Advanced Routing Concept Performance Analysis

Marija Kalendar, *Member, IEEE*, Danijela Jakimovska, *Student Member, IEEE*, Aristotel Tentov, *Member, IEEE,* Goce Dokoski

***Abstract* — Today's world of modern technologies and high communication speeds demands researching new ways of augmenting networking hardware and software routing performance. This paper proposes combining several concepts in a complete routing system. First, a modification to the widely used routing process is proposed, such that a complete list of all Internet organization domain routers is kept enabling computing the entire packet path to the destination. Further, novel network processor architecture is used to support the proposed routing process. Finally, performance characteristics of the proposed system are evaluated.**

***Keywords* — modified advanced routing, network processor, next generation networks, routing protocol, source routing**

I. Introduction

Concidering the recently more vigorous transition from circuit to packet switch networks, enormous increase of network traffic [1] can be observed, and it is expected that Internet throughput would increase to 1 Tb/s soon. Thus, it is essential to research new ways of augmenting routing performances for achieving respectable speeds. Achieving higher processing speeds can not be expected only by modifying and improving hardware network components. This is why modifying the entire process of network routing has to be considered.

The routing process in contemporary Internet exploits a hierarchical concept of keeping routing information, optimized for packet switching process of data exchange. Based on several research projects [2], and predictions for technology advancement, the decision for using such concept was made as far as the 1980's. The main reason for adopting the hierarchical solution was based on the especially high memory cost, and, the huge amount of memory capacity needed [2]. Due to the substantially changed situation in the area of memory technologies and memory cost, we believe that the former concept for keeping the information for every network node in the early ARPANET IMP's, [2], should be re-investigated.

Hence, in this paper we propose reorganizing the traditional routing process by possibly re-inventing two concepts: memorizing the complete Internet routing topology (all Internet organization domain level routers), and source routing; as well as supporting the process by a novel 64–bit RISC based network processor architecture. This novel 64-bit RISC (NET RISC) architecture is explained in more detail in [3].

The rest of this paper is organized as follows: Sections II and III explain the proposed routing information memory, and routing mechanism modifications. Afterwards, in section IV, some performance evaluation of the new approach is presented. Section V concludes the paper.

II. Proposed Routing Information Memory and Processor Architectures

Achieving high performance routing in the routers today is strictly dependent on incoming packet processing tasks. The most time consuming processing tasks in network processors are: classifying, and searching [1]; and in order to be able to compete for achieving 10/100 Gb/s packet processing, these tasks have to be revised, and the routing process should simplified as much as possible.

According to [4], simpler packet processing and higher speeds can be achieved if the most time–consuming processing operations are considered, and some appropriate choices of protocol functionalities are made. Thus, many different techniques for speeding up packet processing have been proposed. Some slow operations, as CRC calculation, can be implemented as hardware functional blocks. Faster table lookup algorithms have been proposed, as well ([5]-[6]). Additional improvement can be achieved if the network device avoids table lookup operations during packet processing. For that purpose, in [4] employing source routing is mentioned as a possibility.

Resulting from the current trends of fast development in memory production technologies, increase of its throughput and speed, and significant decrease of memory cost, today it is a standard to have multi GB of computer RAM memory. Hence, it becomes feasible to incorporate such amount of memory within routers themselves. Due to

Marija Kalendar, Msc, is teaching assistant and Ph.D. student at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia (e-mail: marijaka@feit.ukim.edu.mk).

Danijela Jakimovska, Msc, is teaching and research assistant at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia (e-mail: danijela@feit.ukim.edu.mk).

Aristotel Tentov, Phd, is professor at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia (e-mail: toto@feit.ukim.edu.mk).

Goce Dokoski, Msc, Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia (e-mail: goce.dokoski@feit.ukim.edu.mk).

increased memory throughput and appropriate memory response times, it is very realistic to expect acceptable time response characteristics from memories that have huge amount of stored data.

According to [7], in March 2010 there were near 210 million servers on Internet, and according to [8] in April 2010 around 321604 routes exist in default-free zone (DFZ) routers routing tables. Conceptually, DFZ routers have a "complete" BGP table. With such number of servers and routes advertised in the Internet, and obviously smaller number of routers that facilitate the exchange of data among them, the idea that all possible routes to Internet routers can reside in a router's memory becomes visible. In this paper, we propose the routing tables to encompass all Internet routers on organization domain level (below national domain routers if speaking in DNS terms). Furthermore, memory size is not likely to exceed 16 GB, which is not too expensive for current routers architectures, and sizes. Consequently, it becomes feasible to memorize the complete Internet router topology on organization domain level within each organization domain router. This is equivalent to routing information in ARPANET IMP's [2], in the early days of computer networking.

If the complete topology of all related Internet organization domain routers is being stored in each organization domain edge router, the preferred complete path for every arriving packet could be decided at the entry point into the network. Thus, the preferred network path to the destination could be easily incorporated in IPv4/Ipv6 packet headers. The routers on the preferred network path can read such headers, and according to the provided information, simply put the packet on the corresponding output interface. If the designated output interface is unreachable (traffic congestion, collapse of neighbor router/output link), a router can relatively easy choose another output interface, recalculate preferred network path to destination, make adequate changes in IP packet header, and proceed with packet delivery.

To further speed up the routing process the novel network processor architecture (64-bit RISC Harvard processor architecture) presented in [3] is used as well.

III. Routing Data Organization

The idea behind the proposed memory organization can be summarized to: "storing great amount of routing data in most routers"; actually, keeping "shortest paths" to almost all "more significant" Internet routers. In order to pursue the idea, some modifications of routing data storing principles, and some new data structures would be needed inside routers themselves.

First, to shorten the search process, and minimize shortest path data storing, we propose defining a globally unique identification number for all routers (router ID). If we estimate the number of all organization domain routers on the Internet for this proposed routing protocol to around 2^{20} (quite a large number considering the number of routes in DFZ [8]), only 20 bits would be needed for assigning a unique router ID instead of currently used 32 bits for router identification. Hence, 20 bits would be used for storing Router IDs in the routing process.

Furthermore, modifying the important data tables of the routing process is proposed:

- the traditional Routing Table would now hold the complete shortest paths to almost all Internet organization domain network routers, mapping the destination address (network) of the processed packet to its appropriate shortest path, called Routing Path Table (RPT);

- the Forwarding Path Table (FPT), holding the shortest paths to almost all Internet organization domain network accessible routers, with identical functions, lookup and storing techniques as the current Forwarding table; and

- the ID Map Table (ID MT) would map the neighbor Router ID to the appropriate output port of the router.

The proposed RPT, for every known destination IP network would store the complete shortest path, represented by an array of router IDs (Table 1).

Table 1: Routing path table

Destination Network	***Routing path (consisted of Router IDs)***
93.87.117.213	10000, 5FF00, A4567, 00012, 3445
…	
93.87.114.219	1000A, AFF00, A4567, 00012, 3445
93.87.118.223	1000B, 5FC00, A4C67, 00012, 3445

The RPT, FPT and ID MT tables would be continuously maintained by a control-plane general-purpose processor, using a special modified existing routing protocol for exchanging topology information and Router IDs information. The protocol would be similar to other topology or distance vector-based routing protocols such as OSPF or BGP calculating and updating the needed information in "background" with the possibility of replacing those protocols entirely. The routing protocol would continuously calculate and store the shortest path(s) to every known router in the RPT, and update data concerning neighbor routers and their availability. Having this information already at hand, the data-plane network processors will be able to obtain the routing information for the received packet in only a "few" cycles from the FPT. This part corresponds to current routing techniques, with the exception of extracting the entire path of the packet to the destination, not only the next hop.

To further accelerate the routing process, the responsible router at the network entry point of the packet should insert the entire path to the destination in the packet's header. Then, forwarding the packet in next intermediate hops would comprise of solely reading the next router ID from the packet header, and mapping that ID in the ID MT (Table 2); thus sending the packet to the appropriate output interface right away, skipping the forwarding table lookup process. This step radically reduces the routing time and allows achieving very high processing speeds, since the ID MT is drastically smaller in size than the currently used FT for destination lookup.

TABLE 2: ID MAP TABLE

Neighbor Router ID	***Outgoing port***
10000	Ifc0
…	
1000A	Ifc5
1000B	Ifc8

Even though the packet path through the network is being predefined by the first router, since every router on the packet's path holds the complete Internet organization domain level routing topology, it is however the optimal path of the packet. The concept is flexible enough, since every router on the path can resort to the traditional routing method and recalculate the remaining path of the packet if the predefined next hop router is in no position of handling the traffic.

The path of the packet inside the IPv4 packet header is being stored utilizing the Options Field. The longest hop count on the Internet, measured in [9] is 28, and in [10] it is 32. IPv4 header Options Field length is limited, so after defining the option number, and Options Field length, only around 300 bits are left for storing the path. Since we made a decision to indentify the routers by a 20-bit Router ID, it takes 20 bits to store one Router ID. Accordingly, IPv4 header allows storing the first 15 hops of the packet path. If the entire path is longer, the remaining hops would be inserted by the last router of the previous path. For IPv6 there is no such limit, since the options are defined as extension headers with variable lengths.

To estimate the amount of maximal needed memory capacity for the previously proposed RPT, decisions from the preceding sections can be taken into account, but the number of known IP networks (prefixes) at every router is needed as well. If we estimate this number of IP networks (prefixes) to around 2^{27} (which is again quite a large number), and if we take 15 (i.e. 2^4) as the max length of a packet's path from the source to the destination that can be stored inside IPv4 packet header, then the total memory capacity would be

$$\text{Num.dest.nets} * \text{Path\_len.} * \text{RouterID\_len.} = \qquad (1)$$
$$2^{27} * 2^4 * 20 \ = \ 10 * 2^{32}\text{b} \ = \ 10 * 2^{29}\text{B}$$

This amounts to a maximum of 6 GB of memory for keeping the whole RPT, which is obviously possible to implement in today's routers, considering current trends of memory capacity growing, and memory cost decrease. The ID mapping table, will be of similar size to current mapping tables for interface port to IP address mapping.

IV. PERFORMANCE ANALYSIS

Since the concept of keeping the packet path through the network inside the packet header introduces excess data, first we evaluate the increase of packet header and packets sizes in general. The size of the packet header increases linearly, since every additional hop adds 20 bits of data in the header. Fig. 1 shows the general packet size increase (in %) depending on the path length for three most common packet sizes seen in the Internet. The decision to keep the network packet path inside the header has the biggest influence on the smallest sized packets, while bigger network packets experience less overhead. Due to the fact, that the vast amount of Internet data is being carried by bigger sized packets, the overhead introduced by the path in the packet header would not be very significant.

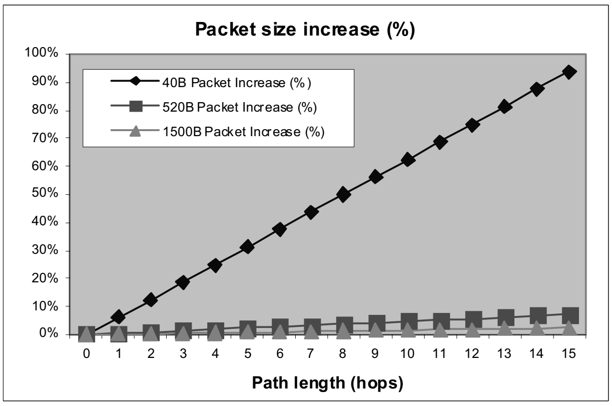

Fig. 1. Packet size increase due to header packet path

Furthermore, we estimate the performance by analyzing assembler programs for IPv4 and IPv6 packet processing. Results have been gathered by comparing the general IP protocol and the modified IP routing protocol using a general purpose RISC processor and the modified RISC network processor (NET RISC) [3].

According to [11] and [12] the average number of memory accesses needed to determine the next hop IP address (or the complete path in our case), in usual circumstances, is around 5 for IPv4 forwarding table lookup, and around 7-8 for IPv6 lookup. Additionally, currently available RAM memories (SRAM and SDRAM) used for FT in routers, according to [6], have memory access times of 5 ns and 20 ns, accordingly. These memory access times (referred to as a memory cycle) are being normalized to the number of processor cycles needed for one memory access. Hence, the performance analysis is being depicted as average processor cycles needed for IP packet processing. We decided to show the comparison using SRAM memory access times, since they are significantly faster than SDRAM memory access times. This proves that the gain when using SDRAM would be significantly higher. All results are normalized according to the length of the path (number of hops) inserted in the header.

The packet processing time for traditional IPv4 protocol is equal in every hop, while for our modified routing algorithm the average processing time decreases with the path length, as a consequence of the fact that the lengthy lookup process has to be performed only at the first hop, while subsequent processing is much faster. The probability that the next hop from the path in the header is not available is considered to be 0.1 in every hop.

Fig. 2 compares the average processing times for IPv4 packets with traditional routing on a RISC processor to our proposed routing algorithm executed on the NET RISC processor.

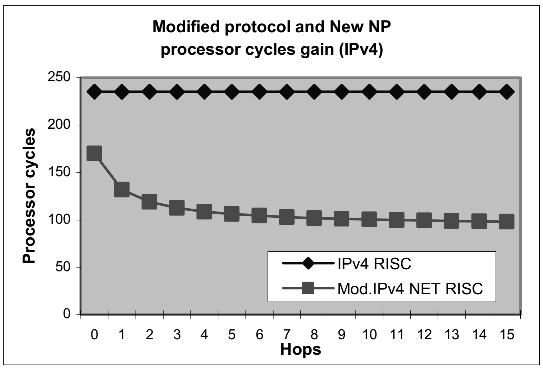

Fig. 2. IPv4 packet processing comparison

Fig. 3 presents similar comparison for IPv6 packet processing. Although the initial lookup process and path loading in the header takes more processor cycles than regular IPv6 lookup, the subsequent much shorter processing enables the new algorithm executed on NET RISC processor to outperform the regular IPv6 processing.

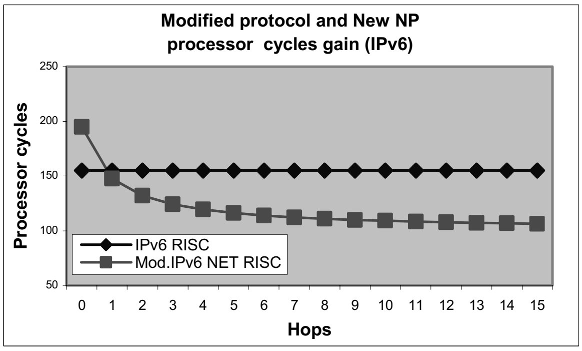

Fig. 3. IPv6 packet processing comparison

Fig. 4 shows the processor cycles gain for both IPv4 and IPv6 processing, again normalized to the path length through the network. As presented, the processor cycles gain grows continually, even though slowly, after the initial gain of 50% reached around 4 hops for IPv4, and 25% reached around 5 hops for IPv6.

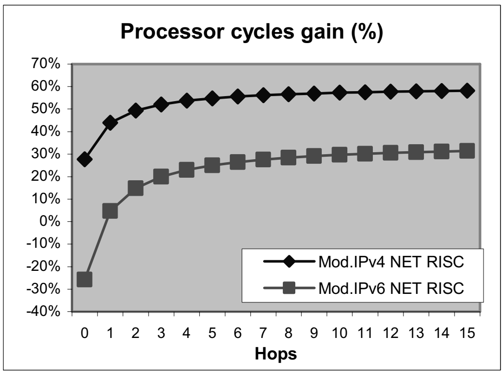

Fig. 4. Ipv4 and IPv6 processor cycles gain with modified protocol and novel NET RISC processor architecture

It has to be mentioned that these are after all only initial results and perhaps some limitations have not been taken into consideration. It is possible that the final gain will not show such percentages. However, these results encourage the authors to further investigate the NET RISC network processor architecture, and the proposed IP routing concept to show that the considered combination can satisfy the theoretical limit boundaries of processor cycles for 10/100 Gb/s processing. Thus, there is ongoing work for performing additional hardware design and software simulations to further improve the proposed concept and achieve better more realistic results.

V. CONCLUSION

In this paper, some new ideas have been proposed: replacing the hierarchically based contemporary routing process with a process similar to the routing in Internet early days; reinvestigating source routing; and finally using adequate network processor architecture able to cope with such routing concepts.

The RISC–based network processor architecture is particularly adapted to support the new routing process architecture with vast amount of memory on each router. All the considerations imply the architecture together with the modified routing concept as very promising and able to cope with multi gigabit processing.

Intended future and ongoing work comprises developing and researching new memory architectures for routing purposes, new strategies for faster memory lookup, simulating the modified routing process, and the network processor in nearly real environment. However, we believe that there are no technological obstacles to achieve the given goals and make them a reality.

REFERENCES

[1] R. Giladi, “Network Processors - Architecture, Programming and Implementation”, Ben-Gurion University of the Negev and EZchip Technologies Ltd., 2008

[2] Kamoun Farouk, “Design considerations for large computer communication networks”, PhD Dissertation, UCLA, 1976

[3] M.Kalendar, D.Jakimovska, A.Tentov, G. Dokoski, “Novel Processor Architecture For Modified Advanced Routing in NGN”, *SAC'11:The 2011 ACM Symposium on Applied Computing 21-MAR-2011, TaiChung, Taiwan*, to be published

[4] S. Hauger, T. Wild, A. Mutter, A. Kirstädter, K. Karras, R. Ohlendorf, F. Feller, J. Scharf, “Packet Processing at 100 Gbps and Beyond—Challenges and Perspectives”, *Proceedings of the 10. ITG Symposium on Photonic Networks*, May 2009, pp. 223-230

[5] P. Gupta, S. Lin, and N. McKeown, “Routing lookups in hardware at memory access speeds,” in Proc. IEEE INFOCOM’98, Session 10B-1,San Francisco, CA, pp. 1240–1247.

[6] W. Eatherton, G. Varghese, Z. Dittia, “Tree bitmap: Hardware/Software IP Lookups with Incremental Updates”, *SIGCOMM Comput. Commun. Rev., vol. 34, no. 2*, 2004

[7] http://news.netcraft.com/archives/web_server_survey.html [online at 24.08.2010]

[8] http://www.cidr-report.org

[9] Aiguo Fei, Guangyu Pei, Roy Liu and Lixia Zhang, “Measurements on Delay and Hop-Count of the Internet“, Departement of Computer Science, University of California, Los Angeles, CA 90095

[10] V. Paxson, “End-to-end routing behavior in the Internet”, *IEEE/ACM Transactions on Networking,* vol.5, no.5, pp. 601-615, October 1997

[11] L. Peng, W. Lu, L. Duan, “Power Efficient IP Lookup with Supernode Caching”, in *Proc. IEEE GLOBECOM '07*, Washington, DC, pp. 215 – 219

[12] Shuping Liu, “IP Lookup in IPv6 Networks”, Postgraduate Course on Networking Technology, Course Topic Spring 2005: IPv6, Mobility and Alternatives, Networking Technology Lab, Helsinki University of Technology, Finland

[13] H. Jonathan Chao, Bin Liu, High Performance Switches and Routers, Wiley-IEEE Press, May 2007