Terzioglu, H., & Kocak, T. (2004). A network processor for a learning based routing protocol. In 2nd Annual IEEE Northeast Workshop on Circuits and Systems, Montreal, Canada. (Vol. 1, pp. 261 - 264). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/NEWCAS.2004.1359081

A Network Processor for a Learning Based Routing Protocol

Hakan Terzioglu and Taskin Kocak
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
hakan@cpe.ucf.edu

Abstract— **Recently, Cognitive Packet Networks (CPN) is proposed as an alternative to the IP based network architectures and shows similarity with the discrete active networks. In CPN, there is no routing table, instead reinforcement learning (Random Neural Networks) is used to route packets. CPN routes packets based on QoS, using measurements that are constantly collected by packets and deposited in mailboxes at routers. The applicability of the CPN concept has been demonstrated through several software implementations. However, higher data traffic and increasing packet processing demands require the implementation of this new network architecture in hardware. In this paper, we present a network processor architecture which supports this learning based protocol.**

I. Introduction

There is a strong demand for novel routing architectures that can provide more efficient and robust service to the Internet constituents. There are already proposed alternative packet-switched network models that would eliminate some of the problems of IP based networks. Some of them have the potential to fundamentally change the structure of the core Internet, making it more scalable for future generations. Active networks, which offer users the capability of adding executable code to their packets, received much attention. There are two approaches to active networks: Integrated and discrete. In the integrated approach (used by Active Network Transport System (ANTS) [2]), every packet is a program fragment that may include embedded data. In the discrete approach, users first send their packets with code in order to program the routers, then, send data packets through such programmed network. In each router the appropriate program is on the packet contents. Recently proposed Cognitive Packet Network (CPN)[5], [4], [6] shows similarity with the discrete active networks. CPN attempts to solve some of the problems associated with the legacy IP networks, such as QoS, the never-ending expansion of routing tables and their related maintenance issues. The primary motivation for this study is the design of a learning based router architecture. We will show that this novel architecture shows very good performance metrics compared to the commercially available NPUs.

II. Cognitive Packet Network

The CPN is a store and forward architecture that achieves intelligent QoS based routing by employing "smart or cognitive" packets. The CPN uses three different types of packets: smart packets, payload packets, and acknowledgement packets. The payload packets carry payload (the user's data) and are source routed by applying routing information generated from the experiences of the smart packets. Smart packets are sent out continuously to search for routes to a destination. Before a particular flow of payload packets can be transmitted, the route information for the QoS class must be available at the source. If it is unavailable, the source node will create and dispatch smart packets to determine routes to the selected destination for the required QoS class. As the smart packets propagate through the network, they collect measurement data with regard to link quality. Acknowledgement packets carry back this measurement data, depositing it at the CPN routers as they travel the reverse route of the smart packets. The original source node uses the data to establish a route for the payload packets. The CPN router acts as a buffer for packets, as a storage area for mailboxes (MBs) where acknowledgement packets deposit measurement data, and as a processor for packets. It receives packets via a finite set of ports and stores them in an input buffer, where sorting based upon QoS requirements may occur. It forwards packets to other nodes via output buffers, and runs the algorithm used to make routing decisions concerning smart packets. Contrary to a conventional IP router, a CPN router does not maintain a routing table. The routing decisions in a CPN router rely upon a learning algorithm[1]. Previous attempts to incorporate learning algorithms and adaptation into packet networks have been insufficiently researched due to the lack of practical mechanisms. CPN routers execute a reinforcement learning algorithm in order to select the output link for smart packets. Payload and acknowledgement packets are source routed with their routes stored within the packets themselves. The reinforcement learning algorithm that smart packets rely on is based on a QoS "Goal". The term "Goal" is used to indicate that there is no QoS guarantee rather there is a best effort attempt to satisfy the QoS objective. The smart packets act as network explorers. They travel through the network, finding routes and collecting data. The measurements and the path traveled by the packet are stored in its Cognitive Map (CM). When the smart packet arrives at its destination router, the router generates a corresponding acknowledgement packet.

0-7803-8322-2/04/$20.00 ©2004 IEEE.

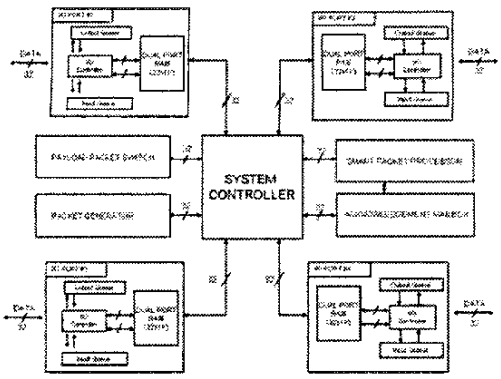

Fig. 1. CPN network processor architecture

The acknowledgement packet inherits the smart packet's source as its destination. In addition, the smart packet's CM is inverted and stored as the acknowledgement's CM. As the acknowledgement travels through the network, routers will reference its CM to find out where to send it next. Thus, the acknowledgement packet is source routed, following the inverse route of the smart packet that initiated it. Before the routers forward the acknowledgement to its next hop, they will read the relevant measurement data from its CM. In the reinforcement learning algorithm, the observed outcome of a decision is used to "reward" or "punish" the routing algorithm with respect to that decision. The "Goal" is the metric that characterizes the success of the outcome, such as packet travel time or transit delay. As an example, the QoS *goal* (G) that smart packets pursue may be formulated by as minimizing *transit delay* (W), *loss probability* (L), *jitter*, or some weighted combination, for instance:

$$G = a * W + b * L \tag{1}$$

where a and b are constants selected by the application layer which signify the relative importance of the delay and loss for this particular application's QoS.

III. CPN Network Processor Design

The architecture of the CPN network processor is shown in Fig. 1. This is an application-specific, hard-wired architecture which supports the learning protocol discussed in Section I. We did not prefer to use FPGAs in our work to exploit the high performance and speed of a full custom ASIC design. In the following sub-sections, we describe each major module individually.

A. I/O ports

In the current implementation, each CPN NPU has four I/O ports. Both input and output queues can store up to 30 memory addresses in their registers. Each register in the queue is 12 bits long and the first two bits indicate the type of the packet stored in that address. Remaining 10 bits represent the memory address which the packet is stored. I/O controller manages the communication between the Dual Port Ram (DPR), input/output queues and the system controller. I/O controller handles the following tasks:

- Calculating the address to be written in input and output queues. This is accomplished by parsing the packet header and calculating the packet length. The address written to the queues is updated according to this value.
- Acting as a buffer for incoming and outgoing packets.
- Controlling the READ and WRITE operations for queues and the DPR. It sets the read/write signals for these modules.
- Functioning as an arbiter for write operation to the DPR.

The management of the input and output queues is slightly different. Memory addresses stored in the output queue are fetched in a FIFO approach whereas addresses in input queue are fetched based on the type of the request. For example, if the request is sent from the payload packet switch, I/O controller searches the first payload packet address in the queue. I/O port can receive/send packets from/to the other nodes and receive/send packets from /to the modules within the router simultaneously since DPR is deployed. Dual Port RAM (DPR) is 32x1K in size and has two separate ports (A and B). Incoming packets from other nodes are written to the DPR through port A. Outgoing packets to other nodes are read from port B. The processed packets from modules (SPP, ack mailbox, packet generator, payload packet switch) are written to the DPR through port B. Outgoing packets to the other modules within the router are read from port A. Thus, the input queue communicates with the DPR through port A, and the output queue communicates with the DPR through port B. Read and Write operations can be done simultaneously through both ports. I/O controller can manage simultaneous write operations.

B. Smart Packet Processor

In our previous work, we have implemented the Smart Packet Processor and showed the functionality of the design[7]. The function of the Smart Packet Processor (SPP) is to determine the outgoing port for arriving smart packets based upon its QoS, source and destination parameters (QSD). The SPP employs the RNN model with the reinforcement learning algorithm to make its decisions. The reinforcement learning algorithm requires the reward value calculated with the data from the acknowledgment packets. Therefore, the SPP is a convergence point for the flow of both smart and acknowledgment packets. To meet its requirements, the SPP needs to interface with two external structures: The acknowledgment mailbox and the system controller. The ideal design allows simultaneous transactions between these components. The system controller waits for the SPP output to forward smart packets. On the other hand, the acknowledgment mailbox is merely a receptacle for the data measurements within the acknowledgment packets. Therefore, the priority of the SPP design must be the servicing of the smart packets. The smart packet processor consists of four different components: The smart packet interface, the reinforcement learning algorithm,

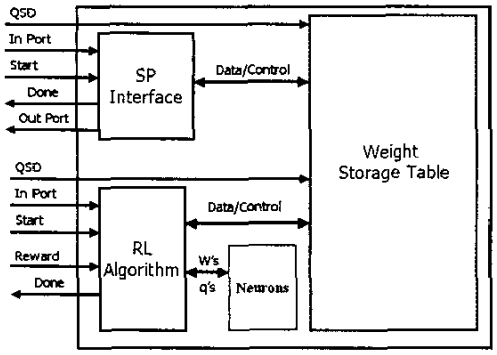

Fig. 2. Composition of the smart packet processor

the neuron array, and the weight storage table. The SP interface is externally connected to the system controller while the RL algorithm receives control from the acknowledgment mailbox. As seen in the figure, both components have data and control paths to the weight storage table. The table is a complex dual port memory structure that stores the RNN weights, thresholds, outputs and QSD indexes. Lastly, the neurons are controlled by the RL algorithm and are used to calculate the steady state output of the RNN.

C. Payload Packet Switch

Payload packet switch (PPS) forwards the payload packets to the next hop in the network. Each node in the network has one PPS module. PPS communicates with the system controller through a 32-bit bus. Each payload packet carry the routing information and the position of the next hop in the cognitive map within its header. Once the next hop is detected through a simple algorithm, relevant information is sent to the system controller and the packet is stored in the output queue of the I/O port that is connected to the next hop.

D. Packet Generator

Packet generator module is triggered by the system controller in two situations: 1) When the node acts as a source node and originates a smart packet, or 2) When either a smart packet or an acknowledgment packet is received and its destined to the current node. In the latter, if the received packet is a smart packet the packet generator use the information in the CM and generate an acknowledgment packet. Node IDs and reward values for the links between the nodes are inversed and the generated packet is destined to the source node of the smart packet. In a similar manner, if the received packet is an acknowledgment packet, the system controller triggers the packet generator and a payload packet is generated using the information in the CM of the acknowledgment packet. Payload data besides the CM is also appended to the payload packet.

E. System Controller

System controller handles all the communications between the packet processing units and the I/O ports. This multitasking unit is interfaced with other modules with separate 32-bit busses to avoid bottlenecks associated with a common bus. When one of the modules in the router is idle, the system controller fetches a packet corresponding to that module from one of the I/O ports in a round robin fashion. For example, if the idle module which requests data is payload packet switch, then the system controller informs the I/O controller for a payload packet search in the RAM. In the case where one of the I/O ports is congested, priority of service will be given to that port. The second major function of the system controller is to determine what kind of routing strategy to be employed. Based on the packet's destination, there can be three different cases. First, if the current node is an an intermediate node in the packet's path then the system controller will trigger the corresponding module (e.g., smart packet - SPP). In the second case, if the packet has arrived at a router connected to the destination router, then the system controller will immediately attempt to forward the packet to its destination. In the final case, if the packet has arrived at its destination router, then it has successfully completed its journey.

F. Acknowledgment Mailbox

The final component of the CPN router is the acknowledgment mailbox. The mailbox is where acknowledgment packets deposit the relevant data measurements they are carrying. In our hardware design, the mailbox is also responsible for calculating a reward value from the measurement parameters. This reward value is used to "reward" or "punish" the routing algorithm with respect to the decision made at the current node when routing the smart packet corresponding to this acknowledgment packet. Next, it must forward the reward value to a component that incorporates the RNN with reinforcement learning. In the meantime, the acknowledgment packet needs to be transmitted to its next hop in its path. Since the packet is source routed, the mailbox features a design similar to the payload packet switch exclusively for use by the acknowledgment packets.

IV. Network Performance Analysis

Simulations are run to measure the performance achieved by the design. We use standard network performance metrics such as latency and throughput for evaluation. As in the software implementation of the CPN testbed [3], [4], it is modeled such that each port of a CPN node uses a 10 Mbps Ethernet link connected with another CPN node.

Cognitive packets or smart packets are of variable size and consist basically of three areas: a header, a cognitive map and data portion (payload). In general, smart packets do not carry payload. However, the payload packet generation and processing modules are not currently available. Hence, for this work, we have used smart packets with payload to measure the performance parameters. Fig. 3 shows the average delay of smart and acknowledgement packets for various payload sizes.

The average delay represents the average latency that a packet experiences from entry to exit. The latency for the

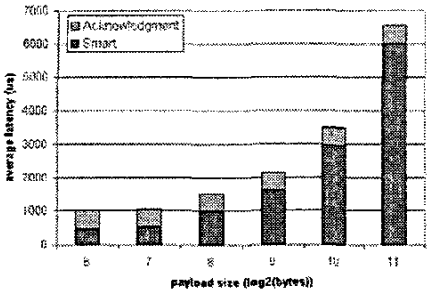

Fig. 3. Latency of smart and acknowledgement packets

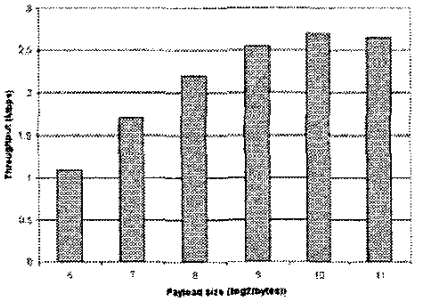

Fig. 4. Throughput of smart packets

smart packets increases with the packet size as expected. Since acknowledgement packets consist only a header and a CM, their latencies are not affected by the payload size change. Fig. 4 shows the throughput of smart packets for various payload sizes. Throughput is determined as the amount of user data tranferred by the CPN network. In theory, as packet size increases packet throughput should also increase. In our simulations, we notice that packet throughput increases nonlinearly as the payload size increases except a small drop for the packet size=2048 bytes. This small drop is within the simulation tolerance limits, and as can be seen from the figure, the throughput is asymptotically converging to between 2.5 and 3 Mbps. We have also looked at the packet loss probability within the CPN network and the results are illustrated in figure 5. The x-axis represents the inter-packet time for the main flow of packets. The experiment is conducted with a fixed input rate for a payload size of 250 bytes. We observe a significant improvement in the packet loss probability as the transmission time increases. This is due to the fact that there is no buffer in the nodes, thus some packets are being dropped if they cannot be forwarded to the next node.

V. Conclusions

A design for a novel network processor, based upon the Cognitive Packet Network model, is presented. Implementation of the most of the blocks in the CPN network processor is completed in VHDL. Pre-synthesis simulations are

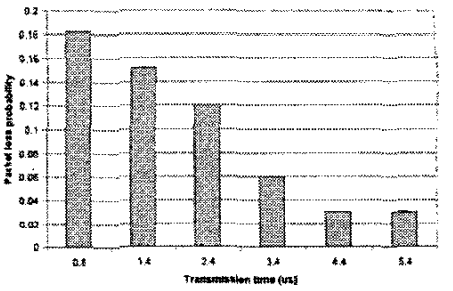

Fig. 5. Packet loss probability for smart packets

run to confirm the proper functionality of the design. Post-synthesis simulations for the smart packet processor shows a 50 Gbps wire-speed processing performance. The next step in the development of the CPN router after completing all the modules is to test the device at the system level by forming a network of CPN routers. Future directions for this work can be investigating whether the design can be converted into an instruction set based one and be made compatible with IP networks. Also, it would be interesting to look into the realizations of the CPN model on different network processor platforms (e.g., Intel IXP series).

References

[1] E. Gelenbe, "Learning in the recurrent random neural network", *Neural Computation* vol. 5, no. 1, pp. 154-164, 1993

[2] D. Wetherall, J. Guttag and D. Tennenhouse, "ANTS: A toolkit for building and dynamically deploying network protocols", *Proc. IEEE OPENARCH*, Apr. 1998.

[3] E. Gelenbe, R. Lent, and Z. Xu, "Design and analysis of cognitive packet networks", *Performance Evaluation*, vol. 46, pp. 155-176, 2001.

[4] Z. Xu, "Design and analysis of adaptive routing in cognitive packet networks", *Ph.D.Dissertation*, University of Central Florida, Orlando, 2001.

[5] E. Gelenbe, R. Lent, and Z. Xu, "Design and analysis of cognitive packet networks", *Performance Evaluation*, vol. 46, pp. 155-176, 2001.

[6] R. Lent, "On the design and performance of cognitive packets over wired networks and mobile ad hoc networks",*Ph.D. Dissertation*, University of Central Florida, Orlando, 2003.

[7] T. Kocak, J. Seeber, and H. Terzioglu, "Design and implementation of a random neural network routing engine", *IEEE Tran. on Neural Networks*, Sept., 2003.