# Thesis Abstract

<table>
<thead>
<tr>
<th>Registration Number:</th>
<th>“KOU”</th>
<th>“OTSU”</th>
<th>Name:</th>
<th>Yohei Kuga</th>
</tr>
</thead>
</table>

**Title of Thesis:**

Packet Scripting Architecture with Precise Packet Timing Control

**Summary of Thesis:**

This dissertation proposes a scriptable packet processing architecture, "EtherPIPE", which is capable of precise timing controlled packet injection and precise timestamping.

With the diversification and performance of Internet services, the backbone of the Internet is growing steadily. Intelligent Network Interface Cards (NICs) of 10 Gbps bandwidth became a commodity and are widely deployed as server-to-server interconnections. Thus, packet processing controlled by software provides programming flexibility with high throughput, applicable for network troubleshooting, benchmarking, packet forwarding and Internet measurement.

Unfortunately, software packet processing cannot achieve precise timing control. Existing software processing is limited by the timer accuracy of the OS scheduler, and transmit and receive timing of packets are controlled at microsecond accuracy. Hardware packet processing can manage the timing to nanosecond accuracy. But network hardware devices are not programmable and supporting complex packet processing using them is difficult. Also, in a high-throughput network with more than 10 Gbps bandwidth, time resolution greatly affects the performance. Thus, timing-sensitive packet processing requires specialized hardware such as FPGA or ASIC.

EtherPIPE is a new architecture for a software packet processing designed for timing-sensitive network applications, which make use of the NIC hardware functions, its device driver and a programmable scripting framework. The EtherPIPE NIC consists of two functions to control packet receiving and transmission timing. The NIC receives a packet with a timestamp of PHY-clock accuracy and schedules the time of packet transmission as designated by scripts. The EtherPIPE packet I/O is abstracted as a character device and each packet data can be translated from a binary stream into an ASCII character string. By using this convention, we can program in the UNIX shell programming manner via character devices that can be connected by Stdin and Stdout.

As a result, the EtherPIPE architecture provides a flexible packet processing framework for applications that need a precise packet timing with the same precision as network special hardwares. To validate the
effectiveness of our approach, we have developed a packet generator and a middlebox for latency emulation, and demonstrate that their timing is 100 times more accurate than software packet processing. Both applications required only a small number of lines of code. This architecture contributes the rapid and low-cost development of high-precision network features for network device prototyping and troubleshooting.

Keywords: Software-Defined Network, NIC hardware, Network I/O, Shell scripting, Internet Measurement.