Network Processors: The Optimal Building Block for Next Generation IP Routers

November 2000
Introduction ¬ 1

The Router Augmentation Workaround ¬ 1

The Race to Deliver a Next Generation
Edge Router ¬ 2

Software-based Solution ¬ 2

Asic-based Solution ¬ 3

Network Processor Solution ¬ 5

Network Processors in the Alcatel 7420 ESR ¬ 7

Packet Flow Through the Alcatel 7420 ESR ¬ 8

Summary ¬ 9
Introduction

Historically, the edge of the network has been served by routers that have migrated from the core of the network. As bandwidth demands increased at the core of the network, backbone router vendors simply introduced bigger, faster routers. Service providers then took the older backbone routers and migrated them out to the edge of the network.

Unfortunately, this traditional migration strategy has, in the last year or so, started to break down. The pervasive influence of the Internet and e-business has created an enormous demand for bandwidth as well as value-added services over IP networks — and the edge has become the undisputed place where all the value-added services need to be applied. It is at the edge of the WAN network where the service provider accepts the customer’s connections and intelligently marks those connections in order to process the packets in different ways. Because of this, it is no longer practical to redistribute packet forwarding equipment from the core of the network to the edge.

While network demands have changed dramatically, router architectures have not. The classic router is a purpose-built device. Every backbone router that has been built in the past 20 years has been optimized for the same job: to receive packets, look up addresses, and forward packets to the next destination in the network — and to perform this task for the largest number of packets in the shortest time possible.

If there were a global design specification for every backbone router on the planet, it would read:

- maintain the topology of the network
- exchange routing information with other routers
- calculate routes
- forward packets
- allow for configuration and management

The problem is that service providers cannot risk sub-optimizing the router’s performance in order to have the router start handling packets with greater intelligence. It is a frustrating trade-off; each additional process that a router applies to a packet degrades the router’s packet forwarding abilities to some extent.

The Router Augmentation Workaround

In response to this problem, carriers have been forced to develop a workaround that involves augmenting traditional routers with other purpose-built devices that perform these value-added services. These devices are generally deployed at the edge of the network, as a kind of front-end to the backbone routers. Typical examples include:

- **VPN routers:** these routers encrypt and decrypt the contents of the packets before they reach the router, which can then simply forwards encrypted packets.

- **Broadband remote access routers:** these purpose-built routers perform authentication and distribution of IP addresses.

Service providers are even considering deploying purpose-built boxes at the edge of the network for quality of service (QoS), because QoS in backbone routers degrades the router’s performance significantly. (QoS is also extremely difficult to administer in a conventional router.)

There are numerous problems with the current solution of deploying a variety of purpose-built boxes to front-end the router. A multi-box solution:

- is hard to manage
- introduces more points of failure
- does not scale well
- offers poor price/performance

This “router augmentation” model is, in fact, contrary to the business and technical models of every service provider, which is to make the network easier and less expensive to manage (which usually translates to fewer devices and vendors). Clearly, the industry cannot keep going in this direction.
The Race to Deliver a Next Generation Edge Router

We are now seeing an architectural competition for next generation edge routers. The three dominant architectural solutions in contention are:

- software-based routers
- ASIC-based routers
- network processor-based routers

The competition will be won by the architecture that can meet all of the following requirements for a next generation edge IP router:

- Higher port density
- Increased throughput
- The ability to perform deep packet inspection, classification and policy enforcement (i.e., QoS) for thousands or tens of thousands of connections without impacting throughput or performance (i.e., packet forwarding)
- The flexibility to provision new services easily and adapt to changing industry standards

As illustrated in Table 1, of the three architectures, only network processor-based routers address all of these requirements equally well. In the case of the other two architectures, there is always a trade-off in one or more critical areas, as shown later in Figure 1.

Network processor-based routers represent the optimal building block for next generation IP networks. This paper reviews these three architectures in detail and shows what they are optimized to do, and how they perform their functions. The paper also suggests how these different routers can be optimally deployed in the service provider’s network to take advantage of the strengths of each architecture.

Software-based Solution

The software-based solution treats the router as an application that runs on a general-purpose computer. Basic GateD software running on a UNIX-based workstation with a few network interface cards is a perfect example of the software-based router. Commercial routers have long been designed in much the same way, using a general purpose CPU, with the routing and forwarding code running as an application. General purpose hardware used in ordinary computers is often used in the implementation of pure software routers. The network interface is modeled after a general-purpose network interface card. There is no hardware assistance for doing any network task apart from data link or physical layer encapsulation and de-encapsulation.

This software-based approach has a number of advantages. It allows the vendor to introduce new features in a rapid and predictable manner. Furthermore, the customer is able to obtain new features and error corrections with a software upgrade. This preserves the investment in the hardware platform.

The downside to the software-based router is that a general purpose CPU is not optimized to the tasks necessary for the operation of a next generation router as described above. A centralized CPU running a software-based router can handle most of the tasks individually. However, it is poorly suited to servicing these tasks collectively, with proper attention given to each task.

The maintenance of the topology is of paramount importance, but computation cycles needed to accomplish this task must not degrade the forwarding performance of the router. The same is true for route calculation, which needs to converge to a stable route topology in a timely fashion. The route calculation process must neither block packet forwarding nor be blocked by packet forwarding.

<table>
<thead>
<tr>
<th>Router Architecture Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Density</td>
</tr>
<tr>
<td>Performance/throughput</td>
</tr>
<tr>
<td>Intelligence/QoS (for high traffic volumes and without impacting performance)</td>
</tr>
<tr>
<td>Ease of provisioning new services and adapting to changing standards</td>
</tr>
</tbody>
</table>
The software-based router suffers from the lack of separation of the control, data, and management planes. Proper separation of the planes is important since traffic load in one of the planes should not affect performance in another plane. Since the central CPU is called on to forward packets, maintain the topology, calculate routes, and provide a management interface, all operations inevitably suffer from load sensitivity. This prevents the pure software-based router from scaling significantly. Indeed, the best pure software-based routers can manage only a few OC-3 ports or a single OC-12 port at best.

When one considers the dimension of value-added services, such as QoS, advanced statistics and billing, VPNs, the pure software-based router becomes even less attractive. It has been our experience that a software-based router’s forwarding performance steadily declines as more features or filters are enabled, placing a greater burden on the CPU. An extension of this solution is to use conventional CPUs on individual line cards within a router, combined with a central CPU. The idea here is to distribute the computation intensive operations across a number of processors. While this is an improvement, it still suffers from the limitation that the CPU on the line card must handle control, data, and management tasks. Furthermore, the conventional CPU is not optimized to the task of moving large amounts of input/output (I/O) traffic. Indeed, the conventional CPU uses the same bus for instruction storage as it does for data storage, resulting in an unavoidable dependency of code performance upon traffic load.

As Figure 1 illustrates, within a software-based router architecture, a bottleneck is evident at the CPU add/drop (A/D) bus. The central CPU must maintain the route database, maintain the protocol state machines, and perform packet-forwarding operations. Even if the CPU on the network device line card is augmented with a copy of the route data base, the control and data paths still place mutual burdens on one another as they compete for the computation and I/O resources of the CPU. Distributing the processing burdens is a good approach, but it accomplishes little unless accompanied by careful separation of the control and data path, and by improved I/O performance of the CPU.

ASIC-based Solution

Application specific integrated circuit (ASIC)-based routers offer one solution to the problems presented by pure software-based routers. In such a design, ASICs are combined with a conventional RISC or CISC CPU in order to balance the workload. Typically, the ASICs handle the data plane and portions of the control plane, whereas the CPU handles the management plane and the bulk of the control plane.1

The ASIC hardware is optimized to the tasks of classifying, filtering, and forwarding IP traffic, and the CPU provides a number of the value-added services, such as statistics collection and billing services. This solution scales well, as the workload can be distributed across ASICs throughout the router to the individual line cards. Also, the control and data paths through the router can be efficiently separated.

1. The control plane is the set of frames processed by the router that contain protocol control information. This includes protocol control frames at Layers 2, 3, and 4. The data plane is the set of frames processed by the router that contains user application data. This is also referred to as the user plane in ATM terminology. The management plane is a subset of the control plane that contains frames that are specific to the management of a protocol, such as SONET OAM frames and ATM ILMI frames.
The ASIC-based solution is well suited to core routers performing bulk forwarding, where the diversity of routing protocols that must be supported are limited, and there is no concern for layered services. In core routers, performance scalability is the chief concern, and ASICs as well as network processors may be used to address this requirement.

The problems with an ASIC-based solution concern feature scalability and investment protection. The addition of new features to an ASIC-based router often means hardware changes. Routing ASICs are designed with a specific feature set, and growing that feature set means replacing the silicon. This is a highly likely occurrence. Layer 2 protocols are largely stable, yet there are still areas of emerging standards, as in the case of MPLS and Diff-Serv. But even if one discounts the Layer 2 protocols, the instability increases as the ASICs reach further into the packet header to perform classification. If classification includes Layer 4 or Layer 7 information, then the hardened ASIC solution proves even less attractive. Consider a function like network address translation (NAT) that must alter the IP addresses embedded in application layer packets. As new applications are introduced, the NAT implementation must be augmented to translate any addresses that are in the application-level packets. This is a simple matter for software to address, but it can require new hardware, depending on the implementation. To be certain, a feature such as NAT really only appears in the edge of the network. Because of the requirement for feature flexibility, ASICs, while appropriate for the core of the network, are undesirable for edge routers.

The need to replace a module to obtain new features, or to fix a bug, hardware upgrades translate to a higher total cost of ownership and potential service outage to customers. Certainly field-programmable logic exists, but it is only now nearing the point where the density and the performance can meet the needs of today’s network, which is growing daily.

The growth of the commercial Internet places additional scalability strains on the ASIC-based solution. A recent report from RHK on Internet traffic growth predicts that Internet traffic will grow at a rate of 300 percent annually. More equipment must be deployed in order to meet that growing demand. It is in the service provider’s interest to find a solution that scales as much as possible and has the lowest total cost of ownership.

Even if the service provider is willing to tolerate a hardware upgrade to obtain new features, the design cycle of an ASIC-based router can be significantly longer than that of a software-based solution. The high cost of ASIC hardware revisions requires a conservative design and verification process. A software-based design requires no less conservative a verification cycle; however, new software versions can be produced much more quickly, for far less money. The result for the service provider is that a properly verified new feature may be delivered more quickly to the market using software instead of ASICs.

As illustrated by Figure 2, within the ASIC-based architecture, the ASIC is directly in the data path of the switch. The ASIC is connected directly to the physical interface and may or may not be responsible for the framing of the data, segmentation and reassembly (SAR) functions, or other Layer 1 operations. A separate CPU that is responsible for configuration and management controls the ASIC. This CPU may

![Figure 2: General ASIC-based Router Architecture](image-url)
gather statistics, service interruptions, manage route tables, and perform any of a number of other operations. It is important to note that the management interface is independent of the data path. This separation of the control and data paths is a further advantage of the ASIC design.

**Network Processor Solution**

The network processor (NP) is a relatively recent innovation that is designed to combine the speed of an ASIC with the flexibility of a conventional CPU to meet the scalability and performance requirements of a next generation router.

The NP is a highly integrated device optimized for the acceleration of packet processing. It is characterized by high I/O bandwidth, an array of hardware assist elements and, most importantly, programmability. It is this last feature that distinguishes an NP from a custom network ASIC. The programmability of an NP is more than simple writing of registers on a chip to configure features. The NP employs a machine language designed to enable the hardware accelerators and also to enable the development of custom algorithms that perform line rate inspection and modification of network traffic. Additionally, an NP is often characterized by an array of processing elements that allow distribution of workload within the NP, and by parallel execution of algorithms.

The NP, like the ASIC, is in the data path of a networking device, and is optimized to perform or enable the following functions:

- **Classification** – Each frame processed by the router is examined in real time. Classification is the process of mapping a series of bytes in the frame header to a user-defined profile that determines the class of the frame. The frame class then determines the policy, or class of service, that the router will use to process the frame.

- **Filtering** – Filtering is the process of explicitly discarding received frames that match a user-defined profile. Like classification, each frame must be examined in real time in order to make the drop decision.

- **Forwarding** – Forwarding is the process of determining the destination port for a frame received by the router, and then delivering the frame to that port.

- **Marking** – In IETF DiffServ terms, marking is the process of associating a per-hop behavior (PHB) codepoint with the classified frame by writing the codepoint into the IP ToS field in order to simplify downstream processing of core routers. When referring to random early detection (RED) queue policing, marking is the process of identifying a queue entry as discard eligible. The RED use of marking is similar to the frame relay concept of setting the discard eligible bit on traffic that is out of profile for the CIR.

- **Policing** – Policing is similar to marking, but measures conformance to a predefined traffic user profile. Frames that are received by the router are examined according to a traffic profile. Those frames outside the profile are marked. Marked, or non conforming frames are either discarded or scheduled for processing after the frames that are conforming.

- **Quality of Service (QoS)** – The ITU defines QoS as “the collective effect of service performance, which determines the degree of satisfaction of a user of the service.” In practical terms, QoS is a function of the delay, delay variance (jitter), and throughput. Differing applications have different sensitivities to delay or throughput. Traffic may be classified according to protocol, application, or other criteria, and assigned a particular QoS level.

- **Statistics Collection** – Statistics collection is the maintenance of transmit and receive statistics. These statistics are used to gauge the level of activity at a particular interface in a router and to support customer billing.

- **Traffic Shaping** – Traffic shaping is an application of one type of QoS. This is the process of modifying the flow of a class of traffic to meet a certain user-defined profile. For example, real time audio is highly sensitive to jitter but not to end-to-end delay. A QoS profile that runs bursty traffic through a controlled leaky bucket in order to minimize jitter is an example of a traffic shaping application.

- **Type of Service (ToS)-Based Routing** – RFC 791 defines the Internet protocol. The type of service field is an eight-bit field in the standard IP header. ToS-based routing uses this field to assign precedence to the datagram. The IETF DiffServ standardization efforts seek to codify the first six bits of the field by mapping them into specific PHBs that define different classes of service.
The NP features a data bus for the network traffic that is separate from the bus used for CPU data and instructions. This clean separation of network traffic from algorithmic execution is a key advantage for the NP.

It is important to note that the fundamental policies that control the flow of traffic through the NP are implemented in software. If these policies were to be hard-wired, the NP would be no more flexible than an ASIC. Instead, by providing hardware assistance where it is needed and by providing a clean separation of the I/O and the computing busses, the NP achieves the high performance of a dedicated ASIC with the flexibility of a software-based solution. The NP also scales well since, like an ASIC, it can be deployed on individual line cards distributing both the computation-intensive as well as I/O-intensive operations.

Customers benefit significantly from the NP design because, like the software-based router, new functionality can be introduced with a software update. Additionally, the NP is flexible enough to support expansion of features, thereby preserving the customer’s hardware investment.

The hardware manufacturer also benefits significantly from the NP design since the risk, time, and expense of a custom ASIC development is eliminated. To be certain, an NP is a complex device that has learning-curve overhead associated with the development cycle, however, emerging common NP API efforts promise to make software integration easier. This will result in quicker time-to-market, which benefits producers and consumers alike. Furthermore, the NP can, in some areas of router design, replace a number of discrete devices reducing transfer cost, part count, power consumption, and module real estate. The NP also allows for the bulk of the router design effort to be focused on the delivery of value-added services. Since time is not spent updating ASICs, resources may be dedicated to delivering value-added software services, that could not be considered in a software-based router, due to CPU resource limitations.

Figure 3 illustrates a general model of a network processor in a switch device. Note that this model is almost identical to the ASIC-based model. In this diagram, the NP sits directly in the data path of the switch. The NP is connected directly to the physical interface and may or may not be responsible for the framing of the data, SAR functions, or other Layer 1 operations. A separate CPU that is responsible for configuration and management controls the NP. Note that early attempts at NP implementations used an in-band management mechanism that caused the data and control paths to be shared. Such a design, whose limitations under high network load are obvious, suffers from the resource allocation problems described in the introduction to this paper.

Although a single NP block is shown, it is often practical and desirable to provide dedicated NPs for each function, and to separate transmit and receive functions entirely. This separation allows the algorithms for transmit and receive to develop independently. The reference implementation discussed later uses this model.
The NP provides the distinct advantage of feature scalability and preservation of hardware investment that is not present in ASICs. Additionally, the NP is commodity silicon and is significantly less expensive for the hardware manufacturer to deploy. This contributes further to the lower total cost of ownership for the NP solution.

Network Processors in the Alcatel 7420 ESR

The Alcatel 7420 Edge Services Router (ESR) makes extensive use of network processors in implementing the forwarding engine module (FEM). The FEM is responsible for forwarding, filtering, classification, queuing, statistics generation, policing, and protocol encapsulation. Figure 4 shows the basic organization of the FEM.

The FEM uses four NPs. Two pairs are dedicated to inbound and outbound processing, respectively. The inbound side has a look-up engine that provides hardware assistance for packet classification and destination resolution. The field programmable gate arrays (FPGAs) function as bus protocol translation devices that convert the 64-bit fast bus to the access busses for the physical interface and the switch fabric.

There is a FEM on each line card in the Alcatel 7420 ESR, giving linear performance scalability as line cards are provisioned in the system. Each FEM runs microcode that can be changed with a software upgrade. All knowledge of protocol encapsulations, queuing disciplines, traffic management, etc., is contained in the FEM microcode. This allows the FEM to evolve as the standards evolve, since new functionality can be introduced with a software upgrade.

Note that there is no packet processing in the switch fabric. The fabric is a basic high speed interconnection configured for a full mesh at system initialization time. Each packet is tagged by the FEM upon receipt and is switched to the appropriate output port by the switch fabric. Note also the separation of the control and data planes, as well as the transmit and receive paths. This optimization makes the design highly scalable and allows the transmit and receive code to be developed somewhat independently of one another.

The PCI bus implements the control and management path for the FEM. Forwarding table updates, policy changes, and statistical information are delivered over this bus, independent of the traffic going to and from the switch fabric.

This highly modular design allows, for not only rapid introduction of new software features, but makes the development of new hardware options easier too, as the interfaces to the PHY and switch fabric are simple and easily understood.
Packet Flow Through the Alcatel 7420 ESR

Packets are received by the 7420 ESR through a physical interface and are delivered first to the inbound data NP. The inbound data NP performs a lookup based on the channel on which the frame was received. The channel is configured by the user and determines the protocol encapsulation. With the protocol encapsulation determined, the inbound data NP forwards the frame header information to the inbound control NP for further processing.

The inbound control NP performs a content addressable memory (CAM) look up to determine the classification profile, and/or a longest match engine (LME) lookup to determine forwarding information. With the frame classified, the inbound control NP updates the appropriate statistics, subjects the frame to policing, if configured, and places the outbound queue and port into a frame notification (FN) message for outbound processing.

The FN message is delivered across the switch fabric to the outbound FEM at the same time that the inbound data NP is notified of the inbound control NP lookup results. The inbound data NP modifies the frame data based on the look-up results, and awaits a transfer request (TR).

The outbound control NP receives the FN and queues it according to the classification information placed in the FN by the inbound look-up process. The outbound control NP implements all of the queuing policy established by the user for the traffic class. The FN moves through the queue and, when the FN is ready for transmission, the outbound control NP signals the outbound data NP to schedule an output slot for the frame. The outbound control NP then sends a TR to the inbound data NP, which has been holding the received frame, awaiting the TR. Upon receipt of the TR, the Inbound Data NP sends the frame across the switch fabric, with a pre-pended data transfer (DT) message, to the outbound data NP, which notifies the outbound control NP of the departing frame, updates the outbound statistics, and transmits the frame out of the physical port.

In the event that the queuing policies in the outbound processing determine that the frame is to be dropped, the outbound control NP sends a frame drop (FD), and the inbound data NP drops the frame and adjusts the frame statistics appropriately.

This process is highly pipelined and takes advantage of the multiple microengines in each NP. The frame notification, transfer request, data transfer, and frame discard messages are relatively small and consume a comparatively small amount of the switch fabric bandwidth in processing. Of course the switch fabric bandwidth needs to be designed to accommodate the worst case of non-blocking processing of a continuous stream of 40-byte frames. It is for this reason that the switch fabric in the Alcatel 7420 ESR runs at twice the capacity of the fastest line cards.

The fast path processing of frames completely bypasses the CPU module. Enabled by the distributed NPs, this is an important performance advantage for the 7420 ESR. The CPU is involved when control or exception traffic is involved. In this case, the determination is made early by the Inbound Control NP, and the destination port is given as the CPU module. The CPU module has a FEM that allows it to participate in the switching of frames in a manner identical to the line cards. Exception and control frames are delivered to the CPU for processing, and the CPU is able to source frames on the switch fabric for delivery to an arbitrary port. For the user, this means that the CPU is involved only in the processing of control and exception frames and is not involved in the per-packet operations in the fast path.

Advanced services like classification, queuing, filtering, marking, traffic shaping, and policing are implemented in the fast path by the NPs. The user configures the services using the network management interface on the CPU module, and the CPU distributes the policies to each of the FEMs in the system. The NP handles the implementation of the policy exclusively. The 7420 ESR leverages the advantages of the NPs to distribute the per-packet processing workload and frees the management CPU to handle the maintenance of the routing protocols, L2 protocols, chassis management, and fault management.
Summary

The network processor is the optimal architecture for a next generation edge router. It effectively combines the performance advantages of an ASIC-based router with the flexibility and rapid time-to-market advantages of a software-based router. The NP has an advantage over ordinary RISC processors in that the I/O portions of the processor are designed to meet the high bandwidth requirements of networking operations. It also has an advantage over ASIC-based routers in its ability to sit directly in the path of a high speed network interface, enabling it to perform complex programmable deep packet classification operations, at wire-speed, that would be impractical to commit to a hardened ASIC, and impossible to accomplish with a general purpose RISC CPU.

The combination of performance and flexibility that make the network processor the optimal building block for next generation IP routers and, in particular, edge routers that must scale both performance and traffic engineering features. The Alcatel 7420 ESR utilizes network processors in a unique architecture to distribute the workload and separate the control and data planes throughout the device. This enables the delivery of value-added network services in a timely and scalable fashion that reduces the total cost of ownership.