at external network interfaces. Packets enter the router through the external network interfaces of the routing nodes. The routing nodes route the input packets in parallel through the switching fabric to the outbound routing nodes. The outbound nodes switch the packets out on to the destination network without going through the IP layer.

Section 2 outlines the related work. In Section 3 operation of the proposed router is described. Section 4 introduces the concept of *remote-network* device. Remote-Network protocol is explained in Section 5. Section 6 briefly describes the implementation of a prototype. Performance measurement is dealt with in Section 7. Section 8 outlines the work in progress.

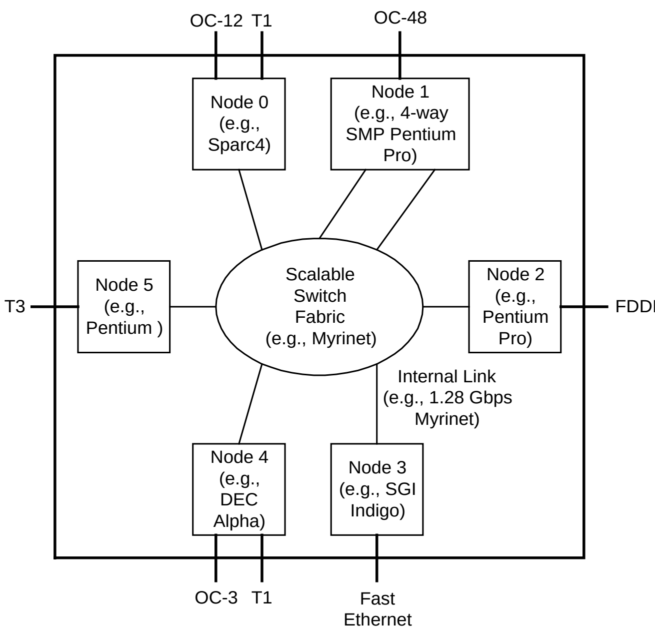

Figure 1. Block diagram of the proposed router

2. Related Work

Realizing the bottleneck introduced by IP routers, different solutions have been proposed. As mentioned in the last section, the problem lies in the router architecture and in the complexity of IP routing. IP-switching [3][8], Tag-switching [4] and Cell Switch Router (CSR) [9] attempt to solve the latter, i.e., the complexity of IP routing, by routing packets at layer-2. Multigigabit routers such as Pluris Inc's Massively Parallel Router (MPR) [5] and Ascend Inc's GRF IP Switch [10] are some of the solution to the architectural issues.

IP-switching dynamically chooses between IP routing and ATM switching, depending on the characteristics of the network traffic. It identifies a flow in IP traffic and assigns it a virtual circuit in the ATM hardware, thus by-passing the IP layer. It suffers from the cost of ATM path establishment for short flows and performs native IP routing for very short IP conversations. The number of flows in an IP-switch is limited by the ATM hardware.

In tag-switching, the edge routers (called the tag-edge routers) attach tags to incoming packets and from then on all the intermediate routers (called the tag-switches) route the packet based on the tags, at layer-2, till the downstream tag-edge router is reached. Each tag switch has a *Tag Information Base* (TIB) which for each input tag specifies the output tag, the output port, and the MAC address of the target. A tag switch hashes into TIB, using the input tag value and replaces the input tag with the output tag and sends the packet through the output port. The entries in a TIB generally correspond to the routing table entries at that tag switch. IP-switching and Tag-switching do not completely eliminate the need for IP routing. With both the approaches, certain nodes will always be heavily dependent on the IP layer. Tag-switches with aggregate routes [4] and IP-switches on the Internet backbone [5] are some such examples.

A MPR router is composed of a large number of single board processing nodes and a proprietary Self-Healing Butterfly interconnect. The processing nodes share the routing load and each has a copy of the forwarding table. A set of processing nodes may be connected to the same LAN/WAN interface through multiplexors. IP routing is done in two steps, first the exit network interface is determined and then one of the processing nodes connected to that interface is chosen. GRF IP Switch similarly has a number of IP Forwarding Media cards connected through a non-blocking crosspoint switch. The Media cards contain the route table, perform IP forwarding and include network interfaces. In the multigigabit router architecture, it is not possible to have multiple interconnects, or multiple network interfaces to a processing node. The proprietary nature of the data interconnects and processing nodes also hinders scalability.

Our approach is simple, cost-effective and the most scalable. It requires simple modification to the existing network layer software. All the hardware components are non-proprietary, and improve in price and performance aspects with the data switching and microprocessor industry. The proposed approach utilizes switching on the routing nodes to avoid IP layer and at the same time provides for a scalable architecture. Our prototype uses four Pentium PCs interconnected by a 4-port Myrinet switch.

3. Router Operation

The basic operation of the router is as follows. An IP packet enters the router through one of the routing nodes (see Figure 1). This ingress node performs traditional IP routing[1] to find the target address (the next hop address or the destination address) for this packet. In addition, it also finds

- A node (the egress node) in the router, which is connected to the destination network,
- The interface (the egress port) on the egress node which is connected to the destination network
- The internal network interface to use in order to reach the egress node.

It then forwards the IP packet with target address and egress port information to the egress node. After receiving the packet, the egress node immediately sends it out through the egress port to the target address. The packet does not go through the IP layer at the egress node. The ingress node acts like a traditional router and the egress node similar to a switch. Note that routing and fragmentation is done at the ingress node, and ARP is done at the egress node.

The information about the egress node, the egress port, and the internal network interface can be incorporated at various layers. For instance, it can be made part of the IP routing table, but this involves changes to IP layer. Instead we propose the concept of *Remote-Network* (RN) devices to keep this information. This has the advantage of using the intelligence of the IP layer to figure out the egress node information but without any changes to the IP layer.

4. Remote-Network Device

The concept of Remote-Network (RN) device is central to our implementation of the scalable router. A RN device is a virtual interface to an IP network. It makes a remote network appear to be directly connected to a node. It is very similar, in properties, to a traditional physical network interface (like an Ethernet interface) except for the following

- There is no actual hardware interface card for a RN device.
- It is not associated with an IP-host address. Instead, it has an IP-network address.
- It is uni-directional. Packets always flow out of this device but they never come in.
- It does not have any hardware (MAC) address associated with it.

A RN device inherits some of the properties of the remote network it represents. It has the same netmask and MTU as the remote network[2]. But RN devices do not allow ARP. In addition, each RN device is associated with a tuple (**N, P, I**) where

- **N** is one of the routing nodes in the router which is directly connected to the network that

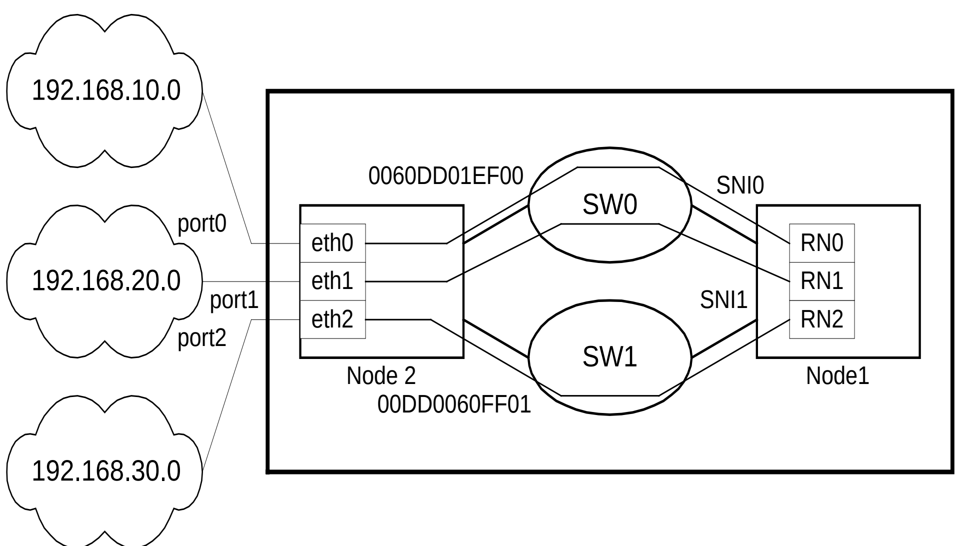

Figure 2. RN devices at Node 1 for the networks at Node 2

[1] Unless the packet is coming from a tag-edge router or a tag-switch, in which case the packet does not go through IP layer.

the RN device represents (the egress node).

- **P** is the external network interface on **N** which is connected to the network which the RN device represents (the egress port).
- **I** is the internal network interface to be used to reach **N** from the node on which the RN device is defined.

Each node in the router has RN devices defined for all the networks directly attached to all the other nodes. Each RN device also has a corresponding entry in the routing table to direct all the packets destined for the remote network through the RN device. Figure 2 illustrates three RN devices *RN0*, *RN1* and *RN2* defined on node **1** for the three networks attached to node **2.** *RN0* represents the 192.168.10.0 network and its path to reach the network is given as (*0060DD01EF00, 0, SNI0*) where *SNI0* is the internal *switch network interface* (SNI) to use, *0060DD01EF00* is the layer-2 address of node **2** (the format of the address is dependent on switching fabric) and the egress port number is *0*. Node **2** has a table for mapping port numbers to operating system dependent interfaces. Figure 3 shows the data structures corresponding to the three RN devices on node **1.**

Port	**Interface**
0	Eth0
1	Eth1
2	Eth2

Port-Interface map at Node 2

RN	**(N,P,I)**
RN0	(0060DD01EF00,0,SNI0)
RN1	(0060DD01EF00,1,SNI0)
RN2	(00DD0060FF01,2,SNI1)

RN tuple table at Node 1

Destination	**Gateway**	**Interface**	**…**
…	….	…	…
192.168.10.0	Local	RN0	…
192.168.20.0	Local	RN1	…
192.168.30.0	Local	RN2	…

Routing Table at Node 1

Figure 3. Information of RN devices

[2] If the MTU of the switching fabric is smaller than the MTU of the remote network, then the RN device has the MTU of the switching fabric.

5. RN Protocol and Software Components

The ingress and egress nodes use the RN protocol to communicate. Each RN packet has a checksum of 2 octets, followed by 2 octets of egress port number, 4 octets of the target address and finally the IP packet.

The IP layer (on ingress node) gives the incoming IP packet and target address to the RN device driver (RND). The RN device driver attaches the target address and the egress port number (associated with the RN device) to the IP packet and sends it to the egress node. At the egress node the packet is handed to the RN packet handler (RNP), the receiving portion of RN protocol. It strips off the RN header, performs ARP on the target address, and hands the packet over to the device associated with the egress port. To make port numbers operating system independent, each node has a table that maps port numbers to network interfaces. If the target address matches the address of egress port, the packet is sent to higher layers for local delivery.

Figure 4 shows the flow of IP packets, between an ingress and an egress node, through various software layers. The circles in the figure represent software layers.

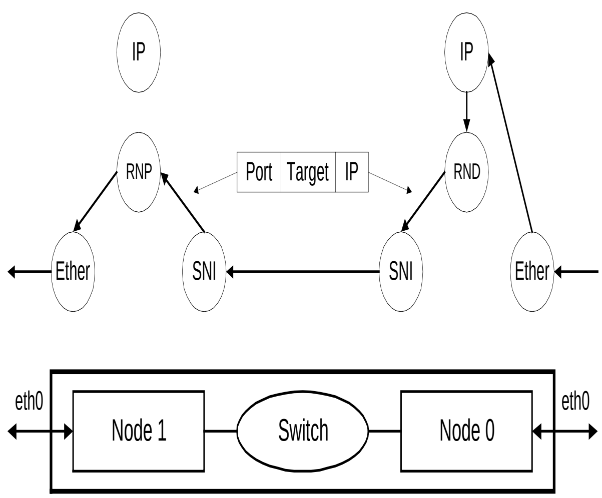

Figure 4. RN Software Components

6. Implementation

A prototype router was implemented using four Pentium PCs running Linux interconnected through a four-port Myrinet switch. The RND and RNP software components were implemented as one Linux loadable kernel module. The Linux *ifconfig(8)* command was used to set the IP-network address, netmask and MTU for the RN device. Another program, *rnconfig*, was developed to set the egress node, the egress port, and the internal network interface for a RN device. The port to device mapping table is also configured using the *rnconfig* command.

7. Performance Measurement

The proposed router architecture has two stages, the ingress stage and the egress stage. This separation of stages provides scalability but may effect the performance. So for now we take a look at the effects of the egress stage on the performance rather than the overall performance of the router. It was found that this separation of stages did not have any effect on the throughput or frame loss rate but resulted in less than 5 percent increase in latency.

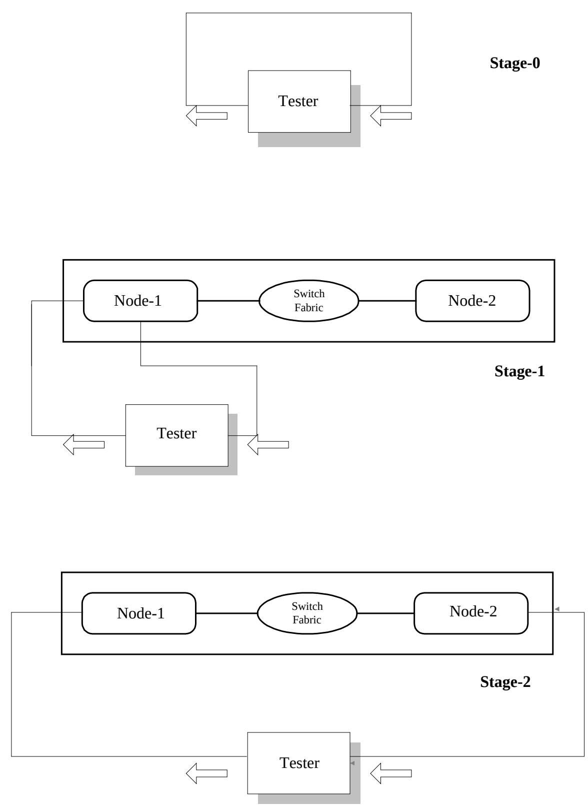

Figure 5. Test Setup Configurations

The various benchmarking test setup configurations are shown in Figure 5. Methodologies specified in RFC 1944 were followed for performing the benchmarks. In Stage-2 the Device Under Test (the prototype router) is connected to the transmitting and receiving ports of the Tester. In Stage-1 only the ingress node is connected to the Tester whereas in Stage-0 the transmitting and receiving ports of the Tester are connected directly. All the benchmarks were run for these three configurations. A two-node router was used for these tests. A Sparc-Station-20 running Solaris 2.5.1 was used as the Tester. Only 10Mb/s Ethernet ports were used for the benchmarks. Null (port 9) UDP packets were used for all the tests.

7.1 Throughput

Throughput is defined as the maximum rate at which none of the offered frames is dropped by the device [6]. Throughput for various test configurations

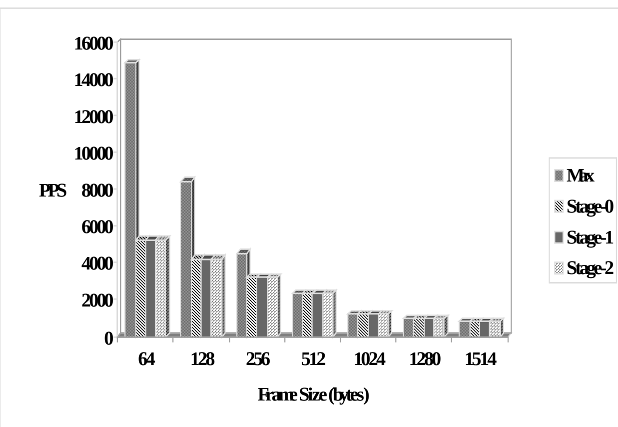

Figure 6. Throughput

and the theoretical frame rates for 10Mb/s Ethernet [7] are shown in figure. It is seen that the egress stage of the router does not have any impact on the throughput of the router. Note that the Tester could not generate frames at the theoretical frame rate for smaller frame sizes. Frames of size 1518 bytes [7] were getting fragmented and hence 1514 byte sized frames were used.

7.2 Frame Loss Rate

Frame loss rate is defined as the percentage of frames that should have been forwarded by a network device under steady state load that were not forwarded due to lack of resources [6]. It was measured for frame sizes of 64, 128, 256, 512, 1024, 1280 and 1514 bytes. No frame loss was noticed for all the test configurations.

7.3 Latency

Latency for store and forward devices is the time interval starting when the last bit of the input frame reaches the input port and the ending when the first bit of the output frame is seen on the output port [6]. The measurements shown in Figure 7 do not show the latency as defined above. Instead it shows the difference between the time the frame was transmitted and then received by the Tester. This includes the transmission times on the media. Note that Stage-1 and Stage-2 involve an extra leg of transmission delay compared to Stage-0. It is seen that there is less than 5 percent increase in this latency due to the egress stage of the router. A good part of this increase is suspected on the mis-aligned Linux network buffers in the implementation of RN device.

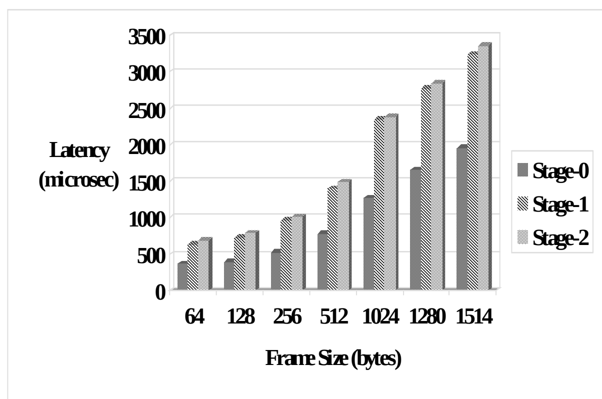

Figure 7. Latency

8. Future Work

Routing protocols may run on some or all of the routing nodes of the proposed router. Issues related to the distributed computation of routing table and its distribution to all routing nodes are being investigated. Various thresholds that move the bottleneck among the router elements such as processor, interconnect, memory and the internal bus are being sought after. The answers will help in the deployment of an optimal router configuration for a given specification. Issues related to switching packets at layer-2 even at the ingress node are being looked into. Ideas of IP and Tag switching may be utilized for this purpose. In the current design only one internal network interface can be associated with a RN device. Association of multiple internal network interfaces to a single RN device is being designed. This will facilitate load balancing and fault tolerance among routing nodes. Plans to include routing nodes of different processor types running varied operating systems into the prototype are underway.

9. Conclusion

Conventional routers are not able to scale to the exponential growth in the Internet traffic. As a solution, a scalable IP router architecture was designed and implemented using off-the-shelf processors and high-speed data interconnects. Central to the design is the concept of Remote-Network devices that provide uni-directional connection to remote networks. A prototype was implemented using Pentium processors running Linux, connected through a multi-port Myrinet switch. Benchmarks run on the prototype indicated that the egress stage of the router did not have any effect on the throughput and frame loss rate but added less than 5 percent to the latency. The proposed scalable IP router has the following special characteristics.

- It is scalable both in terms of network aggregate bandwidth, number of ports, and the routing processing capacity.
- It is cost effective because off-the-shelf and commodity components are used.
- Our approach can be applied to any high performance switches, such as ServerNet or ATM.
- It allows for the use of heterogeneous external network interfaces and heterogeneous routing nodes.
- It provides for a very balanced deployment of the router. Routing nodes of appropriate processing power can be chosen for each external network port. Similarly switches of appropriate bandwidth can be used to match the aggregate bandwidth of the external network ports.

Acknowledgement

We would like to thank Hong Xu of Cisco Systems and Scott Bradner of Harvard University for their help and suggestions.

References

[1] R. Horst, et al, "Performance Modeling of ServerNet Topologies," *Proc of the International Parallel Processing Symposium*, pp. 518-523, April 1996.

[2] Nanette J. Boden et al, "Myrinet -- A Gigabit-per-second Local-area Network", *IEEE Micro*, Feb 1995.

[3] Peter Newman et al, "Flow Labelled IP: A connectionless approach to ATM", *Proceedings of IEEE Infocom*, March 1996.

[4] Cisco Systems, "Scaling the Internet with Tag Switching", White Paper, http://www.cisco.com

[5] Pluris Inc, "Pluris Massively Parallel Routing", White Paper, http://www.pluris.com

[6] S. Bradner, "Benchmarking Terminology", RFC 1242.

[7] S. Bradner and J. McQuaid, "Benchmarking Methodology", RFC 1944.

[8] Peter Newman et al, "IP Switching and Gigabit Routers", *IEEE Communications Magazine*, Jan 1997.

[9] Y. Katsube et al, "Toshiba's Router Architecture Extension for ATM : Overview", RFC 2098

[10] Ascend Inc., "GRF IP Switch", http://www.ascend.com