A distributed architecture of IP routers

Tasho Shukerski, Vladimir Lazarov, Ivan Kanev

Abstract: *The paper discusses the problems relevant to the design of IP (Internet Protocol) routers or Layer3 switches with distributed architecture. A survey has been made on the different modules in the architecture, the tasks that must be processed by them are clearly defined and the packet flow trough such kind of equipment is explained. The main goal of the paper is to describe the place and the role of the network (packet) processors in this distributed switching platform, to define their architectural features and to show the advantages that will bring their implementation.*

Key words: *Network processor, Packet processor, IP routing, Layer3 switching*

INTRODUCTION

Internet traffic is growing exponentially, and different kinds of traffic are being integrated into the Internet Protocol. In the environment of high performance IP Networks the growing customer demand for internet bandwidth has placed excessive load on network providers. Today's router architectures utilize distributed CPU designs enabling data flow of packets to proceed unimpeded along the "fast" path, while management functions such as route table updates are handled separately across the "slow" path. With centralized management and control CPU, line-card processing has evolved for delivering faster "wire-speed" forwarding. Networking requirements continue to be driven by higher wire-speeds and more complex multi-services (e.g. IPv4, IPv6, MPLS, ATM, multicast, VoIP, VLAN) and to meet these demands a new generation of routers, delivering more functionality, increased packet processing capabilities, and new feature sets is driving the need towards fully-programmable fast path architectures.

The goal of this paper is to make a review of the distributed architecture of IP routers and layer3 switches and to define its different modules. Special attention was paid to network processors as they offer fast speed with integrated programmability and the purpose is to show their architectural features.

1. BASIC ROUTER FUNCTIONALITY

The basic functionalities in an IP router can be categorized as: route processing, packet forwarding, and router special services. The two key functionalities [7] are route processing (i.e., path computation, routing table maintenance, and reachability propagation) and packet forwarding.

- **Route Processing**

This includes routing table construction and maintenance using routing protocols (such as RIP or OSPF) to learn about and create a view of the network's topology. Updates to the routing table can also be done through management action where routes are added and deleted manually.

- **Packet Forwarding**

Typically, IP packet forwarding requires the following:

Remove the data link layer header and IP packet validation: The router must check that the received packet is properly formed for the protocol before it proceeds with protocol processing. This involves checking the version number, checking the header length field and calculating the header checksum.

Destination IP address parsing and table lookup: The router performs a table lookup to determine the output port onto which to direct the packet and the next hop to which to send the packet along this route. This is based on the destination IP address in the received packet and the subnet mask(s) of the associated table entries. The result of this lookup could imply a local delivery to the router, a unicast delivery to a single output

port and a multicast delivery to a set of output ports that depends on the router's knowledge of multicast group membership.

The router must also determine the mapping of the destination network address to the data link address for the output port (address resolution or ARP). This can be done either as a separate step or integrated as part of the routing lookup.

Packet lifetime control: The router adjusts the time-to-live (TTL) field in the packet used to prevent packets from circulating endlessly throughout the internetwork. A packet being delivered to a local address within the router is acceptable if it has any positive value of TTL. A packet being routed to output ports has its TTL value decremented as appropriate and then is rechecked to determine if it has any life before it is actually forwarded. A packet whose lifetime is exceeded is discarded by the router (and may cause an error message to be generated to the original sender).

Checksum calculation: The IP header checksum must be recalculated due to the change in the TTL field. Since a router often changes only the TTL field (decrementing it by 1), a router can incrementally update the checksum when it forwards a received packet, instead of calculating the checksum over the entire IP header again.

Determine data link layer address of the next hop.

Add data link layer header to the packet.

- **Special Services**

Anything beyond core routing functions falls into this category: packet translation, encapsulation, traffic prioritization, authentication, and access services such as packet filtering for security/firewall purposes. In addition, routers possess network management components (e.g., SNMP, Management Information Base (MIB), etc.).

As we will focus our research on the user access layer [6] there are used layer3 switches that combine layer 2 switching task together with IP routing functionality. This kind of devices has a great number of Ethernet ports and makes the routing between VLANs (Virtual Local Area Networks)[9] internally eliminating the use of external router. As a result from the research that was made, on Fig. 1 is built a block scheme illustrating how the incoming layer2 unicast and layer3 unicast packet will be basically processed. No other functions such as filtering or classification or protocols such as STP (Spanning Tree Protocol) are presumed.

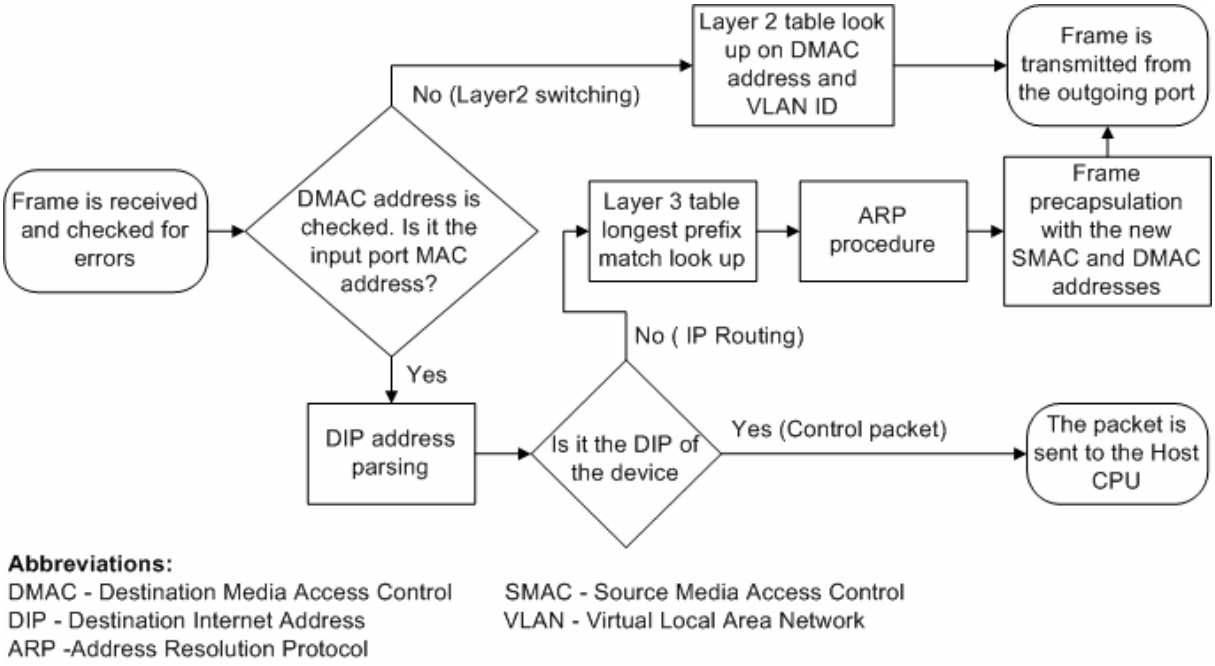

Fig. 1: Layer2 and Layer3 switching

2. GENERIC ARCHITECTURE OF AN IP ROUTER

Routers have traditionally been implemented purely in software. Because of the software implementation, the performance of a router was limited by the performance of the processor executing the protocol code. To achieve wire-speed routing, high-performance processors together with large memories were required. This translated into higher cost. Thus, while software-based wire-speed routing was possible at low-speeds, for example, with 10 megabits per second (Mbps) ports, or with a relatively smaller number of 100 Mbps ports [7], the processing costs and architectural implications make it difficult to achieve wire-speed routing at higher speeds using software-based processing.

The three main bottlenecks in such router architecture are: processing power, memory bandwidth, and internal bus bandwidth [1]. These three bottlenecks can be avoided by using a distributed switch based architecture with properly designed network interfaces [1][7][6]. The summary results that were achieved through the research give the view of distributed architecture of an IP router (Layer3 switch) that is shown on Fig 2.

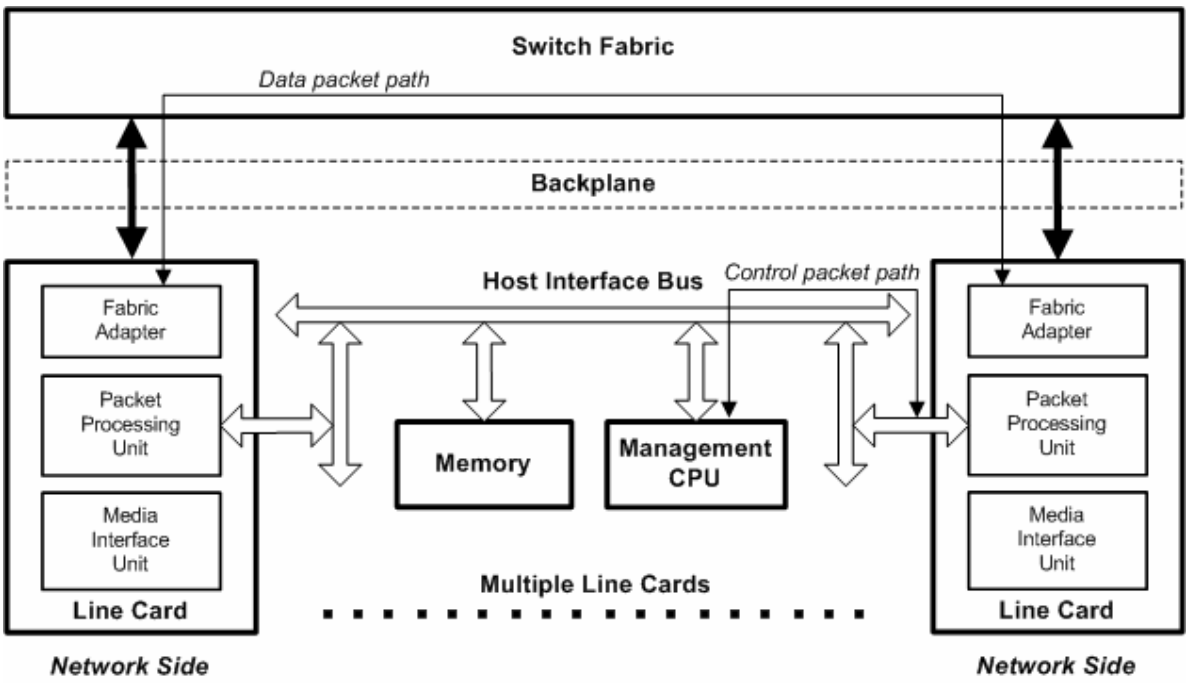

Fig.2: Basic distributed router architecture

Since routers are mostly dedicated systems not running any specific application tasks, off-loading processing to the network interfaces reflects a proper approach to increase the overall router performance. A successful step towards building high performance routers is to add some processing power to each network interface in order to reduce the processing and memory bottlenecks. Each network interface provides the processing power and the buffer space needed for packet processing tasks related to all the packets flowing through it. The Management CPU runs which ever routing protocols is needed in the router and updates the local forwarding informational base in the line-cards. The third bottleneck (internal bus bandwidth) can be solved by using special mechanisms where the internal bus is in effect of a switch thus allowing simultaneous packet transfers between different pairs of network interfaces.

2.1 SWITCH FABRIC

The switch fabric routes data from one of many inputs to any one of many outputs. Packets are fragmented in fix-sized cells at inputs and reassembled at outputs for

transmission. The three main types of switch fabric are shared memory (for low capacity routers), shared bus (for medium capacity routers) and crossbar (for high capacity routers) [1]. There are four main requirements for the switch fabrics [1]:

■ Switch fabrics must provide a method to switch the packets from input ports to outputs ports.

■ The switch fabric must arbitrate traffic when more than one packet arrives concurrently destined for the same output port.

■ Switch fabrics must provide sufficient buffering to handle situations where the packet input rate is greater than the switch fabric's throughput capability. The two possible locations for buffering are at the input of the switch fabric (input queuing) or internally to the switch fabric (shared-memory).

■ The switch fabric card must manage flow control on egress packets at the output of the switch fabric (output queuing).

2.2 BACKPLANE

The backplane is a circuit board containing sockets into which other circuit blocks can be plugged. The backplanes can be divided into active and passive. Active backplanes contain, in addition to the sockets, logic circuitry that performs computing functions. Passive backplanes, in contrast, contain almost no computing circuitry. Passive backplanes dominate in commercial applications as they exhibit superior fault tolerance and make it easier to repair or upgrade system components.

2.3 LINE CARDS

On Fig. 3 is shown a detailed scheme of a line card that is built as a result from the made investigations.

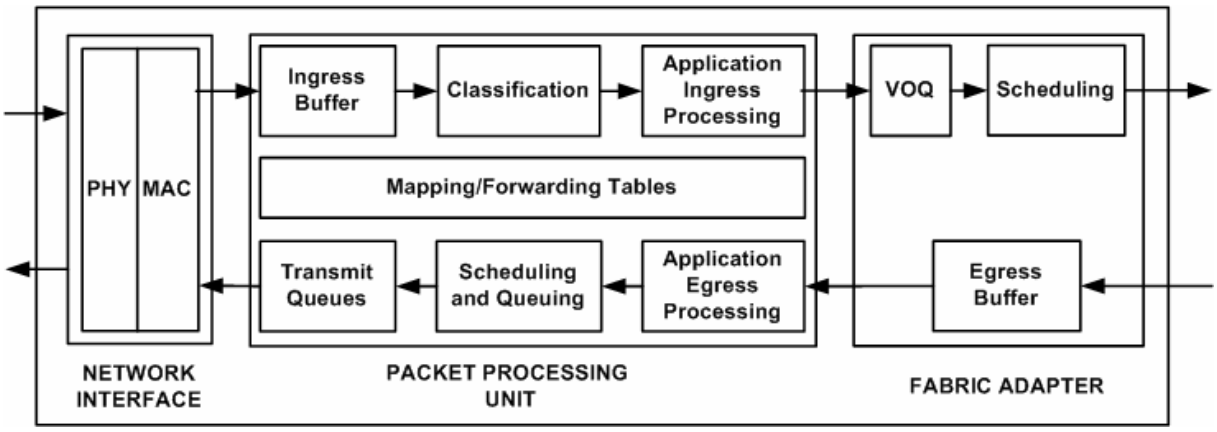

Fig. 3: Line Card Structure

2.3.1 NETWORK INTERFACE

The network interface consists of a Physical Module sub layer (PHY) and a Media Access Control module (MAC) sub layer. There are standardized interfaces between MAC and PHY module. In that way MAC can provide interface to different physical implementations of a network technology. If we take for example Ethernet networks the physical layer can be 10/100 BASE-T copper media or 100BASE-FX fiber media. The MAC sub layer is concerned about the frame manipulation, full duplex or half duplex transmitting and has no impact on the switching and the forwarding behavior of the device. There is standardization in the interface part between Network Interface and the Packet Processing Unit and although that the functions of this interface unit will differ depending on the type of the line interfaces, after the interface termination, the processing will be almost identical.

2.3.2 PACKET PROCESSING UNIT

The Packet Processing Unit resides between the network interface and the Fabric Adapter. It performs the processing of protocol that lies in the critical path of data flow [8]. In the Packet Processing Unit are marked two different flows – ingress and egress [8]. The Mapping/Forwarding Tables are used to be implemented protocol functions. Some protocol functions [1] as classification (map each packet to a predefined flow/connection), scheduling (decide when and which packet to transmit) and queuing (decide when and which packet to drop) are separated from application processing because they are additional to basic switch functions and can be implemented separately in ASIC or FPGA chips. Application ingress processing covers for example Ethernet switching and IP routing and application egress processing – MAC header replacing.

The Packet Processing Unit can be based on different solutions such as ASIC, co-processors, FPGA and general-purpose processors [8]. Traditionally FPGA implementation is more flexible than ASIC and faster than general purpose processors, but the best resolution is to use Application Specific Instruction Processor (ASIP), also called Network processor - a software programmable device with architectural features and/or special circuitry for packet processing at wire speed. It is possible to be made such architecture, because all the manipulations that are accomplished on network packets can be classified into six different categories: pattern matching, look-up, computation, data manipulation, queue management, and control processing [8]. Network processors bridge the divide between ASICs and CPUs by providing a device that is as programmable as a CPU but as fast as an ASIC.

There are some important architecture features that must be emphasized about network processors [8][5][4][3][2]:

■ Wire-speed could not be achieved if one packet is treated at a time. These processors allow multiple packets to be proceeded simultaneously.

■ Tasks involved in network processing are very specific. There must be specialized hardware to speed up common operations.

■ There are specialized instructions that utilize the specialized hardware.

■ A general purpose control processor may reside in the network processor.

■ Multiple processors elements(PE) (also called micro engines, channel processors or task optimised processor) are used to take advantage of the inherit parallelism involved in datagram processing. The PEs are mainly RISC (Reduced Instruction Set Computer) cores or Very Long Instruction Word (VLIW) based architectures. In general, RISC based network processors have their PEs arranged in parallel, so that as a packet enters the network processor it is assigned to a PE, which performs all the necessary processing on the packet. In VLIW architecture the PEs are organised in a pipeline fashion with each PE having a different functionality to perform the required processing on the packet.

■ Multithreading is used, where more than one task can be run on each PE (mainly RISC core), with the PE switching between tasks when one task becomes idle. To ensure fast switching between tasks, the network processor has hardware support for the task switching.

■ Each PE may have locally a small amount of internal memory for storing the program code. However, the amount of available internal memory will be limited in size and interfaces to external memory are provided for storing lookup tables. Accessing the external memory can cause a bottleneck so latest technologies are used to provide faster memory accesses.

2.3.3 FABRIC ADAPTER

The Fabric Adapter implements the egress buffering of the packets that come from the switch fabric and schedules the packets that are destined to the switch fabric. For the case when there is a single FIFO queue at each input a serious problem referred to as

head-of-line (HOL) blocking can substantially reduce achievable throughput. In particular for uniform random distribution of input traffic, the achievable throughput is only 58.6% [1]. A way of eliminating the HOL blocking is by changing the queueing structure at the input. Instead of maintaining a single FIFO at the input, a separate queue per each output can be maintained at each input. To eliminate HOL blocking, virtual output queues (VOQs) [1][6] were proposed at the inputs. However, since there could be contention at the inputs and outputs, there is a necessity for an arbitration algorithm to schedule packets between various inputs and outputs [1].

2.4 MANAGEMENT CPU

The Management CPU is a general-purpose processor that handles the control (slow) path in the router. When the processing unit in the line card recognizes a control packet it sends it through the host interface bus to the management CPU. Using PCI bus is a typical implementation of the host interface bus.

CONCLUSIONS

To achieve wire speed switching in routers they must be built in modular design with distributed processor architecture in witch the most protocol tasks must be concentrated in the line network cards where local forwarding information base reside. The best solution for the computing power in the line cards is to use network processors that offer programmability that makes the architecture flexible and ready to implement new network protocols and to satisfy user traffic demands.

REFERENCES

[1] Chao, Jonathan H., Cheuk H. Lam, Eiji Oki. Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers. John Wiley & Sons, 2001.

[2] Comer, D. Network Processors: Programmable Technology for Building Network Systems. The Internet Protocol Journal, Cisco Systems Inc. December 2004.

[3] Future Software Ltd. Challenges in Building Network Processor Based Solutions. White paper, 2003.

[4] Heppel, A. An Introduction to Network Processors. Roke Manor Research Ltd, 2003.

[5] Kohler, M. NP Complete. Embedded Systems Programming, 2000.

[6] Masatoshi, K., S. Nojima, H. Tomonaga. IP Router for Next-Generation Network. FUJITSU Sci. Tech., June 2001.

[7] Misra, K., F. Kharoliwalla. Study of Internet Router Architectures. 2001.

[8] Shah, N. Understanding Network Processors. Berkeley University, 2001.

[9] Varadarajan, S. Virtual Local Area Networks. Ohio State University, 1997.

ABOUT THE AUTHORS

Tasho Shukerski, PhD student, Department of Computer Systems, Technical University of Plovdiv, E-mail: tshukerski@tu-plovdiv.bg, Phone: +359 32 659704

Prof. Vladimir Lazarov, PhD, Institute for Parallel Processing, Bulgarian Academy of Sciences, E-mail: lazarov@bas.bg

Ivan Kanev, Department of Computer Systems, Technical University of Plovdiv, E-mail: ikanev@it-academy.bg, Phone: +359 32 659704