Temasek Defence Systems Institute

TDSI
Temasek Defence Systems Institute

Contact: tdsbox2@nus.edu.sg

Network Device Software Generation

Shi Ronghua and Se Xi Yang Ronald
Dr. Dennis M. Volpano

Objectives

- To generate a software-defined network device, from elementary behaviours Symbolic Finite Automata (SFA), that can be guaranteed to exhibit only the required network function behaviour, by virtue of verification by construction.
- To run the generated software network device on a server-grade machine and have comparable performance to commercial grade switches.
- To introduce additional elementary behaviours SFA to expand the building blocks inventory

Network Behaviours

Basic Network Behaviours

Forwarding SFA F

	f1	f2
→f1	-	$\lambda x.loc = \{self\ i\}$
f2	$f = x.f \Rightarrow loc \subseteq egress - \{self\ e\}$	-

Informed Unicast Forwarding SFA U

	e1
→e1	$(self\ e \in loc \wedge ucast(f.da) \wedge f.da \in dom(mlt) \wedge t - mlt(f.da).t < 16) \Rightarrow mlt(f.da).port = self$

- Other basic SFAs include MAC Address Learning (ML), Socket Learning (SL) and Stateful Firewalling (SF).

Tensor Product of SFA

- Complex network behaviours can be built by taking the tensor product of the basic SFAs.

Basic Switching (FxUxML)

	f1e1m1	f2e1m1	f2e1m2	f2e1m3
→f1e1m1	-	S	S	S
f2e1m2	S	-	-	-
f2e1m2	S	-	-	-
f2e1m3	S	-	-	-

f1e1m1 → f2e1m2	$\lambda x.loc = \{self\ i\} \wedge ucast(f.sa) \wedge f.sa \in dom(mlt)$
f2e1m2 → f1e1m1	$f = x.f \Rightarrow loc \subseteq egress - \{self\ e\} \wedge (ucast(f.da) \wedge f.da \in dom(mlt) \wedge t - mlt(f.da).t < 16 \wedge mlt(f.da).port \neq self) \Rightarrow loc \subseteq egress - \{self\ e\} \wedge mlt = x.mlt(x.f.sa \mapsto \{t = x.t, port = self\})$

New Network Behaviours

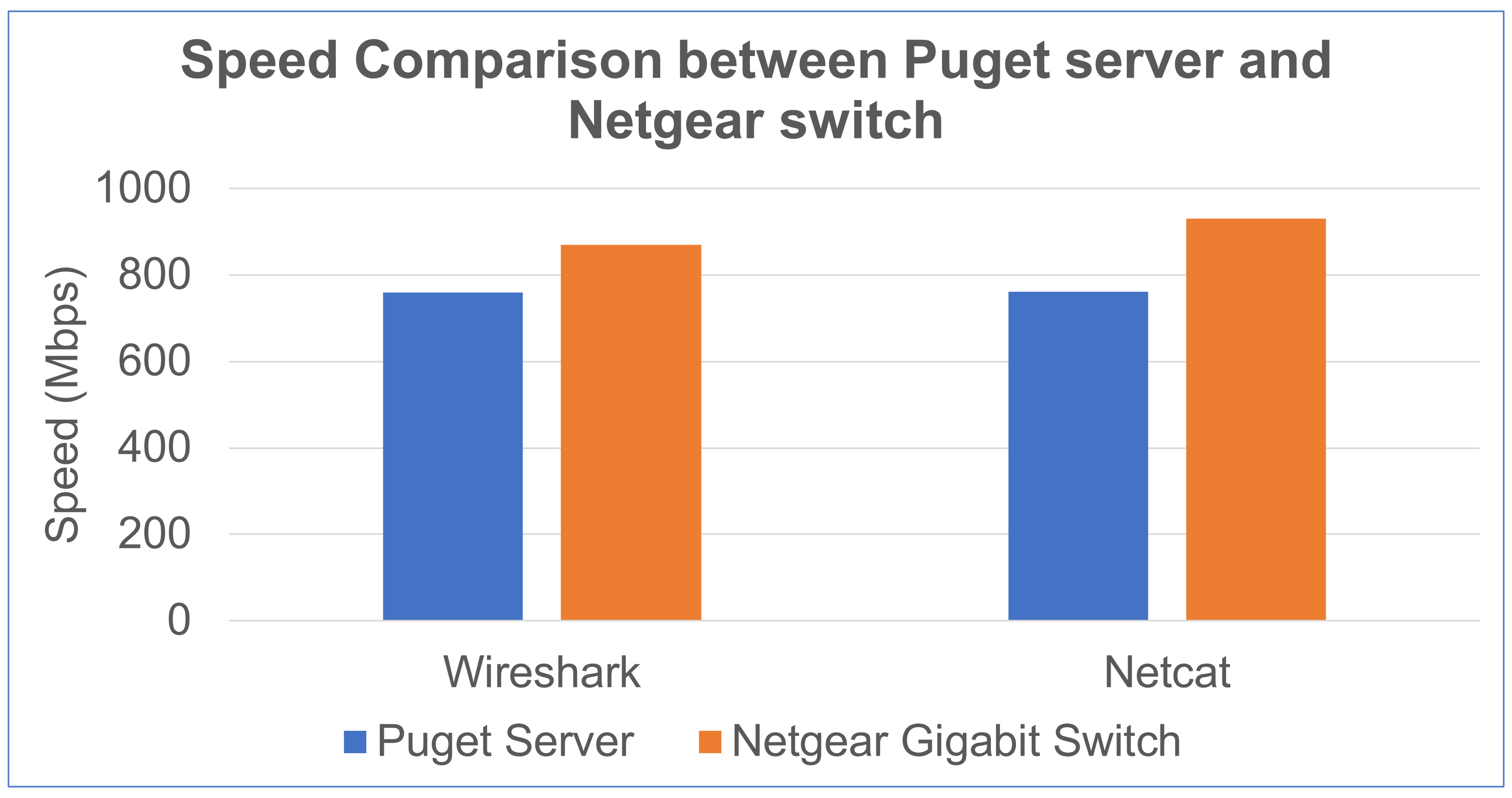
- New basic SFAs includes Routing (R), NAT learning (NL) and NAT Translation (T).

Generation of Network Device

- Intel's Data Plane Development Kit (DPDK) is used to generate code from the tensor product SFAs.
- Code synthesis algorithm adopts a Most Common Literal (MCL) technique to factor the literals with respect to its frequency of occurrence in the transitions.
- Each literal is mapped to a corresponding code block.
- Overall code is compiled and deployed on a Puget server with 4-gigabit ports.

Results

- Code for switching, firewalling with parallel learning is generated and compared with a COTS switch.



- Puget server has comparable link speed performance as compared to COTS devices (less than 100Mbps slower),
- The main advantage is that the data-plane software is correct by construction.
- Code is guaranteed to satisfy the specification with no undesirable behaviours.

Recommendation for Future Works

- Development of other elementary behaviour SFAs to aid in building data-planes with more functions such as a load balancer.
- Code synthesis algorithm can be automated to ease the translation from SFAs representation to actual, deployable code.

NOVEMBER 2019

NUS
National University of Singapore