Benchmarking BGP Routers

Qiang Wu, Yong Liao, Tilman Wolf, and Lixin Gao
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA
{qwu,yliao,wolf,lgao}@ecs.umass.edu

***Abstract*— Determining which routes to use when forwarding traffic is one of the major processing tasks in the control plane of computer networks. We present a novel benchmark that evaluates the performance of the most commonly used Internet-wide routing protocol, the Border Gateway Protocol (BGP). Using this benchmark, we evaluate four different systems that implement BGP, including a uni-core and a dual-core workstation, an embedded network processor, and a commercial router. We present performance results for these systems under various loads of cross-traffic and explore the tradeoffs between different system architectures. Our observations help identify bottlenecks and limitations in current systems and can lead to next-generation router architectures that are better optimized for this important workload.**

I. Introduction

Computer networks provide data communication capabilities by connecting end-systems through links and routers. Routers send network traffic from link to link while ensuring that it eventually reaches its destination. Routers implement most of the complexity of networks and are conceptually divided into two domains: control plane and data plane. The data plane operates on network traffic, while the control plane manages the operations in the data plane. The data plane performance of both software-based and hardware-based routers has been extensively studied and evaluated. With the Internet continuing to increase in size and complexity, control plane performance of routers is becoming increasingly important [1].

Arguably the most important control plane functionality of routers is to determine what the most opportune path through the network is to reach any destination. This is determined through routing algorithms that provide the best "next hop" (i.e., neighboring router) for any destination in the network. Routing algorithms use information about nodes and links in the network, which is obtained by exchanging data with neighboring routers. To achieve scalability in large networks like the Internet, routing can be structured hierarchically: Networks are divided into autonomous systems (AS) (e.g., a corporate network or a Internet Service Provider network) and routing is done separately within an AS (intra-AS routing) and among ASes (inter-AS routing). Intra-AS routing is typically of lower complexity than inter-AS routing since it is constrained to a smaller number of nodes that are all administered by the same entity. Inter-AS routing, in contrast, is more complex as it involves all autonomous systems in the Internet (currently over 20,000) and all advertised IP destination prefixes (currently over 180,000) and is the topic of this paper.

Inter-AS routing needs to be implemented via a single protocol to ensure Internet-wide inter-operability. In the current Internet, the Border Gateway Protocol (BGP) [2] is the routing protocol that is used across all autonomous systems. A number of different criteria are used when computing a path to a destination including distance, local policies, and available alternatives. Determining the best path for all prefixes and processing the information exchanges with peer routers requires significant amounts of computational resources. It is therefore important to understand what processing performance is necessary for BGP routers and what commonly found system architectures perform best. Further, depending on the system design, there is potential for contention for processing resources between the data plan and the control plane. This interaction can have a significant performance impact on routing and forwarding capabilities of a router system.

In this paper, we study the performance of different BGP routers and measure their capabilities under different scenarios. Specifically, the contributions of our paper are:

- The design of a benchmark for BGP routers that exercises different workload scenarios based on different routing information exchanges that occur in a network. This benchmark is the basis for being able to generating repeatable performance measurements of different BGP router systems.
- A discussion of four router systems with different processor designs (uni-core router, dual-core router, network processor router, and commercial router) that represent a broad range of possible router implementations.
- A measurement study of processing performance of all four systems when exercised by the BGP benchmark. This performance is also measured for various levels of cross-traffic that exercises the forwarding capabilities of the router.

The remainder of the paper is organized as follows. Section II discusses related work. Section III introduces the BGP benchmark. The different router systems are presented in Section IV. Performance results are shown and discussed in Section V. Section VI summarizes and concludes this paper.

II. Related Work

Computer networks can use a number of different routing protocols. The most commonly used ones are the Border Gateway Protocol (BGP) [2] for inter-AS routing and Open Shortest Path First (OSPF) [3] and Routing Information Protocol (RIP) [4] for intra-AS routing. The OSPF protocol uses

link-state information to computer a shortest path spanning tree between the router and all destinations. The RIP protocol uses distance vector information to determine the shortest path to all destinations. Both OSPF and RIP use a single metric (hop counts or link weights) to determine the path traffic should take. In BGP, which explained in more detail in Section III, additional policy rules can be used determine routes. This feature increases the complexity significantly over OSPF and RIP.

The scale of the Internet and the complexity of BGP policies has lead to observations of BGP instabilities [5]. A direct implication is that routers need to continuously process BGP updates from their neighbors. Typically, BGP routers need to process in the order of 100 BGP messages per second. In case of network-wide events (e.g., worm attacks) the number of BGP messages can increase by 2-3 orders of magnitude [6]. If a router cannot handle these peak loads, it may not be able to send keep-alive messages to its neighbor and thus trigger additional events. It is therefore important to benchmark the peak number of BGP messages that a router can handle as we do in this paper. In related work, Agarwal et al. have explored the CPU utilization of BGP routers [7], but have fallen short of providing a comprehensive BGP benchmark and analyzing the performance of different router architectures in detail.

In the context of forwarding packets, routers need to determine to which neighbor traffic is supposed to be sent. In the Internet, IP addresses are allocated by prefixes as specified in Classless Inter-Domain Routing (CIDR) [8]. In order to determine to which destination prefix a packet belongs, address lookup algorithms are used. This lookup process can be the most processing intensive component of packet forwarding when routing tables are large. Ruiz-Sánchez et al. have compiled a survey of contemporary lookup algorithms [9]. With more diverse uses of the Internet, there is also a trend towards classifying packets by more than just their destination address (e.g., different service levels). This requires more complex flow classification algorithms [10].

III. BGP Benchmark

To put our BGP benchmark into context, we first provide a more detailed overview of BGP operation. We then describe the relevant performance metric for our benchmark and discuss eight operational scenarios that reflect a typical BGP workload.

A. BGP Overview

As described in Section I, BGP is used as the standard routing protocol between autonomous systems in the Internet. An autonomous system is a collection of IP networks and routers under the control of one entity (or sometimes more) that presents a common routing policy to the Internet. To exchange routing information between different ASes, the Border Gateway Protocol [2] is the only deployed inter-domain routing protocol. BGP is a path-vector routing protocol and works in an incremental manner. That is, when two BGP speakers (i.e., two routers that implement BGP) connect to each other, they first exchange all their routing information with each other. After that, routing updates are sent only upon changes in network topology or routing policy.

There are two kinds of routing update message used in BGP: *announcement* and *withdrawal*. A route announcement indicates that a router has either learned how to reach a new destination network or made a policy decision of preferring another route to a destination network. The destination network is represented by an IP prefix. Route withdrawals are sent when a BGP speaker makes a new local decision that a network is no longer reachable via any path. BGP limits the distribution of a router's reachability information to its neighbor BGP routers only. Each route is attached with an AS path indicating the AS sequences along which the route passes through. After a BGP speaker detects a routing change by receiving update messages from neighbors, the BGP speaker needs to re-select the new "best" path based on what it has learned from the update messages. The route selection in BGP can be very complicated and it is always policy-based [11], which depends on the commercial relationship between different domains. If the AS relationship is not considered, most vendors implement the best path selection based on the length of AS path [12], although it is not specified in the BGP RFC document [2].

In a typical BGP implementation, all the routes are stored in Routing Information Bases (RIBs) in the memory of the BGP speaker. There are three RIBs in BGP, namely, the *Adj-RIBs-In*, the *Loc-RIB*, and the *Adj-RIBs-Out*, as defined in [2]. Routes that will be advertised to other BGP speakers must be present in the *Adj-RIB-Out*; routes that will be used by the local BGP speaker must be present in the *Loc-RIB*; and routes that are received from other BGP speakers are present in the *Adj-RIBs-In*. For a given prefix, the router's BGP decision process computes the most preferred route to that prefix from all the *Adj-RIB-In*s and stores it in the *Loc-RIB*. The decision process then determines what subset of the *Loc-RIB* should be advertised to each neighbor and that subset is stored in a per-neighbor database, the *Adj-RIBs-Out*. In summary, the *Adj-RIBs-In* contain unprocessed routing information that has been advertised to the local BGP speaker by its neighbors; the *Loc-RIB* contains the routes that have been selected by the local BGP speaker's decision process; and the *Adj-RIBs-Out* organize the routes for advertisement to specific neighbors by means of the local speaker's announcement/withdrawal messages.

It is important to note that *Loc-RIB* is different from the forwarding table used by the router's forwarding engine. The *Loc-RIB* table is a structure maintained by BGP routing software and the forwarding table is co-located with the forwarding software in the operating system kernel or in hardware. Once the BGP process updates the *Loc-RIB* table, it needs to update the forwarding table used by the router forwarding engine accordingly.

B. BGP Benchmarking Methodology

In order to create a realistic operation environment for a BGP router, we need to use at least two BGP neighbors that interact with the router under test. One neighbor can create

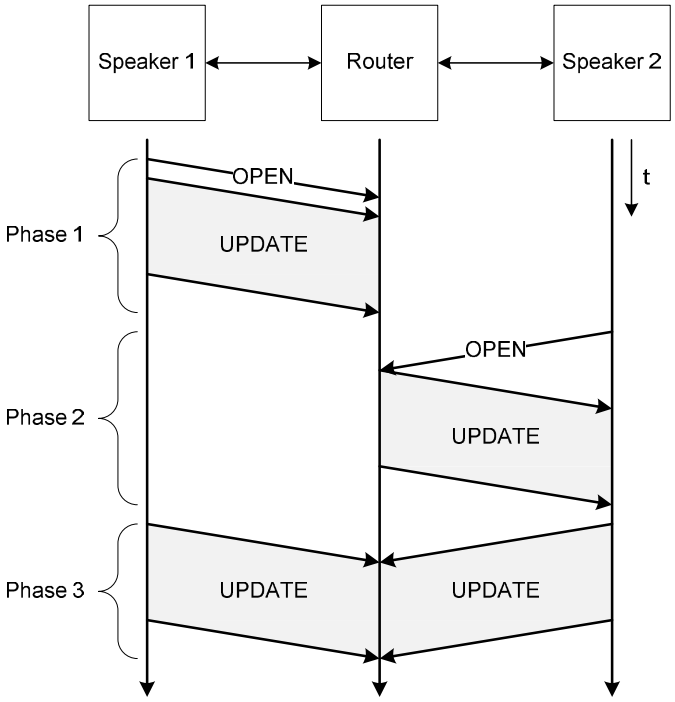


Fig. 1. BGP Benchmark Setup. Two BGP speakers are used to generate BGP processing workloads on the router under test.

BGP announcements to test the ability of the router to process messages. The second neighbor is necessary to test the ability of the router forward message to other BGP routers.

Figure 1 illustrates the experimental methodology that we have chosen for our benchmark. The router under test is connected to two BGP speakers. The space-time diagram also shows the order of BGP interactions between all systems. Responses to OPEN, NOTIFICATION, and KEEP-ALIVE messages are not shown for clarity. Also UPDATE messages sent by the router in Phase 3 are not shown. In Phase 1, Speaker 1 sends announcements to the router, which is assumed to start with empty RIBs. In Phase 2, after processing the routes received by Speaker 1, the router passes this information on to Speaker 2. In Phase 3, Speaker 1 or 2 send incremental updates to the router. This setup allows us to test all possible BGP interactions. In some cases, a phase can be omitted. For example, to test the start-up processing of a router, only Phase 1 is necessary.

C. Performance Metrics

From the above discussion of BGP operation, we see that the essential operation of a BGP speaker is computing the *Loc-RIB* table according to the messages it receives from neighbors. BGP activities can have a significant impact on routers in an operational ISP network. Existing measurement work [7] shows that BGP processes consume over 60% of a router's non-idle CPU cycles. Thus, the interesting metric in evaluating the performance of a router is how fast the router can process BGP messages, including announcement and withdrawal messages. We therefore use a metric of *transactions per second* to quantify BGP processing performance. To distinguish between different types of messages, we define a number of benchmark scenarios that consider these differences.

D. Benchmark Scenarios

In order to benchmark the performance of a BGP router, we create difference scenarios of BGP interactions that require processing. To make the benchmark representative and relevant, the scenarios relate to workload scenarios that occur in operational networks. For example, in one scenario a BGP speaker injects a large number of announcement messages. This experiment represents the situation where a router is just powered up and needs to learn routes from neighboring routers as fast as possible. Similarly, we can measure how fast the tested router can response to withdrawal messages that invalidate previously announced routes. This experiment represents the situation where a link is down or another router has failed. We also consider a set of experiments to test a router's response to BGP's incremental activities, where routes are announced from two different BGP neighbors and the router has to choose which ones it prefers. This experiments represents a BGP router's working state during most of the time. In addition to different BGP operations, it is also important to consider if update messages are sent in individual messages (i.e., small packets) or are clustered into larger packets (i.e., large packets).

Our BGP benchmark considers all these cases in eight different scenarios. The first four scenarios focus on the router's response in the BGP start-up phase and ending phase, where large numbers of announcement and withdrawal messages are processed. The other four scenarios test the router's response to incremental BGP updates. The eight scenarios considered in our BGP benchmark are listed in Table I. The following explains each scenario in more detail. Scenarios that differ only by packet size are explained together. In the case of small packet sizes, a single UPDATE is contained in a packet. In the case of large packets, 500 prefixes are contained in an UPDATE message.

1) Scenarios 1 and 2: These scenarios test the speed at which a router can update the *Loc-RIB* table and the forwarding table. Speaker 1 injects a large routing table into the tested router in Phase 1. An announcement message is used in this scenario and all prefixes are added to both the *Loc-RIB* table and the forwarding table of the router.

2) Scenarios 3 and 4: These scenarios test the speed at which the router can process route withdrawals. Initially, Speaker 1 injects a large routing table into the tested router in Phase 1. After the router finishes processing all the BGP update messages in Phase 1, Phase 2 is omitted and Speaker 1 sends withdrawal messages for each prefix it previously injected into the router in Phase 3. Therefore, all added prefixes are removed. Both the *Loc-RIB* table and the forwarding table of the router are updated in the process.

3) Scenarios 5 and 6: These scenarios test the speed at which the router can process announcement messages that do not alter the forwarding table. Speaker 1 injects a large

TABLE I
BGP BENCHMARK SCENARIOS.

BGP Operation	Start-Up		Ending		Incremental Operation			
UPDATE Message Type	ANNOUNCE		WITHDRAW		ANNOUNCE			
Forwarding Table Changes	Yes				No		Yes	
Packet Size	Small	Large	Small	Large	Small	Large	Small	Large
Scenario Number	1	2	3	4	5	6	7	8

routing table into the router in Phase 1. After the router finishes processing all the prefixes and changing the forwarding table, Speaker 2 establishes a connection with the router. The router transfers its current route information to Speaker 2 in Phase 2. In Phase 3, which is the relevant phase for these scenarios, Speaker 2 sends the router the same prefixes as the Speaker 1 sent previously, but with a longer AS PATH. The BGP update messages sent by Speaker 2 thus do not affect the forwarding table in router. But the router still needs to check every prefix and make a decision on whether or not it should update the forwarding table.

4) Scenarios 7 and 8: These scenarios test the speed at which the router can process announcement messages that alter the forwarding table. The setup for these scenarios is equivalent to Scenarios 5 and 6. The only difference is that routes announced by Speaker 2 in Phase 3 have a shorter AS PATH. Thus, the router needs to update the forwarding table when it replaces the routes announced by Speaker 1 with those announced by Speaker 2.

When calculating the transactions per second that a router can achieve under different scenarios, only the appropriate phase of the benchmark scenario is considered. For example, only Phase 3 is relevant in Scenarios 5–8. Time spent setting up the scenario in Phase 1 and 2 is not considered to evaluate the performance of the router for these tests.

IV. BGP ROUTER SYSTEMS

Before delving into measurement results in Section V, we discuss the hardware and software components of typical BGP router systems. Understanding the different system designs and interactions between control and data path is important to interpret our measurement results. While we discuss specific system architectures, configurations, and software in this Section, it is important to note that the BGP benchmark discussed in the previous section is applicable to *any* BGP router.

A. Hardware

BGP routers have been implemented in a number of different ways. To benchmark the performance of BGP routers, we need to consider different system architectures. In this work, we explore four system designs that are representative of a large number of existing BGP router systems. Figure 2 shows the block diagrams of these four systems. The data path that traverses the packet forwarding component is shown in red. The control path that interacts with the BGP routing component is shown in green. Depending on the type of processor used for BGP processing and depending on the interactions between control and data path on shared resources, difference performance trends can be observed.

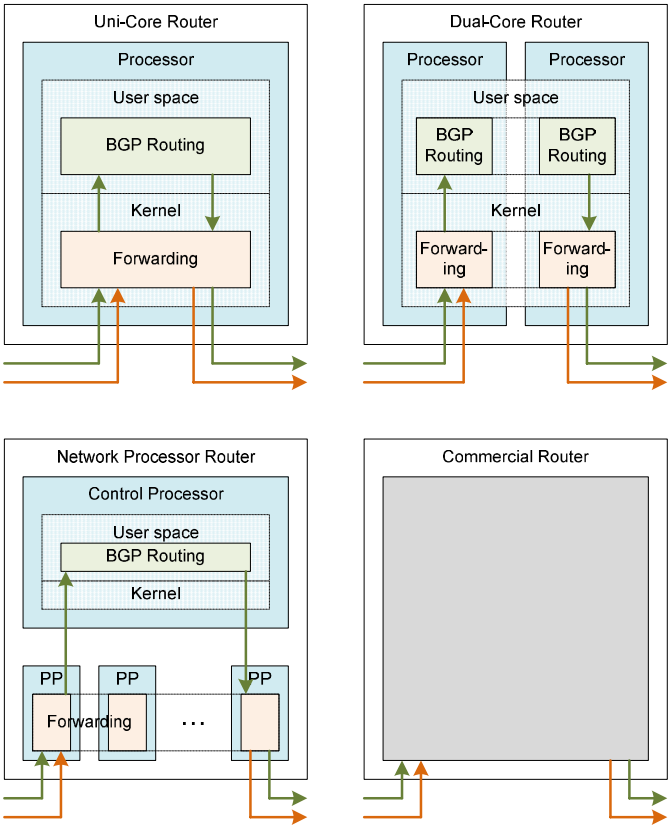


Fig. 2. Block Diagram of BGP Router Systems (PP = packet processor). The data path is shown in red, the control path in green.

1) Uni-Core Router: The uni-core router is a typical software-based router system that uses commodity workstation hardware. Multiple network cards connect through a PCI (or PCI-X or PCI Express) bus to a single processor. A commodity operating system is used to host BGP processing and data path forwarding. Forwarding is typically implemented in the operating system kernel whereas routing computations are performed in user space.

This simple and straightforward design is often used for network gateways that handle data rates in the order of hundreds of Mbps. The general-purpose processing capabilities make it possible to deploy additional network services on this router (e.g., firewall [13], network address translation (NAT) [14], intrusion detection [15]).

The main drawback of this design is that forwarding and BGP processing share the same processing hardware. While

TABLE II
SYSTEM CONFIGURATIONS OF BGP ROUTERS TESTED.

Name	System Type	Hardware		Software	
		Processor	Memory	Operating System	BGP Software
Pentium III	Uni-core router	Intel Pentium III (800MHz)	256MB	Linux 2.6.18	XORP 1.3
Xeon	Dual-core router	Dual-Core Intel Xeon (3.0GHz)	2GB	Linux 2.6.18	XORP 1.3
IXP2400	Network processor router	Intel IXP2400 (XScale processor (600MHz))	256MB	Linux 2.4.18	XORP 1.3
Cisco	Commercial router	Cisco 3620		IOS 12.1(5)YB	

the operating system can provide some sort of isolation, interference between complex BGP computations and packet forwarding are unavoidable.

2) Dual-Core Router: The dual-core system differs from the uni-core router insofar that it has two physical processor cores. This architectures represents the increasingly widely deployed class of multicore systems. The operating system schedules both kernel and user space processing among the processor cores.

While multi-core systems are generally higher-performing than uni-core systems, they are susceptible to the same problem where BGP processing can interfere with forwarding. However, the effect are mediated when competing processes are placed on separate processor cores.

3) Network Processor Router: A router architecture where BGP routing and forwarding is placed on separate and independent hardware components is a so-called "network processor" [16]. A network processor uses a number of simple packet processors to perform the simple and highly parallelizable task of packet forwarding. Routing and other complex control processing tasks are computed on the control processor. All these processors are co-located with memory and I/O components on a single system-on-a-chip.

Current network-processor-based routers do not use a runtime environment for managing packet processors. Instead, processing tasks are statically allocated. On the control processor, a conventional embedded operating system with kernel space and user space is used. There, the BGP routing processing is performed in user space, similar to the uni-core router.

4) Commercial Router: Commercial router designs are often proprietary and may contain custom system components (e.g., ASICs). We view a commercial router system as a black box without any detailed insight on how BGP processing or forwarding is performed.

B. Software

The software that is used on router systems for BGP processing and forwarding can be different but has to adhere to minimum standards defined by RFCs 4271 [2] and 1812 [17] respectively.

1) BGP Routing: An implementation of BGP routing that is widely used in the research community is available through the eXtensible Open Router Platform (XORP) [18]. This software-based router platform implements a number of different router functions including BGP processing. Since XORP is available under a BSD-style license and portable to different systems, it is possible to use it on the uni-core router, dual-core router, and network processor router.

2) Forwarding: The forwarding component of router systems is typically implemented in the operating system kernel or on specialized packet processors or ASICs. The processing tasks necessary to perform RFC-1812-compliant forwarding include checking the Internet Protocol (IP) header checksum, decrementing the time-to-live (TTL) field (and discarding the packet if the TTL has reached zero), and updating the checksum accordingly. Further, it is necessary to perform a lookup of the destination IP address in the forwarding information base (FIB) that is managed by the BGP routing software. Due to the importance of high-bandwidth, low-delay forwarding, many operating system kernels have been tuned to make this process as fast as possible (e.g., direct memory access (DMA) from the network interface card, specialized instruction sets in network processors, custom ASICs in commercial routers).

C. System Configurations

To present the results of our BGP benchmarking study in Section V, we use four specific system configurations that reflect different hardware architectures and software implementations discussed above. The configurations of these four systems are listed in Table II.

The uni-core router is an older Intel Pentium III processor with small amounts of memory. It uses Linksys EG1032v3 PCI32 Gigabit network cards for data plane forwarding. This configuration represents lower-end software-based router systems. The dual-core router uses a high-end Intel dual-core Xeon processor system with hyper-threading (two threads per core). It uses two PCI Express gigabit network cards, a Broadcom NetXtreme BCM5752 and a Intel 82545GM, for forwarding. This configuration represent a high-end commodity system. The network processor router uses an Intel IXP2400 network processor [19] that consists of eight packet processors for forwarding and an embedded XScale processor for BGP processing. The network processor is implemented on a Radisys ENP-2611 prototype board with three Gigabit network interfaces. This configuration represents a router with a high-end data path and a low-end control processor. The commercial system is a Cisco 3620 router. No details on the hardware configuration is available. Unlike the other routers that use XORP 1.3 for BGP processing, the Cisco system uses IOS 12.1.

V. MEASUREMENT RESULTS

We evaluate the performance of the BGP router systems discussed in the previous section in two different contexts.

TABLE III
BGP PERFORMANCE WITHOUT CROSS-TRAFFIC IN TRANSACTIONS PER SECOND.

	System			
	Pentium III	Xeon	IXP2400	Cisco
Scenario 1	185.2	2105.3	24.1	10.7
Scenario 2	312.5	2247.2	36.4	2492.9
Scenario 3	204.1	2898.6	26.7	10.4
Scenario 4	344.8	1941.7	43.5	2927.5
Scenario 5	1111.1	3389.8	85.7	10.9
Scenario 6	3636.4	10000.0	230.8	3332.3
Scenario 7	116.6	784.3	11.6	10.7
Scenario 8	118.7	673.4	14.9	2445.2

First, we evaluate system performance in a setup where BGP processing is the only active task on a router. Second, we augment the workload by sending network traffic to the router for forwarding. This cross-traffic shows interactions between control and data plane.

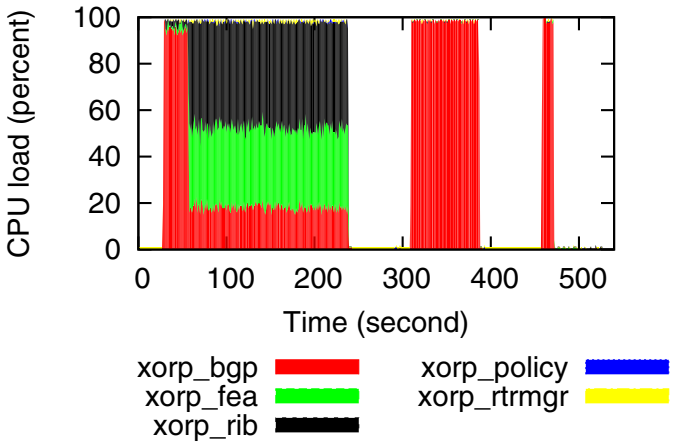
A. BGP Processing Performance

Figure 3 illustrates how the processing phases of a benchmark scenario are reflected in the CPU load of the Pentium III, Xeon (loads for all threads are added), and IXP2400 system. In this case, Scenario 6 is shown. The XORP software uses five processes that become active at different stages of the benchmark. On the uni-core router, all processes compete for the same processor. Thus, the processor is clearly the bottleneck in the system. On the dual-core router, processes are distributed across two cores and two threads each. In this case, the CPU only becomes a bottleneck when one process requires more processing power than a single CPU can provide (as is the case in the beginning of Phase 1 and during Phases 2 and 3 in Figure 3). On the IXP2400, similar trends as on the Pentium III can be observed. The CPU is a bottleneck for all phases of the benchmark scenario. In addition, it can be seen that the router manager process (*xorp_rtrmgr*) uses a considerable amount of processing time. This process is hardly visible on the Pentium III router and the Xeon router. However, on the underpowered XScale it becomes a considerable component of the total workload.

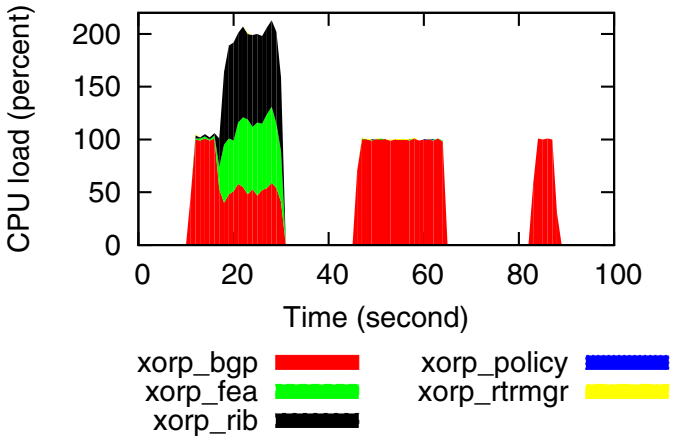
It can also be observed in Figure 3 that the processing times shown on the x-axes are very different for these three systems. The Xeon completes all phases in less than 90 seconds whereas the IXP2400 requires more than half an hour. To quantify the processing performance in more detail, we show the number of transactions per second performed by each router system for all benchmark scenarios in Table III.

The results in Table III lead to several observations:

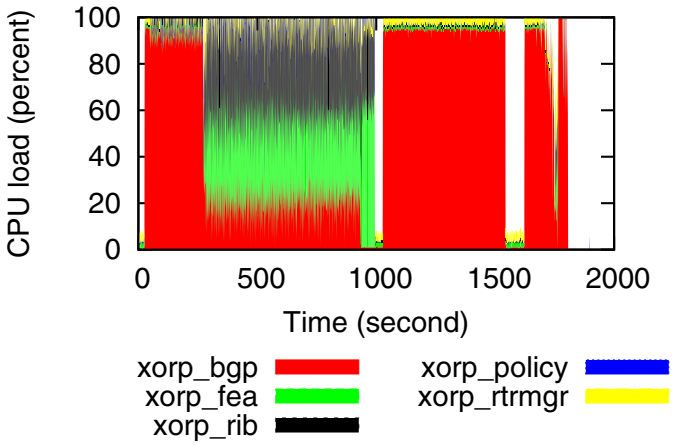
- The dual-core router achieves the highest number of transactions per second for most benchmarks. The commercial system outperforms the dual-core system only in scenarios 2, 4, and 8.
- Between the dual-core router and the uni-core router a performance drop of roughly one order of magnitude can be observed. A similar performance drop occurs between the uni-core router and the network processor router.



(a) Pentium III



(b) Xeon



(c) IXP2400

Fig. 3. Activity of Different BGP Processes During Scenario 6.

This is expected as it tracks the approximate performance differences between the Xeon, Pentium III, and XScale system.

- Scenarios that do not change the forwarding table are processed faster than those that require changes. Beside route computation, changing the forwarding tables involves a large amount of other operations (e.g., memory read and write, inter-process communication).
- Packet size has significant impact on BGP update message processing time. Benchmarks scenarios with large packets achieve a higher transaction per second rate than those which use small packets. Although packet size does not affect the operation of changing forwarding table, it does affect how router process BGP update messages. Figure 4 shows a comparison of Scenario 1 (small packets) and Scenario 2 (large packets) on the Pentium III system. For the scenario with small packets, we observed that *xorp_bgp*, *xorp_fea* and *xorp_rib* compete for CPU resource for the entire measurement phase. For scenarios with large sized packets, *xorp_bgp* first runs a certain period of time. Then *xorp_fea* and *xorp_rib* begin to compete for the CPU.
- In scenarios with small packets, the commercial system performs worse than the network processor router.

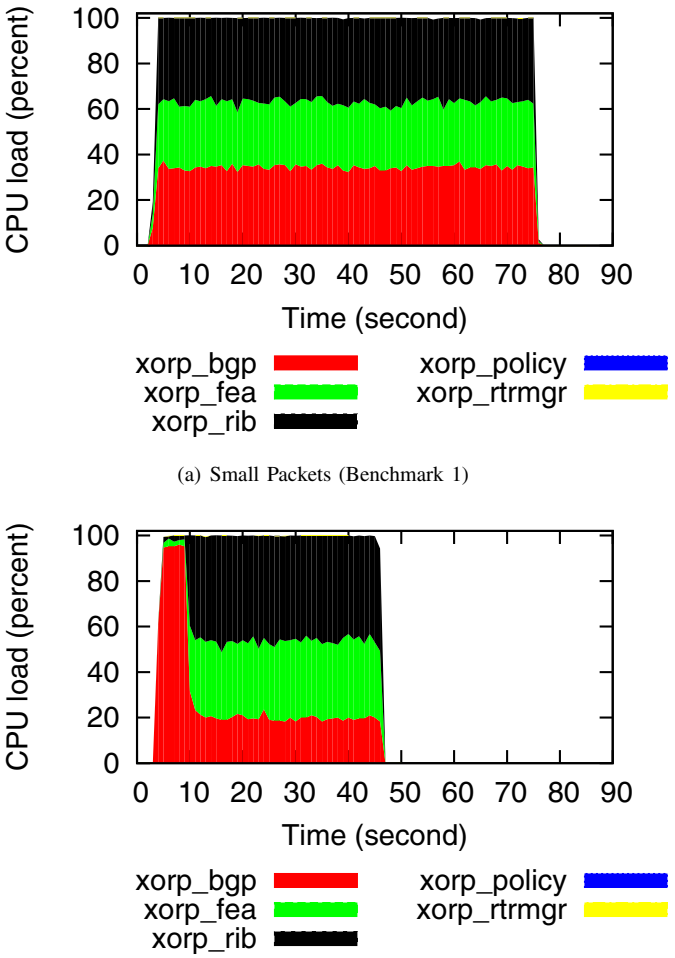
This data shows that the XORP implementation of BGP routing achieves a performance that is comparable to that of a commercial router for large packets. But it also shows there are significant differences in performance between different router implementations. This difference further increases when considering cross-traffic.

B. BGP Processing Performance with Cross-Traffic

Neither the uni-core router nor the dual-core router differentiate processing resources for control path and data path. Since forwarding traffic is handled by the Linux kernel, and the BGP routing program runs in user space, cross-traffic is given higher priority by the operating system. Therefore, the rate at which cross-traffic is injected into the router has a significant impact on BGP processing performance.

To explore this issue, Figure 5 shows the BGP processing performance in transactions per second for all benchmark scenarios and all router systems for an increasing level of cross-traffic. Each of the router system has a different limit for the maximum data rate that it can forward (Pentium III: 315Mbps due to PCI bus limitations; Xeon: 784Mbps due to PCI Express bus limitations; IXP2400: 940Mbps due to network interconnect limitations; Cisco: 78Mbps due to 100Mbps router ports). The BGP processing performance can only be measured for cross-traffic rates below this rate. For a cross-traffic of 0Mbps, the performance values correspond to the values shown in Table III. We can make the following observations:

- As cross-traffic increases, the BGP processing performance drops for all systems except the network processor router. The network processor router uses completely independent processing resources for forwarding and thus can achieve the same BGP processing performance on the XScale processor for 1Gbps of cross-traffic as it does for no forwarding load.
- On the Pentium III and Xeon systems, the BGP processing performance drops gradually with increasing cross-traffic.
- On the Cisco system, practically no degradation can be observed for the already low processing rate on small packets. For large packets, the processing rate drops drastically (note the logarithmic scale on the y-axis) as cross-traffic approaches 100Mbps.



(a) Small Packets (Benchmark 1)

(b) Large Packets (Benchmark 2)

Fig. 4. CPU Load of Pentium III with Small and Large Packets.

To illustrate the cause for the decrease in processing performance on the Pentium III system (and similarly on the Xeon system), we show the CPU load for benchmark Scenario 8 in Figure 6. The figure shows the scenario without any cross-traffic (Figure 6(a)) and with 300Mbps of cross-traffic (Figure 6(b)). It can be observed that cross-traffic causes a considerable increase in interrupt processing (totaling 20–30% of CPU load) caused by packets arriving on the receiving network interface. This reduces the available CPU time for

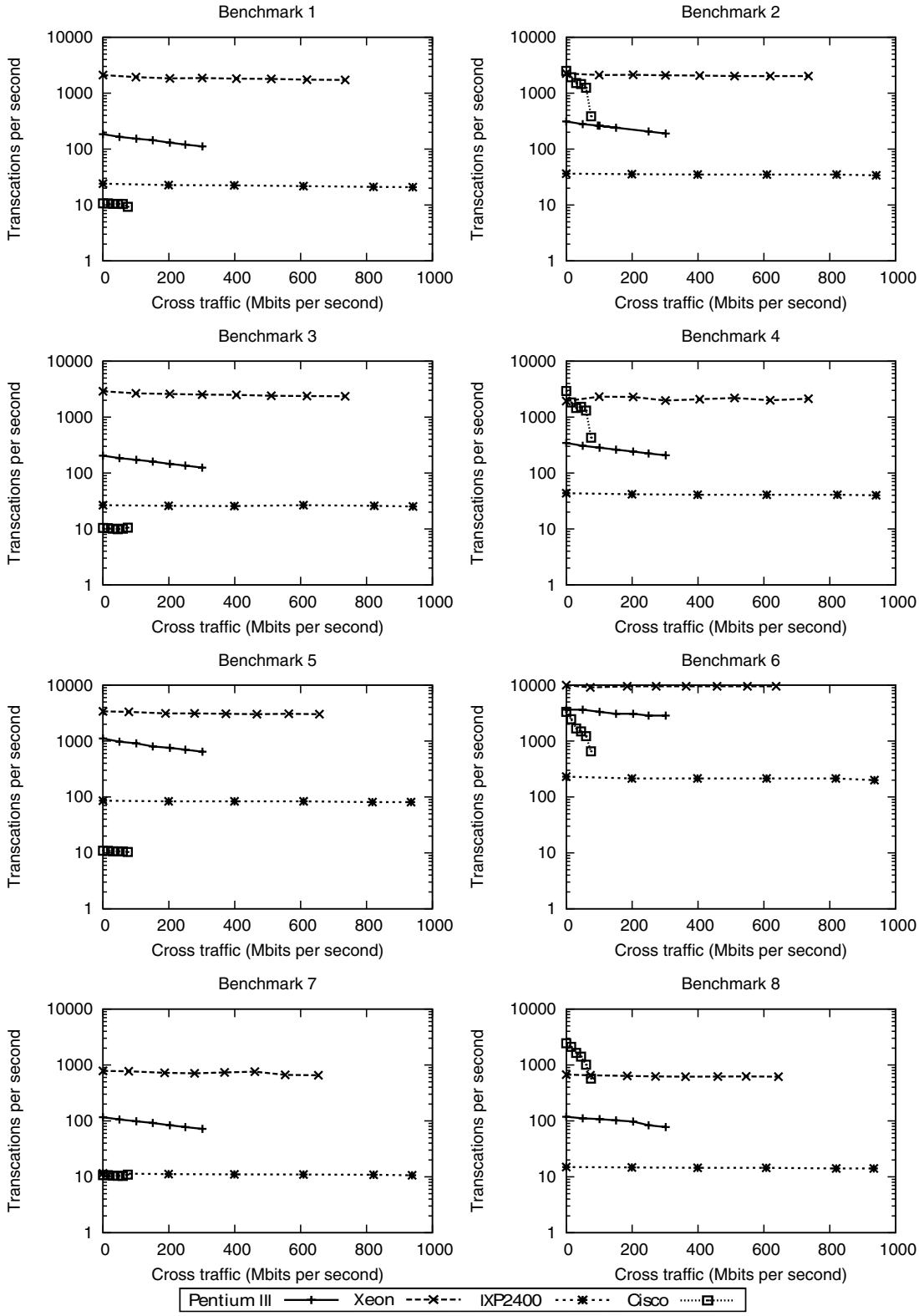
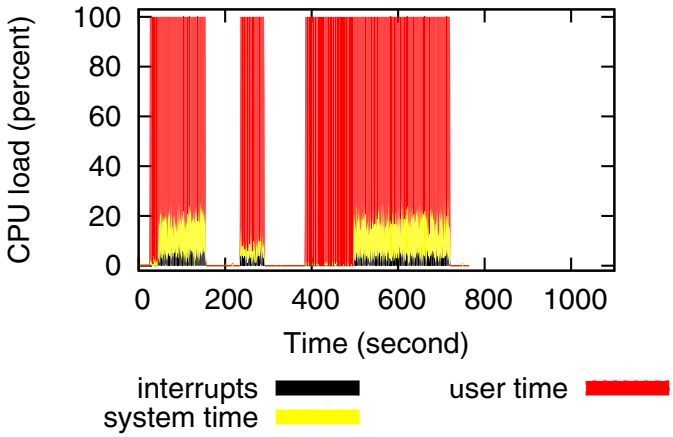
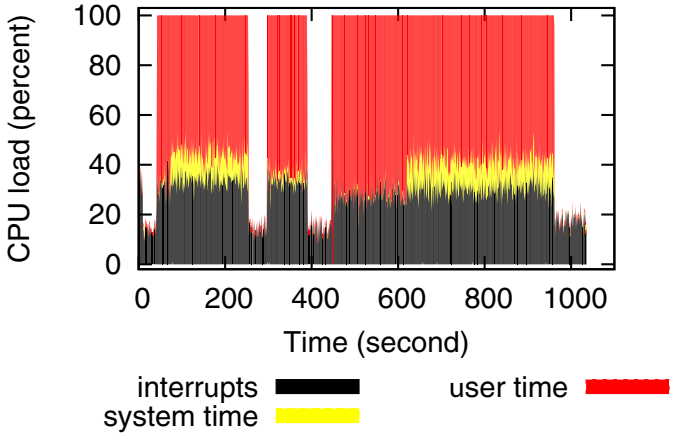


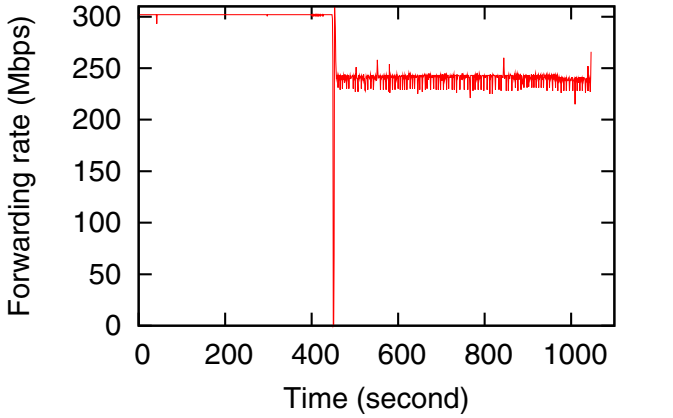
Fig. 5. BGP Performance for Different Benchmarks and Cross-Traffic Loads.



(a) CPU Load without Cross-Traffic



(b) CPU Load with 300 Mbps of Cross-Traffic



(c) Forwarding Rate with 300Mbps of Cross-Traffic

Fig. 6. CPU Load on Pentium III during Benchmark 8.

BGP processing and thus extends the time it takes to complete the benchmark scenario.

Interestingly, the cross-traffic load on the CPU also has an impact on the forwarding performance of the system. Despite the higher priority given to forwarding processing in the kernel, BGP processing may cause packet loss for some scenarios where a large number of prefixes is injected into the routing table. Figure 6(c) shows the rate at which forwarded traffic leaves the router (x-axis matches that of Figure 6(b)). Shortly after the start of Phase 3, the forwarding rate decreases.

C. Implications for Router Design

The measurement results above provide us with a comprehensive view of BGP processing requirement and how different router architectures perform in handling this workload. From our work, we can extract several implications with regards to router design:

- Simple embedded processors as used on network processor routers are insufficient to process a typical BGP load in the order of hundreds of messages per second.
- High-performance dual-core systems can handle typical BGP workloads and may be able to handle some network events that cause higher BGP traffic. However, no system can handle more than 10,000 messages and thus no system can handle the workload generated by routing updates due to a worm spreading in the Internet [6].
- Shared processing resources for data plane and control plane can cause interference between them. Even a full-blown operating system cannot avoid reduced BGP performance and forwarding packet loss under load. For high-performance routers, it is therefore imperative to use different processing resources for control and data plane.

Similarly, we can find implications for BGP operational issues:

- It is important to aggregate update messages into large packets to obtain best BGP processing performance by eliminating per-packet overheads.
- BGP implementations that use multiple processes perform better on multi-core platforms. With increasing numbers of available processor cores in workstation and embedded systems, it is imperative to continue designing BGP implementations that are highly parallelizable.

Another important aspect of router design that is beyond the scope of this paper is the limitation of power that is available for control processing. Due to limits on power supplies and cooling capacities, routers cannot consume an arbitrary amount of power. Clearly, a dual-core Xeon processor consumes a large amount of processing power that would not be available to perform data path processing. An interesting tradeoff is how much power should be dedicated to the control plane and how much to the data plane to obtain a balanced and versatile router design.

VI. Summary and Conclusions

BGP routing is an essential feature of routers that connect autonomous systems in the Internet. We present a benchmark

that considers all practical BGP workload cases in eight scenarios. Using four router system architectures that differ in their implementation of control and data plane, we compare their BGP processing performance. We quantify the number of transactions per second that each system can process in the best case as well as under forwarding load. While we observe that systems with high-end processors perform better, we also see that systems with shared data path and control path show interference that decreases BGP and forwarding performance. We conclude with several observations on how to improve router design by considering BGP processing an essential part of the workload of the entire router system.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0447873.

REFERENCES

[1] M. Beesley, "Router/switch control plane software challenges," *Keynote Presentation at ACM/IEEE Symposium on Architectures for Networking and Communication Systems (ANCS)*, San Jose, CA, Dec. 2006.

[2] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," Network Working Group, RFC 4271, Jan. 2006.

[3] J. Moy, "OSPF version 2," Network Working Group, RFC 1247, July 1991.

[4] C. Hedrick, "Routing information protocol," Network Working Group, RFC 1058, June 1988.

[5] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions Networking*, vol. 6, no. 5, pp. 515–528, Oct. 1998.

[6] J. Cowie, A. T. Ogielski, B. J. Premore, and Y. Yuan, "Internet worms and global routing instabilities," in *Proc. of ITCom 2002 (Scalability and Traffic Control in IP Networks II)*, Boston, MA, July 2002, pp. 195–199.

[7] S. Agarwal, C.-N. Chuah, S. Bhattacharrya, and C. Diot, "Impact of BGP dynamics on router CPU utilization," in *Proc. of Passive and Active Measurement Workshop (PAM)*, Antibes Juan-les-Pins, France, Apr. 2004.

[8] V. Fuller, T. Li, J. Y. Yu, and K. Varadhan, "Classless inter-domain routing (CIDR): an address assignment and aggregation strategy," Network Working Group, RFC 1519, Sept. 1993.

[9] M. A. Ruiz-Sánchez, E. W. Biersack, and W. Dabbous, "Survey and taxonomy of IP address lookup algorithms," *IEEE Network*, vol. 15, no. 2, pp. 8–23, Mar. 2001.

[10] P. Gupta and N. McKeown, "Algorithms for packet classification," *IEEE Network*, vol. 15, no. 2, pp. 24–32, Mar. 2001.

[11] L. Gao and J. Rexford, "Stable internet routing without global coordination," in *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Santa Clara, CA, June 2000, pp. 307–317.

[12] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of internet policy and topology on delayed routing convergence," in *Proc. of the Twentieth IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, Apr. 2001, pp. 537–546.

[13] J. C. Mogul, "Simple and flexible datagram access controls for UNIX-based gateways," in *USENIX Conference Proceedings*, Baltimore, MD, June 1989, pp. 203–221.

[14] P. V. Mockapetris and K. J. Dunlap, "Development of the domain name system," *SIGCOMM Computer Communication Review*, vol. 25, no. 1, pp. 112–122, Jan. 1995.

[15] *The Open Source Network Intrusion Detection System*, Snort, 2004, http://www.snort.org.

[16] T. Wolf, "Challenges and applications for network-processor-based programmable routers," in *Proc. of IEEE Sarnoff Symposium*, Princeton, NJ, Mar. 2006.

[17] F. Baker, "Requirements for IP version 4 routers," Network Working Group, RFC 1812, June 1995.

[18] M. Handley, O. Hodson, and E. Kohler, "XORP: An open platform for network research," in *Proc. of First Workshop on Hot Topics in Networking*, Princeton, NJ, Oct. 2002.

[19] *Intel Second Generation Network Processor*, Intel Corporation, 2002, http://www.intel.com/design/network/products/npfamily/ixp2400.htm.