OpenFlow DDoS Mitigation

C. Dillon, M. Berkelaar

February 9, 2014

University of Amsterdam
Quanza Engineering

Introduction

- Distributed Denial of Service attacks
- Types of attacks
  - Application layer attacks (low volume)
  - Network layer attacks (high volume)
- Popular mitigation methods
  - BGP Remotely Triggered Black Hole (RTBH)
  - In-line filtering appliances
  - Scrubbing center
- OpenFlow DDoS mitigation
  - While keeping the target online

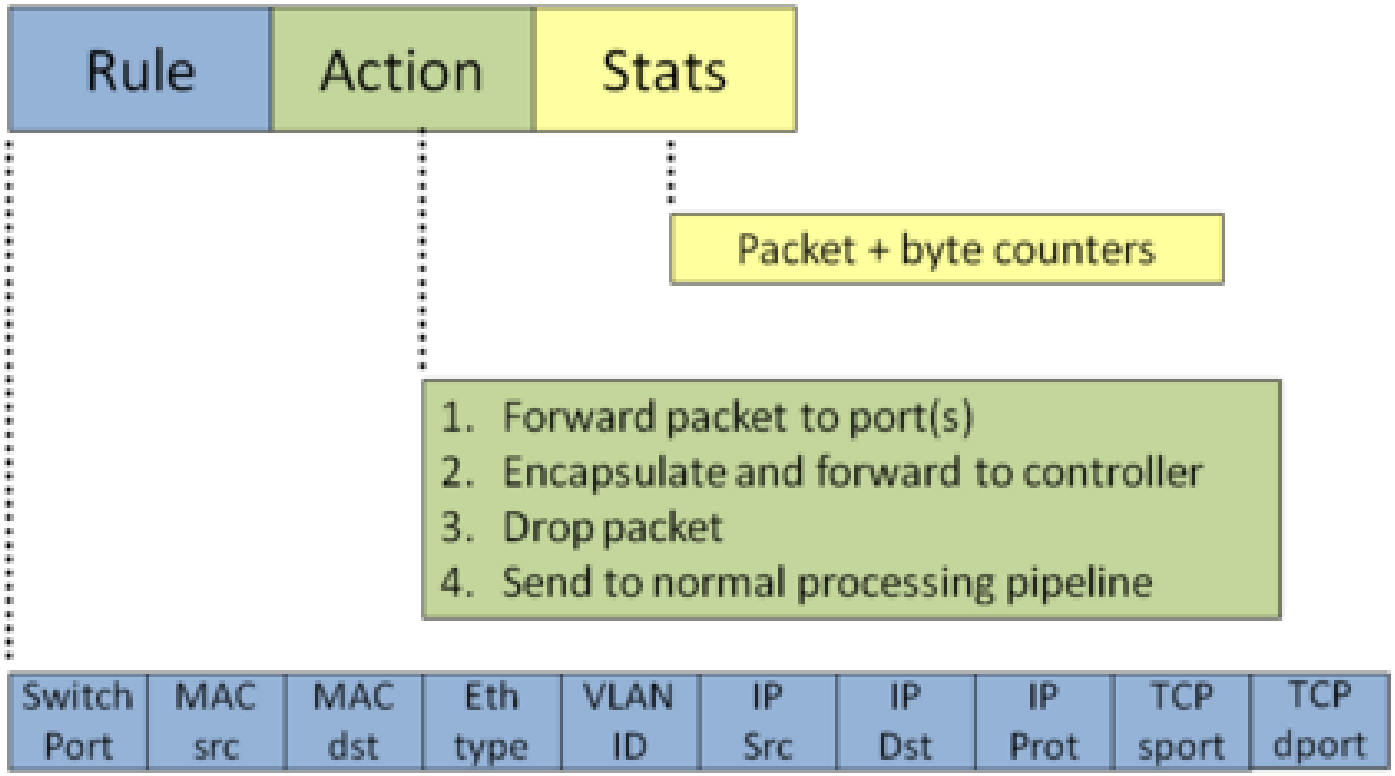
Research Question

How can Openflow be used in DDoS mitigation?

- How can flow statistics be analyzed to detect DDoS attacks?
- Can packet symmetry in sample traffic be analyzed to detect malicious traffic sources?
- Can malicious traffic sources be detected by temporarily dropping outgoing traffic?
- Can OpenFlow be used to efficiently block malicious sources while allowing legitimate traffic?

OpenFlow

- Separation between control- and data plane
- Controller creates and pushes flows to data plane
- TCAM table



[0] http://yuba.stanford.edu/cs244wiki/index.php/Overview

OpenFlow: Flow Statistics

- Per flow:
  - Duration
  - Byte counters
  - Packet counters
- Polled by controller
- Network load overview

OpenFlow: Traffic Sampling

- Packet-in channel
  - Samples to controller
  - Strip payload
  - Encapsulation by switch
  - TCP stream
- Mirroring
  - Multiple output ports for a flow
  - To any IDS on the network

OpenFlow: Traffic Dropping

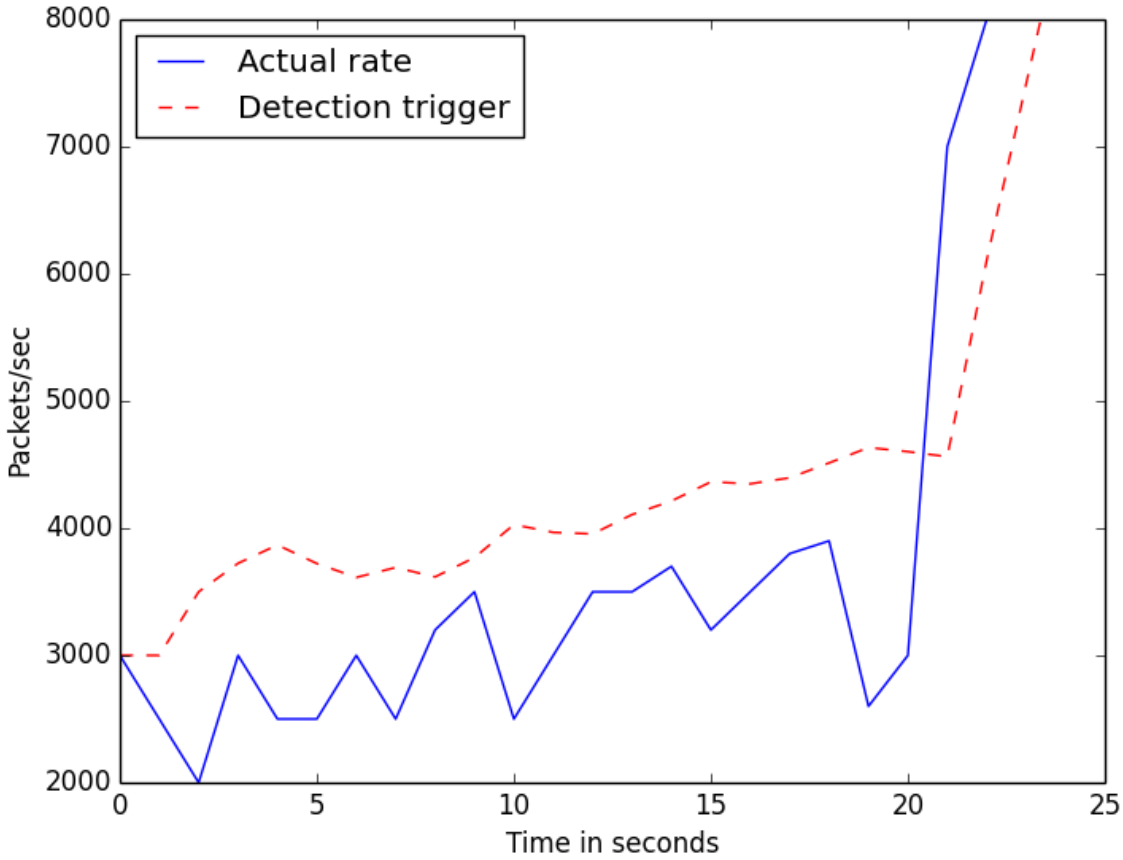
- Flexibility in dropping traffic:
  - Source based *blocking*
  - Destination based *filtering*
    - Only block TCP/UDP destination port
- Limited by capacity of TCAM table

Proposed Solution

1. Initial detection
   - Monitoring flow statistics
   - Detect traffic spikes
2. Identification of attackers
   - Traffic sampling
   - Packet symmetry
   - Block outgoing traffic
3. Blocking the attack
   - Drop traffic from malicious sources

Proposed Solution: Initial Detection

- Detection of traffic spikes in flow statistics
  - Detection based on the standard deviation
  - Lightweight
  - Initial detection: Used to trigger further detection mechanisms



Proposed Solution: Packet Symmetry

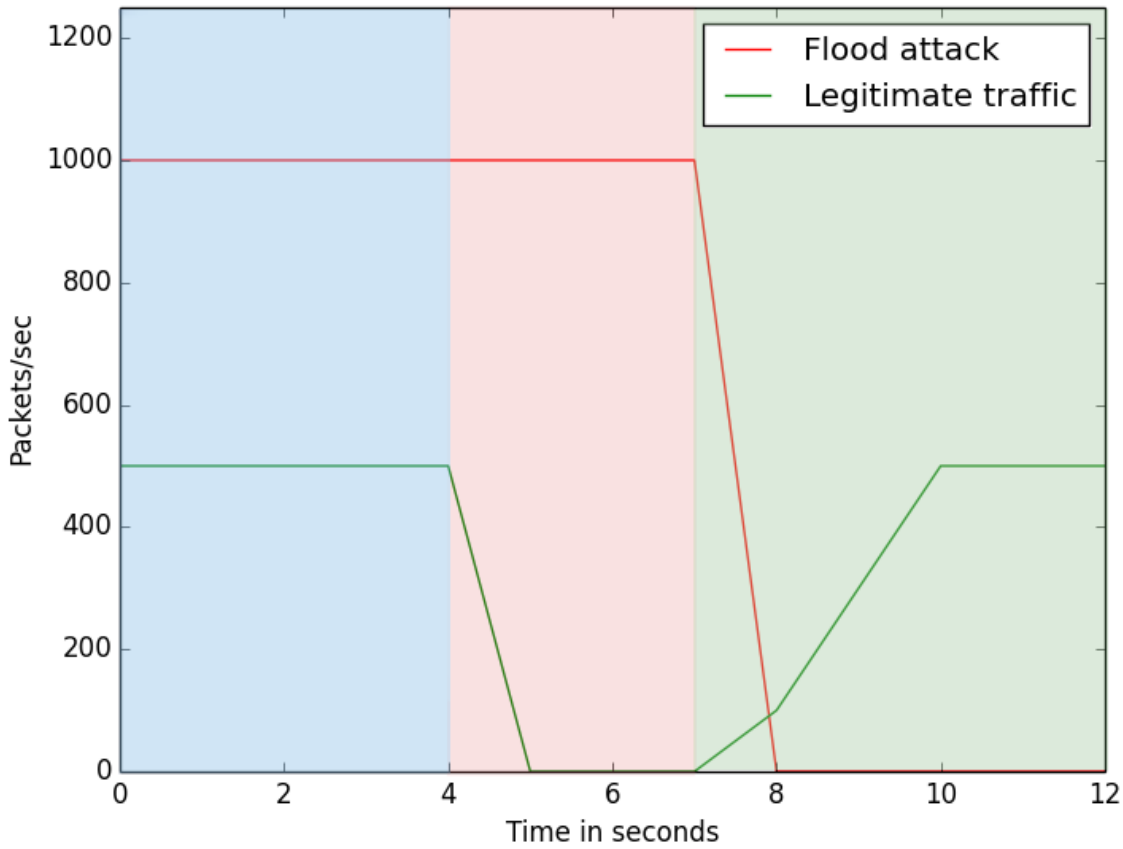
- Mirror traffic from and to DDoS target
- Distinguish attackers with packet count symmetry analysis
  - Legitimate traffic shows typical ratios between 1:1 and 8:1.

Proposed Solution: Block Outgoing Traffic

- A short interruption of the outgoing flow could distinguish bad sources.
  - TCP retransmit interval should increases
  - Typical request-response protocols may show equal behaviour
- Expecting a declining rate of packets
- OpenFlow can easily and rapidly modify flows that enable this

Proposed Solution: Block Outgoing Traffic

1. Sample
2. Block + sample
3. Analyse



Proposed Solution: Drop Malicious Traffic

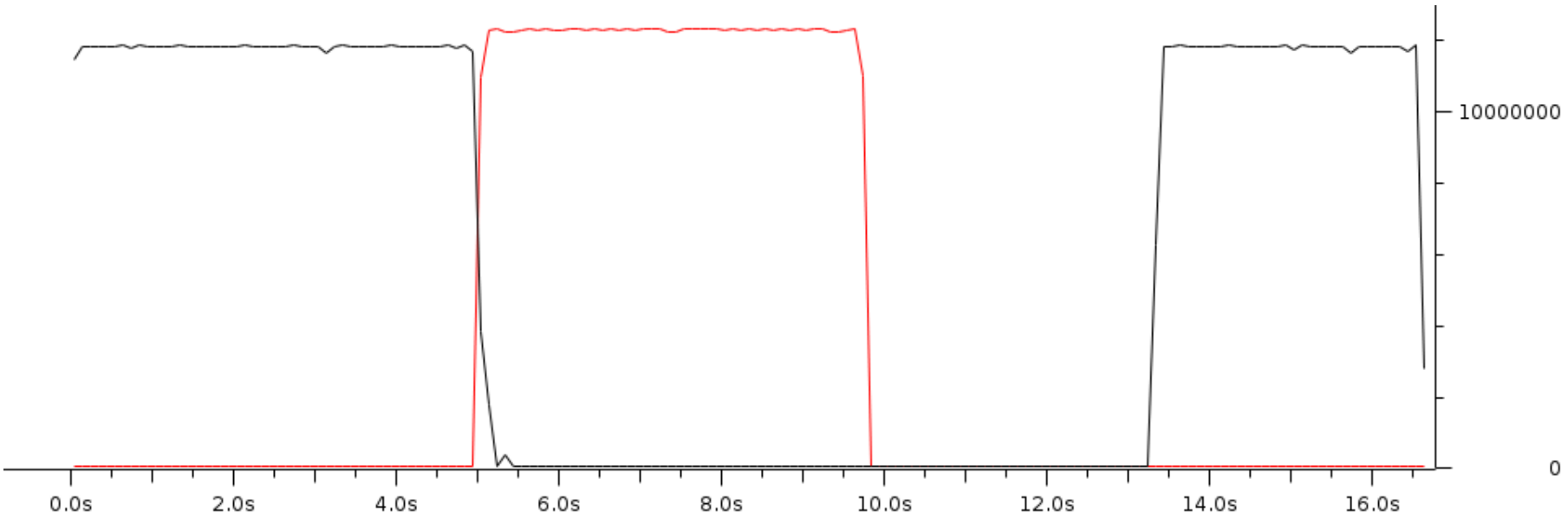
- Explicit drop flows using OpenFlow
  - Source-based blocking explored
  - Idle drop flows expire automatically

Proof of Concept: Experimentation setup

- Ryu SDN framework
  - Python based OpenFlow controller
  - Detection mechanisms in the controller
- Software environment
  - KVM + OpenVswitch
- Hardware environment
  - Arista 7050 OpenFlow switch
  - 10Gbit simulations
  - Not as flexible as OpenVswitch
- Traffic simulation
  - Victim and Attacker machines
  - Legitimate + DDoS traffic

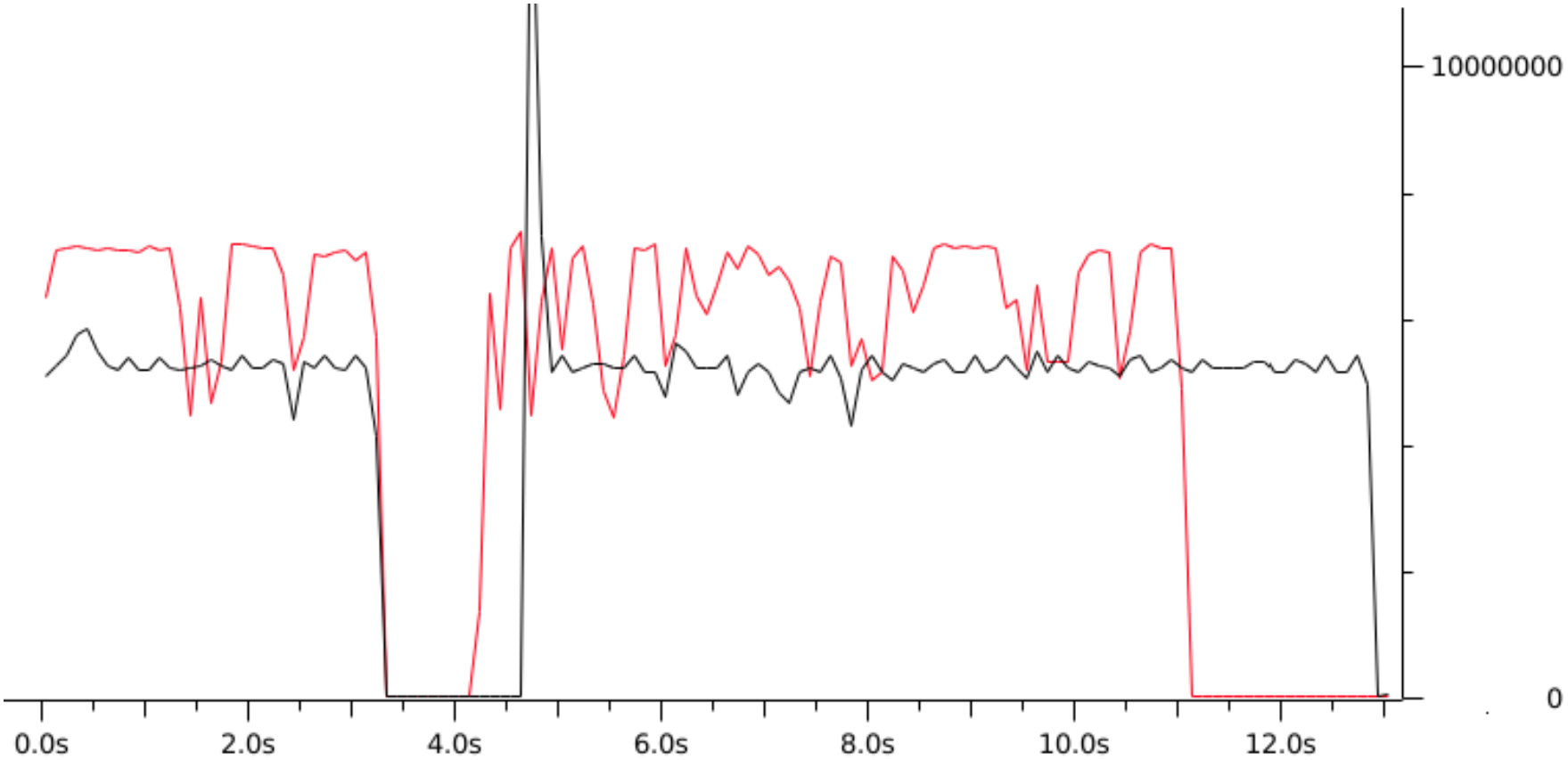
Proof of Concept: Packet Symmetry

- Hping3 flood stalls the Curl



Proof of Concept: Block Outgoing Traffic

- Timing issues with hardware.
- Flood never stopped. Curl retransmitted at a declining rate.



Conclusion

- Using the OpenFlow infrastructure to mitigate high volume attacks shows potential.
- Hardware currently shows limitations:
  - TCAM table size
  - Timing of OpenFlow operations in our experiment caused issues

Questions

Questions?