

FPGA based Packet Classification

By: Ruchi Saha

Guided by: Dr. Noor Mohammad SK

1. Aim

Our Aim is to create a dynamically generated memory efficient multi-match packet classifier on FPGA which can sustain High speed as well. The solution should also provide design flexibility of implementation and upgradation.

2. Need

- Technology is drastically changing day by day.
- It is costly to keep up with the changes in ASICs (Application Specific ICs) as it takes 18 months or longer to get it fabricated. FPGA provides the required flexibility.

3. Existing Works and it's limitations

- Existing works generates hardware only for the worst case. It does not provides hardware according to our needs hence wastes a lot of resources.
- Almost all of the solutions use TCAMs for prefix match which contributes to the longest path delay in case of FPGA.

4. Work done

- **Created algorithm, identified and optimised basic module.**
TCAMs prove to be the bottleneck on FPGAs. In order to avoid that, we identified and designed some basic hardware modules on which the architecture was based.
- **Constructed and pipelined the classification algorithm**
The design was created in verilog language.
- **Written code to dynamically generate the hardware for given parameters**
Python scripts have been written which can generate the given hardware (in verilog) with different pipeline widths according to the parameters. The parameters are taken from a configuration file (.ini file).
- **Implemented and evaluated the classifier on FPGA**
The design was implemented for Virtex-7 xc7vx485t board. The rule memory is implemented using Block memories available on the FPGAs.

5. Results

Table 1: Comparison with the existing FPGA-based designs. Comparison is for medium-scale databases (v2.9c Snort header rule database)

| Design | Device | f (MHz) | f' (MHz) | Throughput (Gbps) | Resources (#Slices) | Resource Efficiency (Gbps/Slice) |
|--------------------|----------|------------|-------------|----------------------|------------------------|--|
| BV-TCAM[30] | Virtex-5 | 125 | 230 | 18.4 | - | - |
| FSBV [12] | Virtex-5 | 167 | 230 | 73.6 | 3566 | 0.02 |
| Hybrid[14] | Virtex-4 | 211 | 317 | 25.36 | 549 | 0.046 |
| Proposed | Virtex-7 | 637 | 637 | 203.84 | 847 | 0.24 |

f: Original clock frequency, f': Normalized clock frequency

Table 2: Throughput Comparison with TCAM-based designs

| Rule Database | Throughput (Gbps) | | | |
|---------------|-------------------|------------|------------|----------|
| | GI [39] | SSA-2 [40] | Bitmap [4] | Proposed |
| v2.9c | 132.62 | 73.13 | 22 | 203.84 |
| v2.9.17 | 5.37 | 6.48 | 5.49 | 38.72 |

6. Conclusion

In this work, a logic-based multi-match packet classification architecture is proposed, where look-up is performed using logical comparators. The classification architecture is optimized in accordance with the real-world NIDS header rule database structural properties, for achieving resource efficiency and high clock frequency. Implementation results on the largest Snort header rule database has shown that, the proposed architecture achieved a throughput of 38.72 Gbps in the worst-case while consuming 3 the resources available on a medium scale Virtex-7 FPGA. The throughput achieved by the proposed design is at least 2.7x and 5.9x times higher than the existing FPGA-based and TCAM-based designs respectively. The resource efficiency of the proposed design is at least 5x times higher than the best existing FPGA-based design which demonstrates the resource efficiency of the proposed design while achieving high throughput. The design has been optimised for FPGAs, hence it provides the design flexibility of implementation and upgradation.