SDN Switch Design

Rohan Jani

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

Introduction

The project address's the challenge to introduce a 5-port SDN (Software Defined Network) switch for small-scale commercial and residential enterprise.

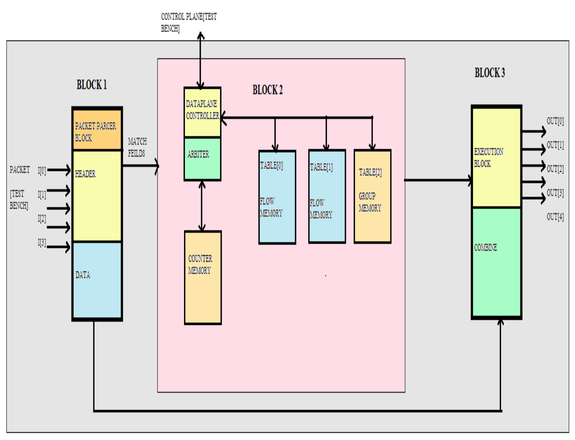
The project in-corporates the Flow Table Block design and Data Controller with control plane interaction.

Why SDN ?
SDN technology creates abstractions in the network, which reduces protocol dependencies by separating the logic from the hardware.

- Cheaper hardware by using intelligent, separate software control.
- Reduced downtime by improved failure recovery.
- Complete network control for enhanced flexibility and user adaptability.
- Can route traffic to a a central firewall to boost security.
- Improved Resource Management by creating a virtual environment by centralizing network control. Removes manual configuration of devices and provides efficient use of bandwidth and network recourses.

Thus cheaper hardware, network control and improved security makes it ideal for 5 port switch.

Data Plane Architecture



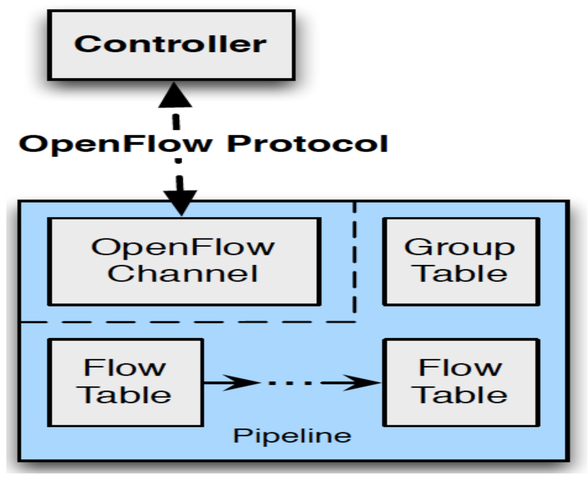
Block 1: it parses the ingress packets in two separate entities of fields and data. It stores data in memory and returns its address pointer. It outputs an array of 244 bits consisting of empty action set, match fields and metadata.

Block 2: Performs match algorithm and updates the empty action set. Interacts with control plane to fill flow tables and resolves process queries.

Block 3: Performs commands by evaluating the action set. Generates outbound packets and provides QoS, load balancing.

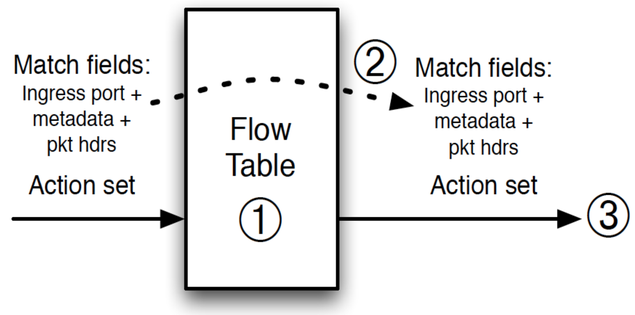
Data Controller

The data controller performs two major tasks
Function1 : It finds the match using the packet fields and the flow entries. Thus would update the action set, meta data and the counter accordingly.



Function2: It communicates with the control plane using Controller-Switch, asynchronous and symmetric messages to fill up flow tables and resolve packet processing queries.

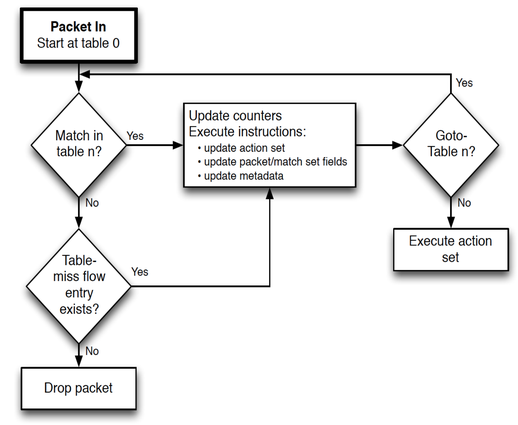
Open-Flow pipeline follows three base rules to process packets with multiple tables



1. Find the highest priority flow entry
2. Apply instructions: Update action set (clear actions and/or write actions instructions) and Update metadata
3. Send the metadata and action set to the next table.

- If a flow entry is found, its instruction set is executed. These instructions may explicitly direct the packet to another flow table (using the Goto)
- Pipeline processing can only go forward and not backward. Thus the flow entries of the last table of the pipeline can not include the Goto instruction.
- If the matching flow entry does not direct packets to another flow table, pipeline processing stops at this table.
- When pipeline processing stops, the packet is processed and sent to the execution block

If there no match and no got instruction to execute then pipeline processing stops and controller alerts the control plane. This phenomenon is called table miss.



The above figure describes the design flow of the data controller packet processing

Result

Have successfully implemented data plane controller with four single port synchronous ram memories.

Here the control plane and the parser block are simulated by a test bench.

The controller performs the immediate action of drop, goto and priority commands.

The design follows open-flow pipeline and the match algorithm to process packets.

The operational frequency is at 200 MHz and the design takes 5 cycle to find a flow entry match and 3 for a group entry.

The counter entries are designed as separate memories to meet the product requirement to be low cost, low maintenance.

For M flow and M1 group memories with N entries each, the

Time Latency = [(5*M) +(3*M1)]*N * Clock Cycle

Thus the worst case table miss match is designed to be under 1000 cycles.

The design performs linear search of memory find a match and also provides mechanism for future development of the product.

Conclusion

The project describes the design and the architecture of a small five-port switch and extensively deals with the design and verification of the data controller with four memories. The design is based open flow pipeline and the data-plane design of open flow specification 1.4.0. The project demonstrates flow/group table functionality, along with three data controller function. It also demonstrates symmetric message based control plane interaction .

Thus the device presented improves productivity, performance and security by providing Software Defined network control.

References

[1] Open Flow Version 1.4.0
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf.

[2]Table pattern: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/OpenFlow%20Table%20Type%20Patterns%20v1.0.pdf

For Further Information

Please contact *jani.r.rohan@gmail.com*. System Verilog code, simulation files and test cases are available upon request.