Enhancing OpenFlow Actions to Offload Packet-In Processing

H. Farhady, P. Du, A. Nakao

U-Tokyo

NV 研究会 2014

Introduction

- Software-Defined Networking (SDN)
  - Not a new concept (e.g., Active Nets)
  - Provides a couple of APIs such as OpenFlow
  - Focuses on the programmability of the control plane
  - Data plane proposal is hardware-centric

- Cons
  - Programmability of the data plane? (e.g., Southbound APIs)

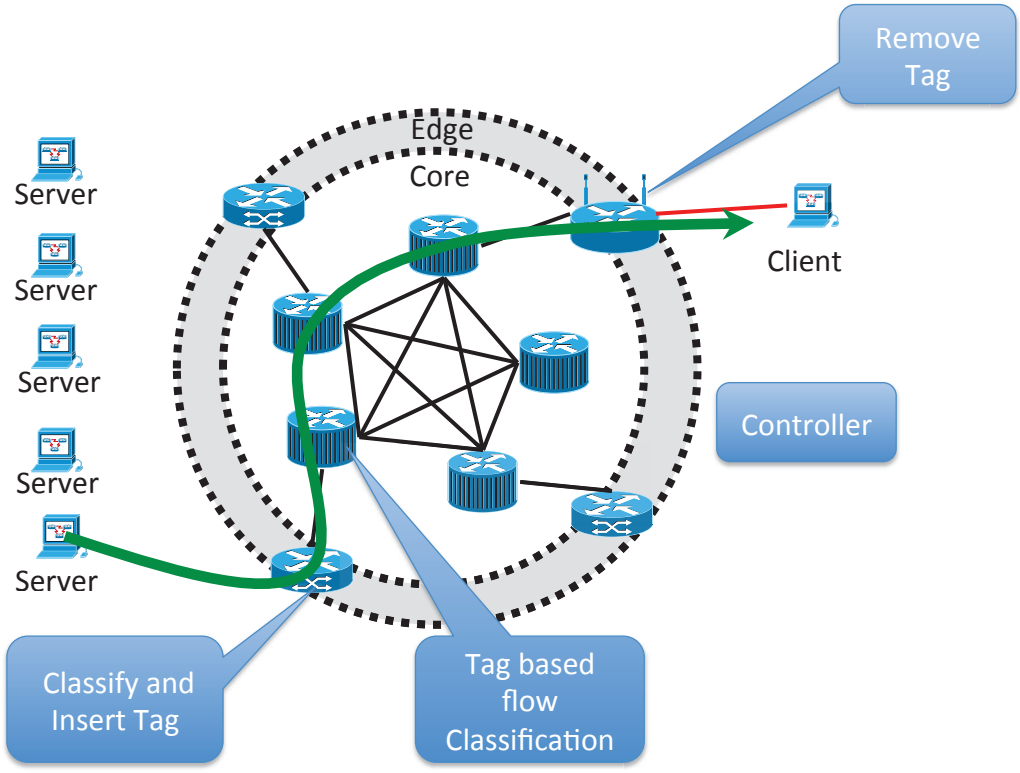
- Contribution
  - To relax the D-plane programmability constrains of OpenFlow

OpenFlow D-plane Constraints

- OpenFlow: *<match, action>* processing
  - Matching fields: Predefined (e.g., Source/Dest. IP)
  - Actions: Predefined (e.g., Drop, Forward)
  - Have to replace switch for every OpenFlow Update
  - TCAM = hardware-centric data plane
- TagFlow[1] a tag based classification/forwarding method
  - User-defined matching fields
    - Tagging packets at the network edge
    - Switching using labels at the core
- User-defined actions
  - Adding arbitrary actions to the switch

[1] Hamid Farhadi, Akihiro Nakao, "TagFlow: Efficient Flow classification in SDN", accepted in IEICE Transaction on Communications, 2014.

TagFlow Architecture



Main objectives of TagFlow

- Free the core from classification load
  - Offload the classification to the edge
  - Apply one-field classification at the core
- Use the freed capacity for
  - Application layer classification at the edge
  - Running User-Defined Action at the core

Evaluation

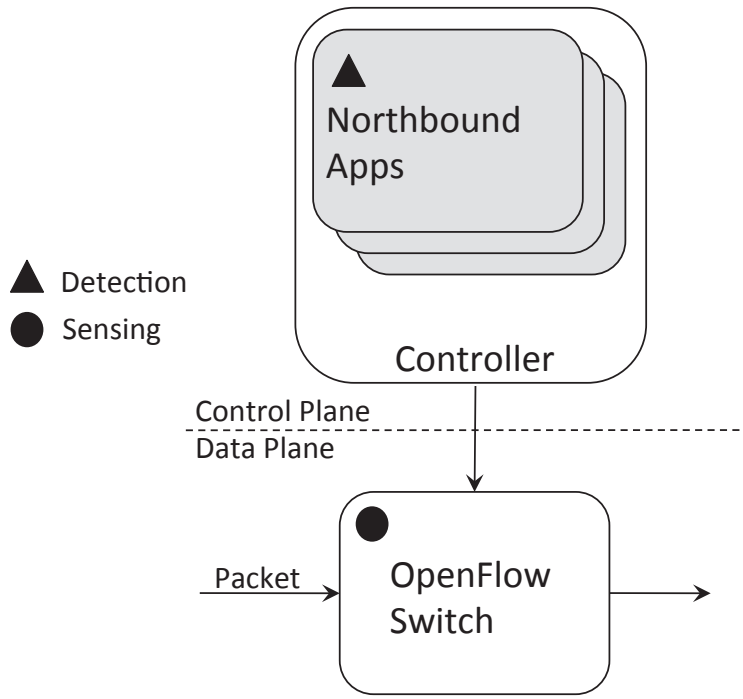
- PC based Experiments
  - To show the functionality
- FLARE based Experiments
  - To show the performance

Sample User-defined Actions

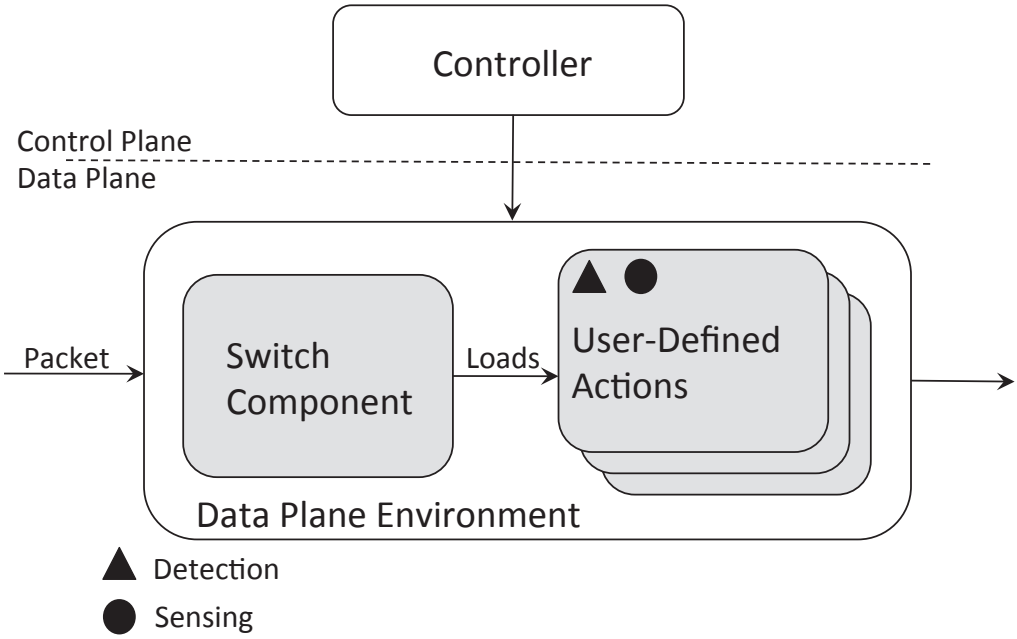
- Portscan detector(PSD)
  - Detects machines scanning other machines
- BotMiner detector(BHD) [1]
  - Uses PSD to detects machines scanning other machines
  - Co-clusters scanner nodes against low traffic nodes
- P2P Plotter [2]
  - Co-clusters long lasting nodes against low traffic nodes

[1] Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A Clean Slate 4D Approach to Network Control and Management. In Proceedings of ACM Computer Communications Review, 2005
[2] T. Yen and M. K. Reiter. Are Your Hosts Trading or ploting? Telling P2P File-sharing and Bots Apart. In Proceedings of IEEE ICDCS, 2010

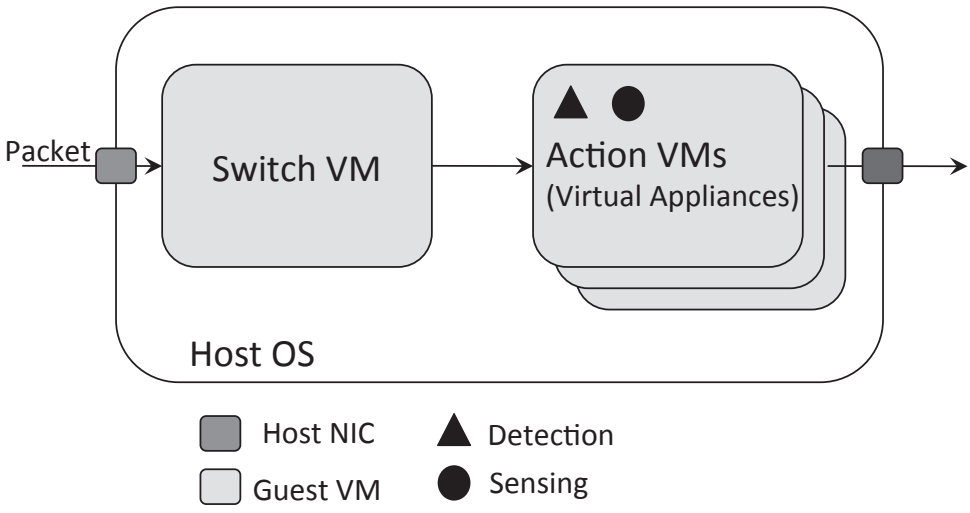
Architecture 1: C-Plane app



Architecture 2: D-Plane



Architecture 3: NFV app



Overhead of User-Defined Actions

	SDN		NFV
	C-plane Overhead (ms)[1]	D-plane Overhead (ms)	Virtual Appliance (ms)
Port scanner Detector	7.196	0.000001	0.001280609
BotMiner Detector	15.421	0.000004	0.001630215
P2P Plotter	11.775	0.000004	0.001312178

Overhead of C-plane apps are in milli-second scale, NFV appliances are in micro-second scale and User-Defined Action are in nano-second

[1] Seungwon Shin et al. "FRESCO: Modular Composable Security Services for Software-Defined Networks." NDSS'13, USA, 2013.

Ease of Programmability

Algorithm	**Middlebox**[1] (C LoC)	**C-plane app** NOX [1] (Python LoC)	**D-plane UDA (C + config LoC)**
TRW-CB[2]	1060	741	196 (181+15)
Rate Limit[3]	991	814	225 (205 + 20)

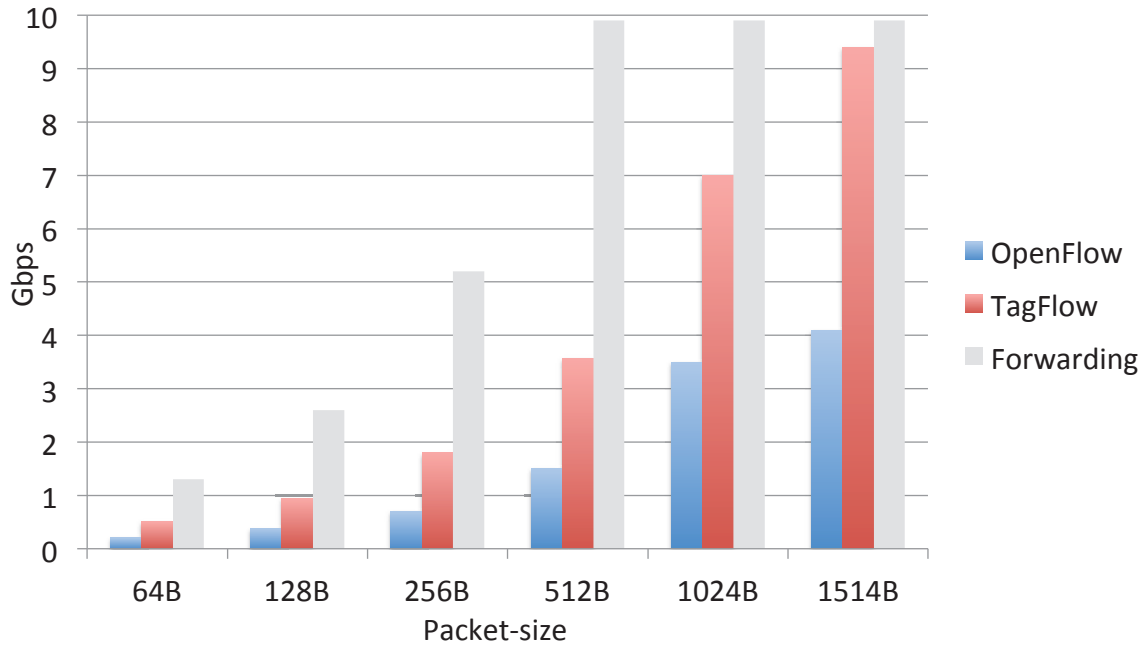
D-Plane based User-Defined Actions are easier to program than their equivalent implementations on Middlebox and C-plane

[1] Revisiting Traffic Anomaly Detection Using Software Defined Networking. In Proceedings of Recent Advances in Intrusion Detection, 2011.
[2] Schechter, S.E., Jung, J., Berger, A.W.: Fast detection of scanning worm infections. In: RAID. pp. 59–81 (2004)
[3] Williamson, M.M.: Throttling viruses: Restricting propagation to defeat malicious mobile code. In: ACSAC (2002)

FLARE

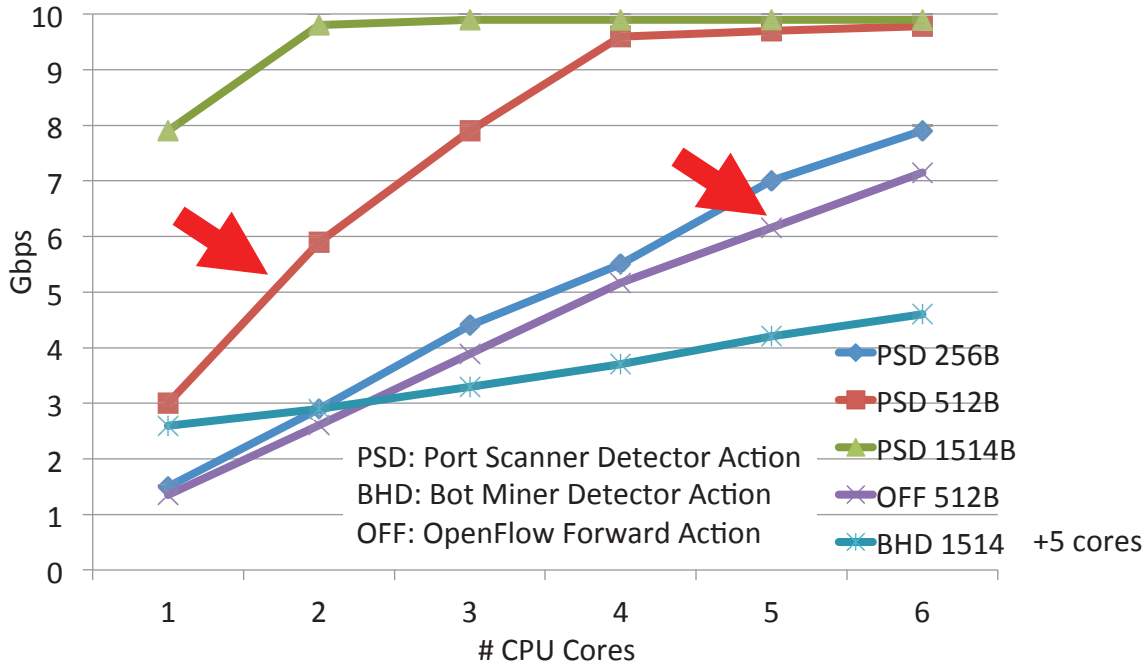
- Is a Fully Programmable Software Switch
- Combination of NFV/SDN
- Provides coexisting virtual programmable switches
- Click based programming Model
- Using many core general purpose processors
- 4 x 10G SFP+ ports
- Packet Generator: Xena Packet Generator

OpenFLow vs TagFlow (1-core)



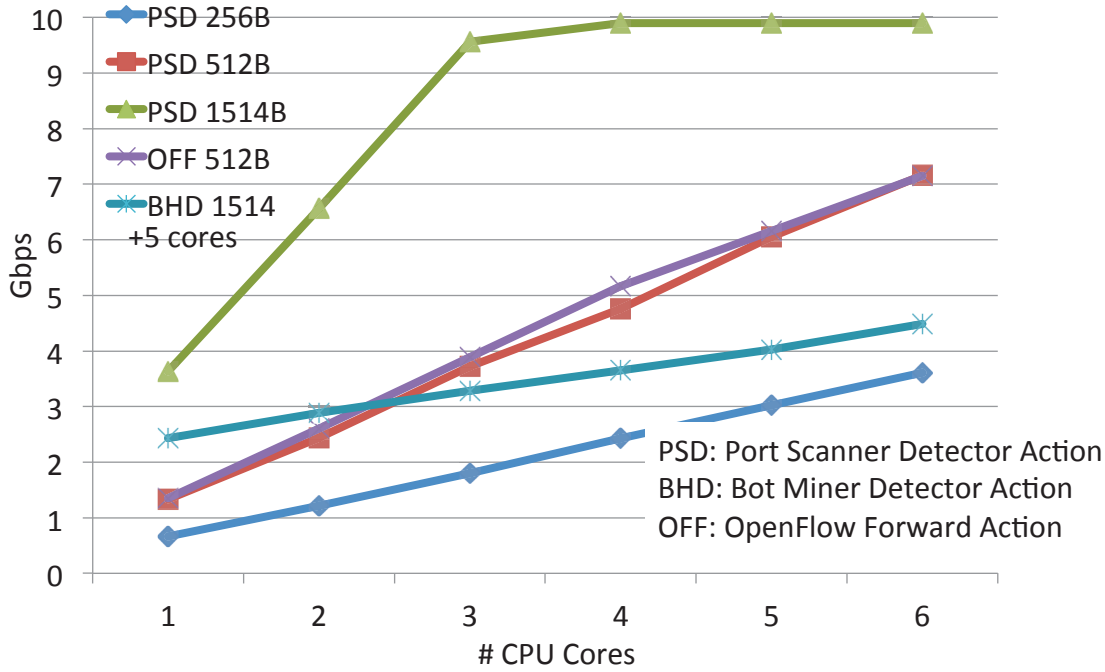
TagFlow forwarding is about 40% faster than OpenFlow at the core network

TagFlow + User-defined Actions



User-Defined Actions added to TagFlow switch can perform faster than OpenFlow forwarding

OpenFlow + User-defined Actions



User-Defined Actions added to OpenFlow software switch have minor performance degrade

Conclusion

- The is a need for a programmable D-plane
- SDN actions can be user-defined
- We showed user-defined actions are easy to program
- We showed user-defined action are fast enough to be considered as a feasible choice

Questions