Netslice Bottleneck

David Kelly, dmk257
Ross Hanson, reh237

Context

- High performance packet processing libraries allow for line rate speeds on commodity hardware
- Netslice vs. netmap vs. Intel DPDK
- Netslice seems to hit a bottleneck at ~30Gbps

Motivation

- From an educational standpoint: to learn about packet processing frameworks working at bleeding edge speeds
- From an engineering standpoint: to improve upon Netslice and match its portability with performance

Analysis

- “Static analysis”
- Fuzz testing
- **OProfile**

Optimizations

- Compile time optimizations
- schedule() vs. schedule_timeout()
- Batching expensive updates

Results

- Localized where the bottleneck is coming from
- Small successes in mitigating it

1 Line	9195.2 Gbps
2 Lines	17922.4 Gbps
3 Lines	27926.2 Gbps
4 Lines	32204.2 Gbps

Future Work

- Create a final fix to the bottleneck
- Test in the wild with a production application

Lessons Learned

- Do not assume “mature” software can be implemented with minimal issues
- According to the internet, everyone innately knows how to profile kernel modules but no one has ever created a tutorial about it

Conclusion

- The current bottleneck does not seem to be the lack of zero-copy
- Given this, it seems plausible that Netslice could eventually reach the performance of netmap with better portability