An Evaluation Environment for Programmable Forwarding Elements

Jing Fu and Olof Hagsand
Department of Microelectronics and Information Technology,
Royal Institute of Technology (KTH)
S-16440 Kista, Sweden
{jing, olofh}@imit.kth.se

Current routers can be divided into two logically separated parts including control elements (CEs) and forwarding elements (FEs). CEs are responsible for processing of routing, management and signaling protocols while FEs are responsible for packet forwarding in real time. Current FE devices are often based on the ASIC, FPGA, GPP or network processor technology. Deciding which technology to use involves a trade-off between flexibility, performance, cost, development time and power consumption. In this work, we look at network processors, programmable forwarding elements that can be programmed to support different forwarding services, including Diffserv, probe-based admission control for multimedia traffic and other forwarding services for multimedia traffic.

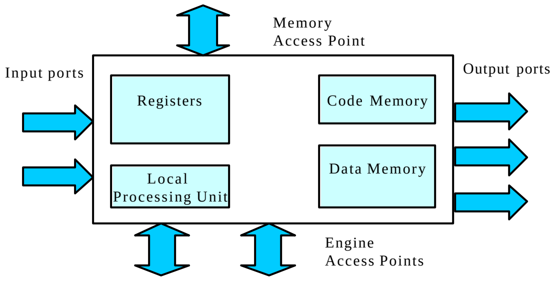

Figure 1: The Structure of a Processing Block

We have designed and implemented an evaluation environment for programmable FEs. The main components of the environment are a set of *processing blocks* and *engines*, which are internally interconnected in an infrastructure topology. A processing block represents a program running on a processing element while an engine represents the functionality provided by a special-purpose hardware engine. The structure of an example processing block is shown in *Figure 1*. A program running on the local processing unit, may receive packets on input ports, transmit packets on output ports, access memory through the memory access point, access engines through engine access points, read and modify registers, etc. Processing blocks are internally connected into a processing block topology using unidirectional channels. The topology can be seen as a directed graph where processing blocks are nodes and channels are edges. Packets entering a FE may take different paths in the topology depending on the packet processing.

We provide mechanisms to evaluate the design of programmable FEs. We have chosen to evaluate the FEs based on *programmability* and the *performance*. Current FE devices are very complex in architecture and are usually difficult to program. The evaluation environment for evaluating programmability provides an API to program FE devices and supports several programming paradigms. A designer can program a forwarding service using different styles, and run the programs interactively. Except measuring the difficulty of programming a forwarding service using a specific style, we may also measure the performance achieved using this programming style.

In order to evaluate the performance, the environment provides a simulation timing model to measure packet throughput and delay. Each component processes the packets and advances in the simulation time in a synchronized manner. A set of time parameters such as memory access time, and engine access time is used to describe the hardware characteristics of the forwarding element. By experimenting with time parameters, number of processing blocks and the processing block topology, we are able to evaluate a forwarding element design. The time parameters can also be configured according to the hardware characteristics of a real FE device. By developing forwarding services such as IPv4, Diffserv, etc, we are able to evaluate the performance a FE device provides, and also the performance of the given forwarding service. This includes packet throughput, packet delay, jitter, and how are different packets such as multimedia packets, probe packets treated inside the forwarding element.

In short, we have designed a general evaluation environment for programmable forwarding elements. With this environment, we can model several existing network processors. We have started with Intel's 2400 and Xelerated's X10. We can then program a forwarding application, and evaluate them in terms of performance and ease of programming. From this work, we believe that we can evaluate the different designs, and hopefully influence the design of next generation network processors and FE devices.