

# **Paper Evaluation, Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN**

Jiani Jiang <jianij@kth.se>

## **1. Paper summary**

As OpenFlow is popular as the interface between the control plane and the data plane, the Match-Action approach that it used to apply actions to packets, is implemented in hardware to develop pipelining and parallelism. However, the speed of switching chips is limited with memories and costs. Also, the hardware needs to support feasible programmability without affecting the forwarding speed. This paper compared Single Match Tables and Multiple Match Tables, and explored RMT (Reconfigurable Match Tables) that improved from the previous two tables for Match-Action hardware to support packet processing at run-time. To be more specific, RMT allows pipeline stages that each with a match table and reconfigure four more methods to easily tolerant new protocols with new header fields. In addition, the RMT model is perfectly compatible with the OpenFlow protocol. The general framework of RMT is a multi-pipelining parallel architecture consisting of a reconfigurable parser and an arbitrary number of match stages.

## **2. Top 3 contributions**

1. This paper imposed the design and implementation of RMT and make it possible to support arbitrary configuration flexibility by reconfigurable matching without any loss of performance, so as to improve the level of the network programmability, which is very innovative.
2. This paper gave the specific form of reconfigurability and described the implementation of a real switch chip that proved its feasibility in timing and chip cost and made it convenient for network operators to learn and implement RMT in real-world switches.
3. This paper encouraged better utilization of chips, while chip design is not the filed of SIGCOMM. It gave people a good way to think about using hardware to achieve flexibility that comes at almost no cost.

## **3. Problems**

In my opinion, compared software and hardware, software processing speed is lower than the hardware, but software can maximize rule processing flexibility. At the same time, can avoid burst traffic problems due to hardware itself small, limited flow table size and memory. However, the design of RMT based chip realizes flexible processing of data from the hardware perspective. So, I think, in the data plane, which elements can be handed over to the hardware, and which elements can be delivered to the software, are also a problem that is worth considering.