Design and Development of Network Monitoring Strategies in P4-enabled Programmable Switches

Damu Ding

Supervisor: Dr.Domenico Siracusa (FBK)

Co-supervisors: Dr.Marco Savi and Dr.Federico Pederzolli

University of Bologna, Bologna, Italy

FONDAZIONE BRUNO KESSLER

Ph.D. defense

18th May, 2021

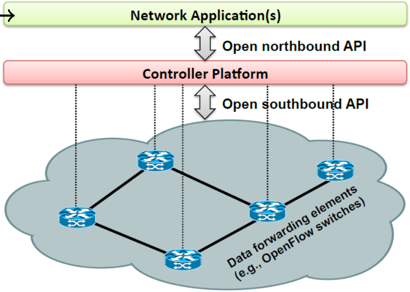
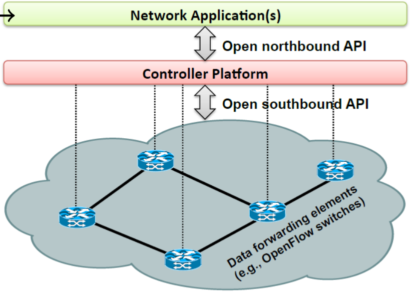


Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

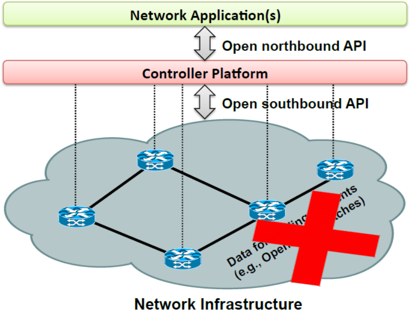
Network monitoring functionalities in Software-Defined Networks



1. Significant communication overhead
2. The latency caused by interaction
3. Cannot perform monitoring at line-rate speed (Up to 100 Gbps)

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

Network monitoring functionalities in Software-Defined Networks

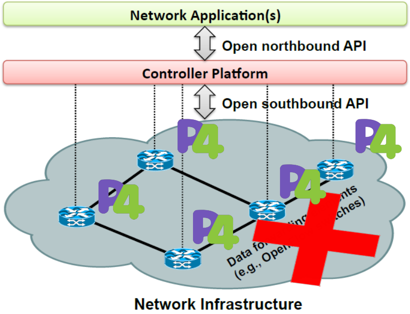


Data plane programmable switches

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

Damu Ding damu.ding2@unibo.it

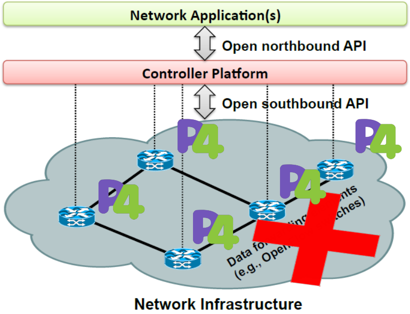
Network monitoring functionalities in Software-Defined Networks



Data plane programmable switches

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

Network monitoring functionalities in Software-Defined Networks



Data plane programmable switches

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

Network monitoring functionalities in Software-Defined Networks

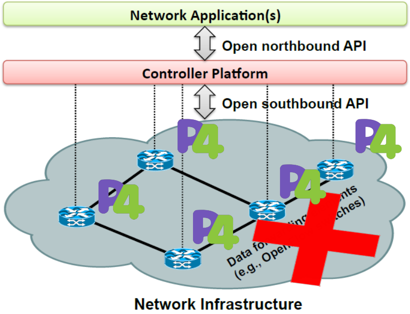
Heavy-hitter detection

Flow cardinality estimation

Network traffic entropy estimation

Traffic volume estimation

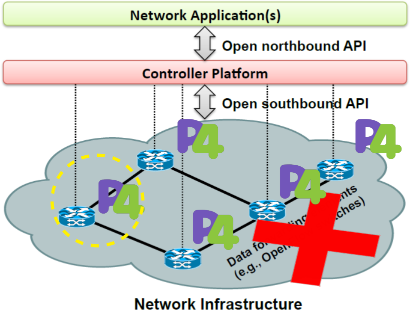
Volumetric DDoS detection



Data plane programmable switches

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

Network monitoring functionalities in Software-Defined Networks



Heavy-hitter detection

Flow cardinality estimation

Network traffic entropy estimation

Traffic volume estimation

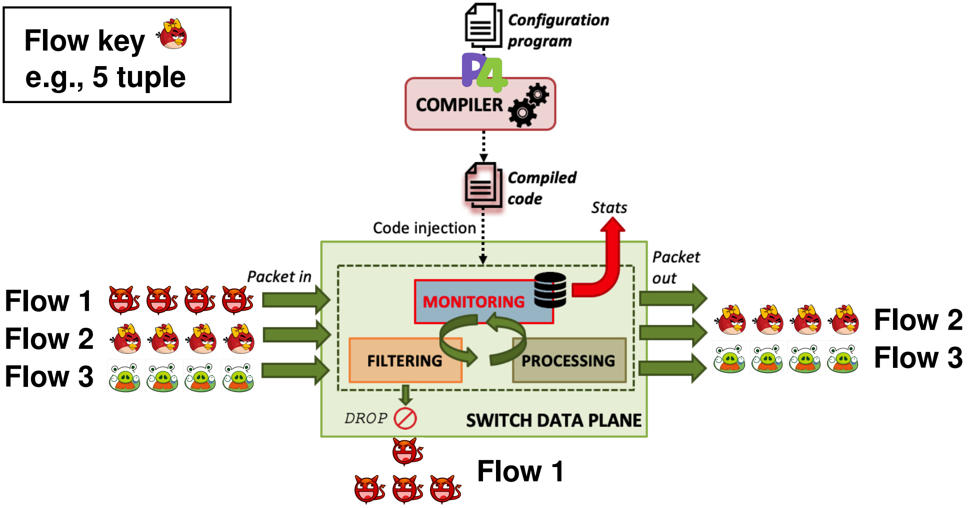
Volumetric DDoS detection

Data plane programmable switches

Figure source: Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76. and https://n0where.net/real-time-network-monitoring-cyberprobe

P4-enabled programmable data plane for monitoring

Flow key
e.g., 5 tuple



Motivation

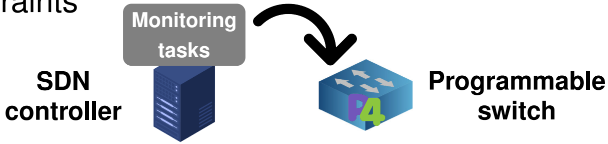
However

Network monitoring tasks in literature cannot be directly offloaded to programmable switch data plane

- Limited hardware resource (e.g. memory)
- Computational constraints to assure fast packet processing

Goal

Design and develop new strategies for specific monitoring tasks in P4-enabled programmable data planes considering the switch constraints



Outline

Part 1
Network-wide heavy-hitter detection

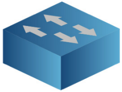
Part 2
Normalized network traffic entropy-based volumetric DDoS detection

Part 3
Per-flow cardinality-based volumetric DDoS detection

Network-wide heavy-hitter detection

Heavy-hitter detection

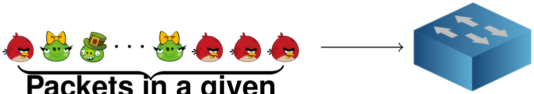
Packets in a given time interval



Heavy hitter

ID	#pkts
	2000
	60
	40

Heavy-hitter detection



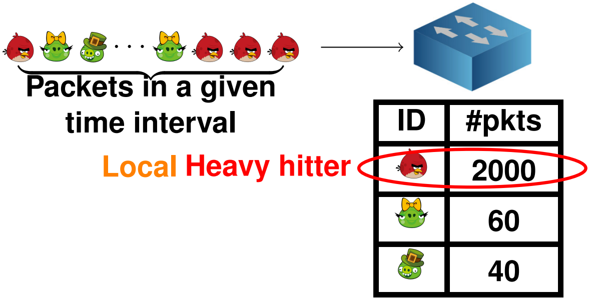
Packets in a given time interval

Heavy hitter

ID	#pkts
	2000
	60
	40

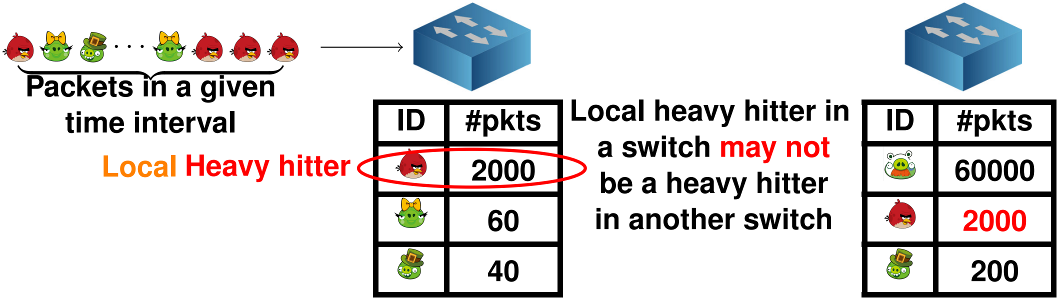
- **Heavy-hitter detection**: identifies the flows that contain more than **a fraction of total packets (i.e. a threshold)** in a given time interval
- **Applications**: DoS (Denial of Service) and anomaly detection, flow-size aware routing, and Quality of Service (QoS) management.

Heavy-hitter detection



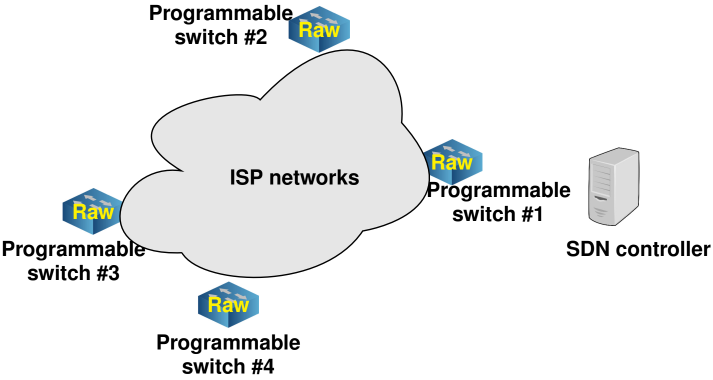
- **Heavy-hitter detection**: identifies the flows that contain more than **a fraction of total packets (i.e. a threshold)** in a given time interval
- **Applications**: DoS (Denial of Service) and anomaly detection, flow-size aware routing, and Quality of Service (QoS) management.

Heavy-hitter detection



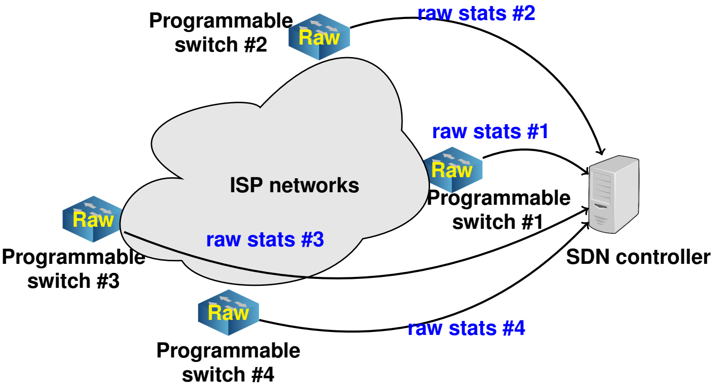
- **Heavy-hitter detection**: identifies the flows that contain more than **a fraction of total packets (i.e. a threshold)** in a given time interval
- **Applications**: DoS (Denial of Service) and anomaly detection, flow-size aware routing, and Quality of Service (QoS) management.

Network-wide heavy-hitter detection



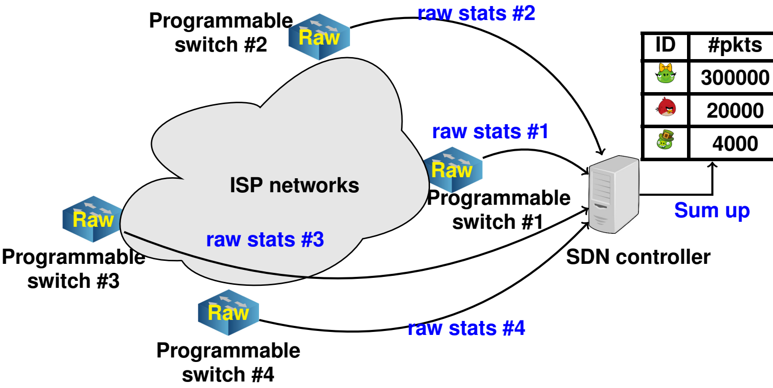
Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Network-wide heavy-hitter detection



Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Network-wide heavy-hitter detection SOTA

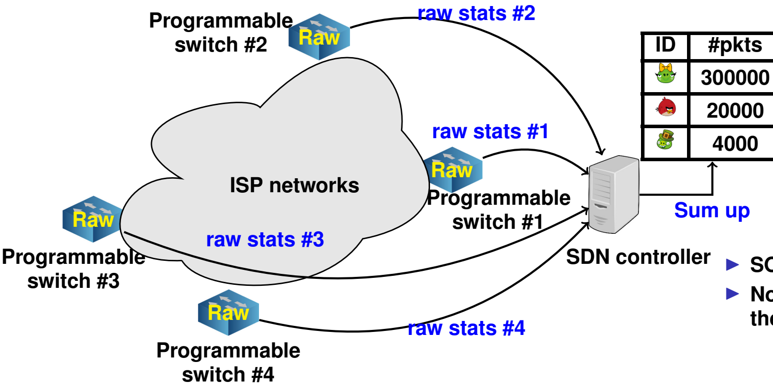


ID	#pkts
	300000
	20000
	4000

Network-wide heavy-hitter detection: identifies the flows that contain more than a fraction of overall number of packets in the network

Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Network-wide heavy-hitter detection



ID	#pkts
	300000
	20000
	4000

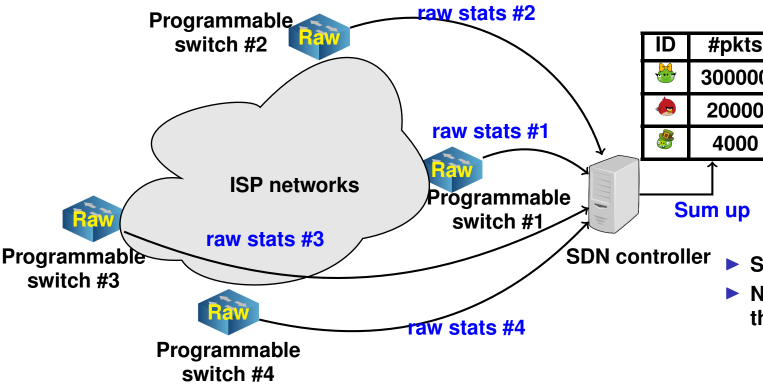
Network-wide heavy-hitter detection: identifies the flows that contain more than a fraction of overall number of packets in the network

Limitations:

- SOTA stores raw data in the switch
- Not easy to define a threshold for the sum of flow packet counts

Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Network-wide heavy-hitter detection



ID	#pkts
	300000
	20000
	4000

Network-wide heavy-hitter detection: identifies the flows that contain more than a fraction of overall number of packets in the network

Limitations:

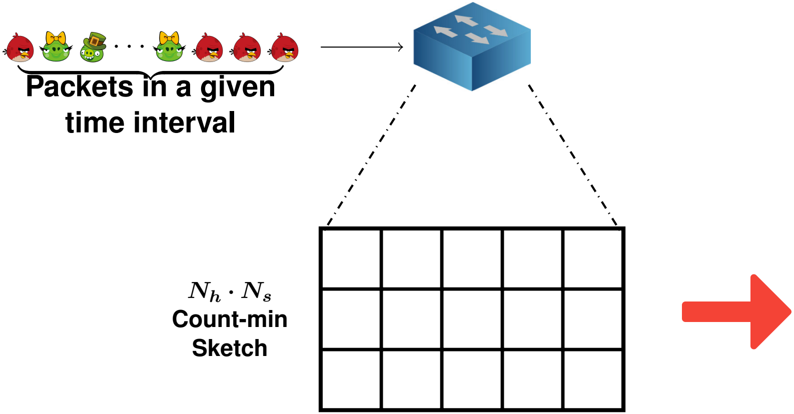
- SOTA stores raw data in the switch
- Not easy to define a threshold for the sum of flow packet counts

- **RQ1**: How to efficiently collect flow statistics in the switch?
- **RQ2**: How to accurately merge flow statistics in the controller?

Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Count-min Sketch (CMS)

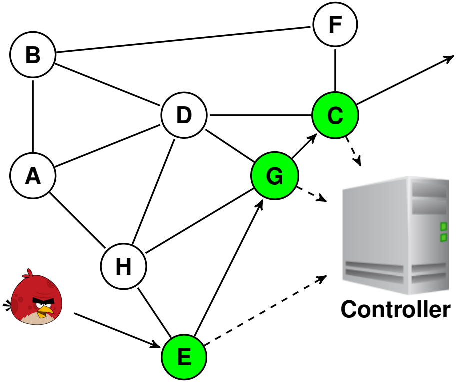
Count-min Sketch is a memory-efficient data structure to store flow statistics



ID	#pkts
	60000
	4000
	200

N_h : Number of hash functions, N_s : Output size of hash functions

Packet double counting problem



Node E	
ID	#pkts
	5000
	2000
	200

Node G	
ID	#pkts
	200
	120
	60

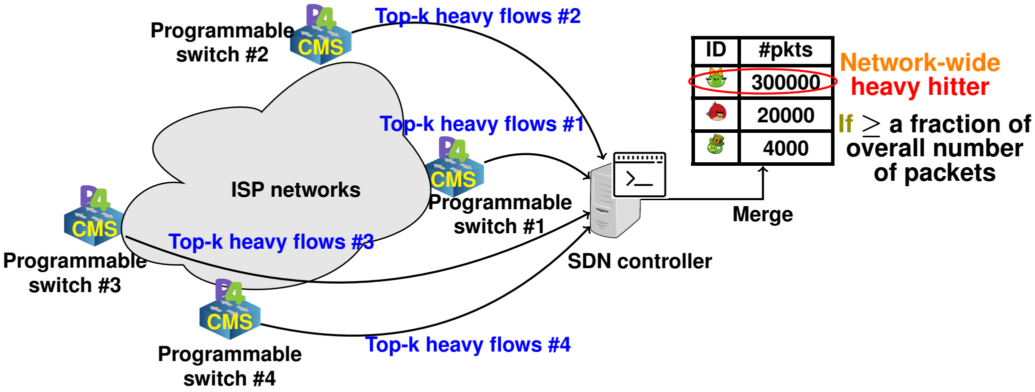
Node C	
ID	#pkts
	4000
	200
	120

Merge

Controller			
ID	#pkts	ID	#pkts
	5000		4000
	2000		200
	120		60

Basat, Ran Ben, et al. "Network-wide routing-oblivious heavy hitters." Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems. 2018.

Network-wide heavy-hitter detection (NWHHD+)

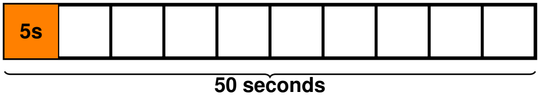


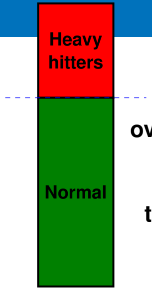
Damu Ding, Marco Savi, Gianni Antichi, and Domenico Siracusa. *Incremental deployment of programmable switches for network-wide heavy-hitter detection.* IEEE Conference on Network Softwarization (NetSoft) 2019.

Damu Ding, Marco Savi, Gianni Antichi, and Domenico Siracusa. *An incrementally-deployable P4-enabled architecture for network-wide heavy-hitter detection.* IEEE Transactions on Network and Service Management (TNSM) 17.1 (2020): 75-88.

Evaluation settings

Normal traffic (CAIDA 2018)

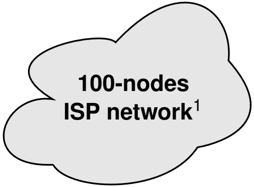




$\geq 0.05\%\cdot$ **overall number of packets in the time interval**

$$Recall = \frac{TP}{TP+TN} \qquad Precision = \frac{TP}{TP+FP}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



- Each node has the same probability to be an ingress or egress point
- Each packet is forwarded from the ingress point to the egress point following the shortest path

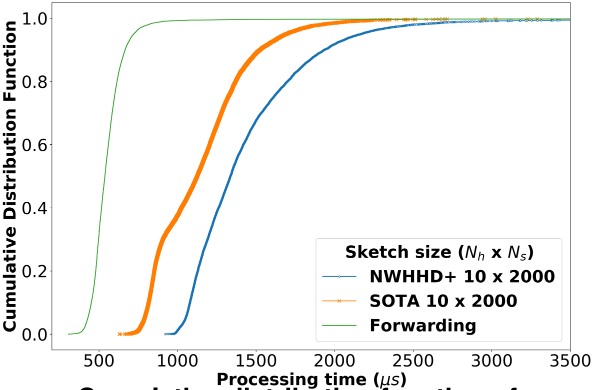
[1] https://sites.uclouvain.be/defo

$$TP = Count^{detected/true}_{Heavyhitter},\ FP = Count^{detected/false}_{Heavyhitter},\ TN = Count^{undetected/true}_{Heavyhitter}$$

Simulation and emulation results

Evaluation metrics	SOTA [2]	NWHHD+
F1 score	0.821	0.907
Communication overhead*	71877	60354
Occupied memory*	760042	60255

*Measurement units	ID	#pkts
		2000



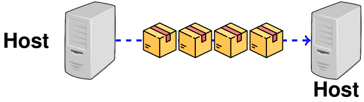
Cumulative distribution function of packet processing time in mininet (10000 packets)

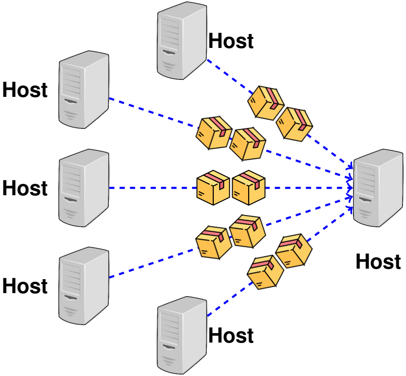
[2] Harrison, Rob, et al. "Network-Wide Heavy Hitter Detection with Commodity Switches." Proceedings of the Symposium on SDN Research, 2018.

Normalized network traffic entropy-based DDoS detection

Normalized network traffic entropy

Normalized network traffic entropy H_{norm} indicates network traffic distribution

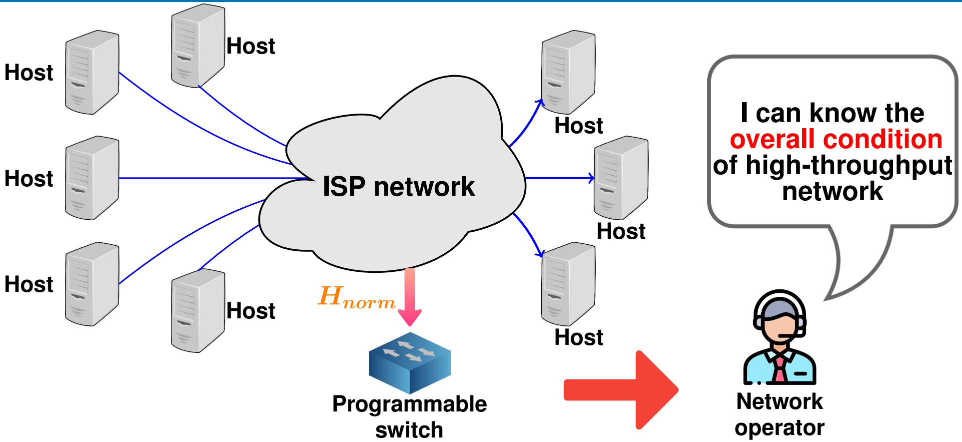




$$H_{norm} = 1$$

$$H_{norm} = 0$$

Normalized network traffic entropy in programmable switches

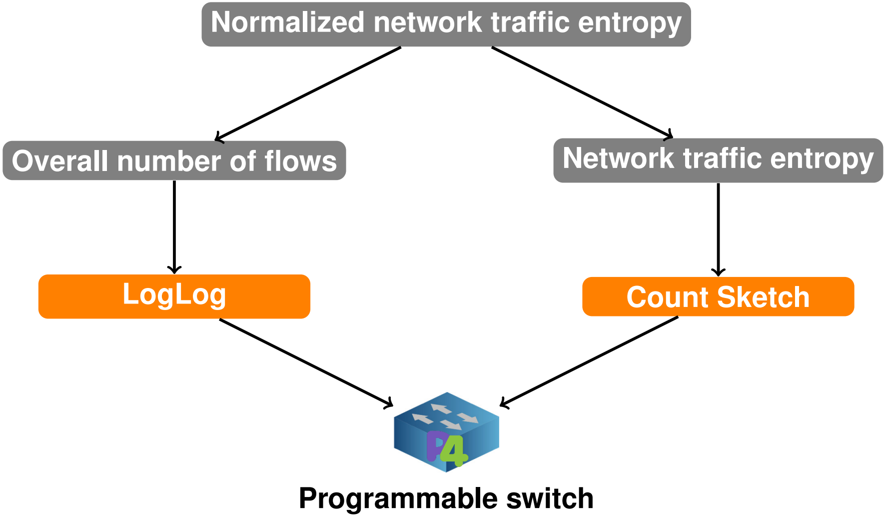


Normalized Shannon entropy

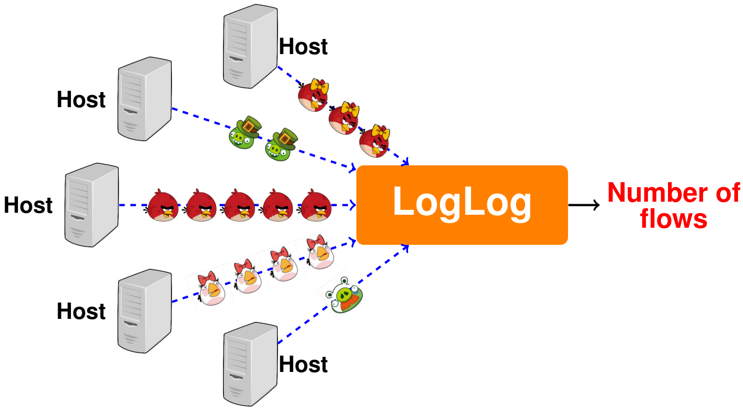
$$H_{norm} = \frac{-\sum_{i=1}^{n_{tot}} \frac{f_i}{S_{tot}} \log_2 \frac{f_i}{S_{tot}}}{\log_2 n_{tot}}$$

- f_i: Packet count of flow i
- S_{tot}: Overall number of packets ✓
- n_{tot}: Overall number of flows

Normalized network traffic entropy



Estimate overall number of flows

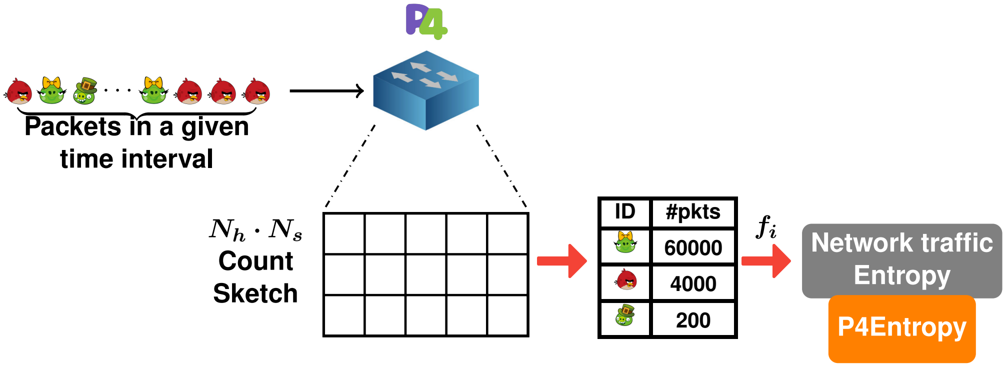


- **Fast:** Time complexity is only O(1)
- **Efficient and accurate:** 2560 bytes can estimate 10^9 numbers with standard error 2%.
- **Implementable in P4**

Durand, Marianne, and Philippe Flajolet. "Loglog counting of large cardinalities." European Symposium on Algorithms. Springer, Berlin, Heidelberg, 2003.

	Count Sketch	Count-min Sketch
Bias of the flow packet count estimation	Low	Relatively high
Heavy hitter detection	Good	Good
Network traffic entropy estimation	Good	Bad
Update speed	1x	2x

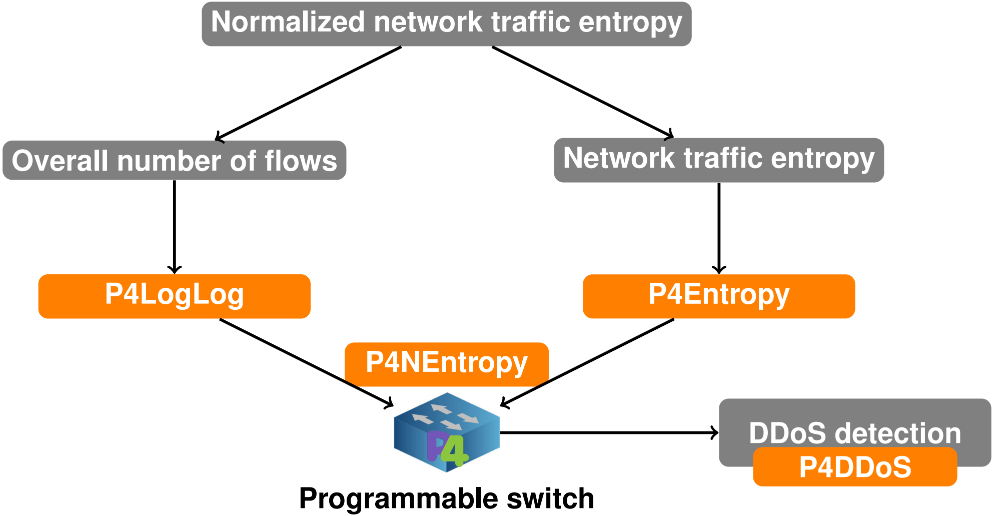
Network traffic entropy



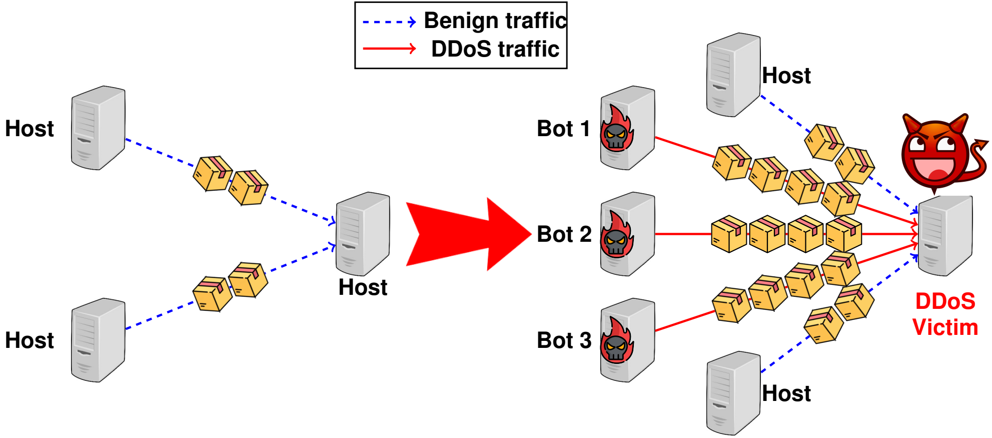
N_h : Number of hash functions, N_s : Output size of hash functions

Damu Ding, Marco Savi, and Domenico Siracusa. *Estimating logarithmic and exponential functions to track network traffic entropy in P4.* IEEE/IFIP Network Operations and Management Symposium (NOMS) 2020.

Normalized network traffic entropy-based DDoS detection



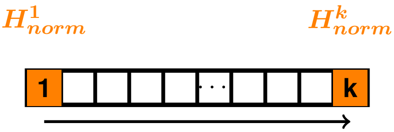
Property of volumetric DDoS attacks



H_{norm} **of destination IPs**

Exponentially weighted moving average (EWMA) of normalized entropy

Legitimate traffic



k time intervals

$$EWMA^1_{norm} = H^1_{norm}$$
$$EWMA^2_{norm} = \alpha H^1_{norm} + (1-\alpha)H^2_{norm}$$
$$\cdots$$
$$EWMA^k_{norm} = \alpha H^{k-1}_{norm} + (1-\alpha)H^k_{norm}$$

DDoS threshold

$$\lambda^k_{norm} = EWMA^k_{norm} - \epsilon$$

If $H^{k+1}_{norm} < \lambda^k_{norm} \rightarrow$ **DDoS attacks** $\rightarrow$ **Trigger an alarm**

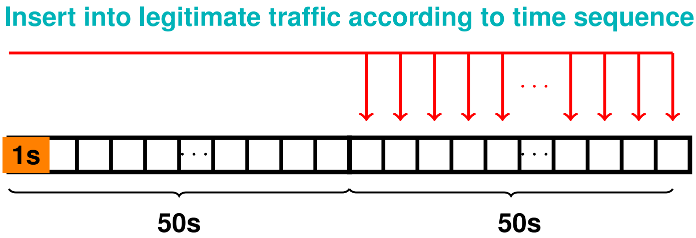
Else $EWMA^{k+1}_{norm} = \alpha H^k_{norm} + (1-\alpha)H^{k+1}_{norm}$

Damu Ding, Marco Savi, and Domenico Siracusa. *Tracking Normalized Network Traffic Entropy to Detect DDoS Attacks in P4* submitted to IEEE Transactions on Dependable and Secure Computing (TDSC).

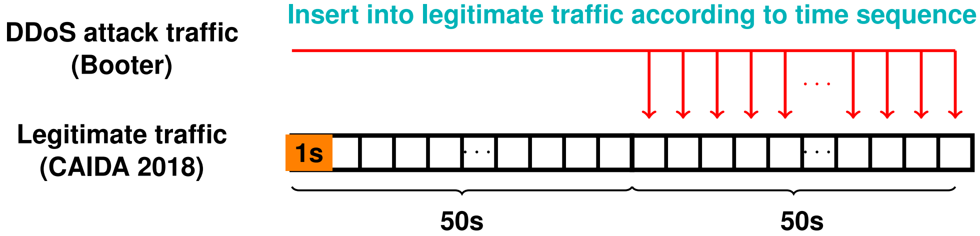
Evaluation settings

DDoS attack traffic
(Booter)

Legitimate traffic
(CAIDA 2018)



Evaluation settings

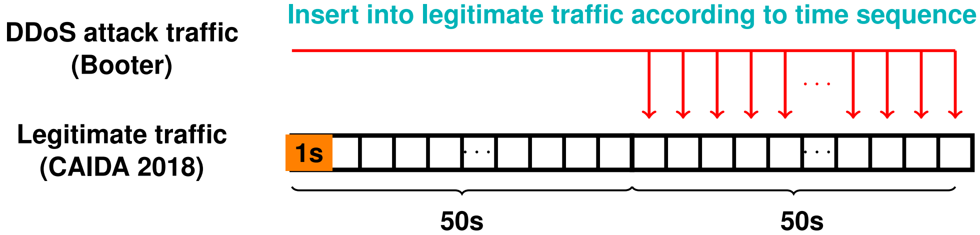


DDoS trace name	Packets per second	Attack source IPs
Booter 6	~ 90000	7379
Booter 7	~ 41000	6075
Booter 1	~ 96000	4486
Booter 4	~ 80000	2970

DNS-amplification DDoS attacks

Booter is a class of on-demand services
that provide illegal support to launch DDoS attacks
targeting websites and networks.

Evaluation settings



	DDoS	NO DDoS
Alarm	True Positive (TP)	False Positive (FP)
NO alarm	False Negative (FN)	True Negative (TN)

$$D_{tp} = \frac{\#Time\ intervals[TP]}{\#Time\ intervals[TP+FN]} \qquad D_{fp} = \frac{\#Time\ intervals[FP]}{\#Time\ intervals[TN+FP]}$$

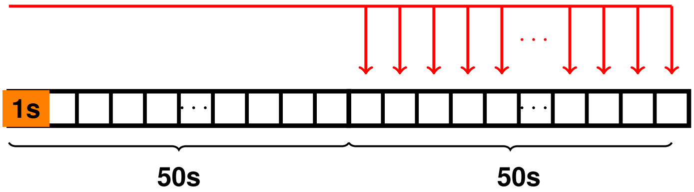
$$D_{acc} = \frac{\#Time\ intervals[TP+TN]}{\#Time\ intervals[TP+TN+FP+FN]}$$

Evaluation settings

Insert into legitimate traffic according to time sequence

DDoS attack traffic (Booter)

Legitimate traffic (CAIDA 2018)

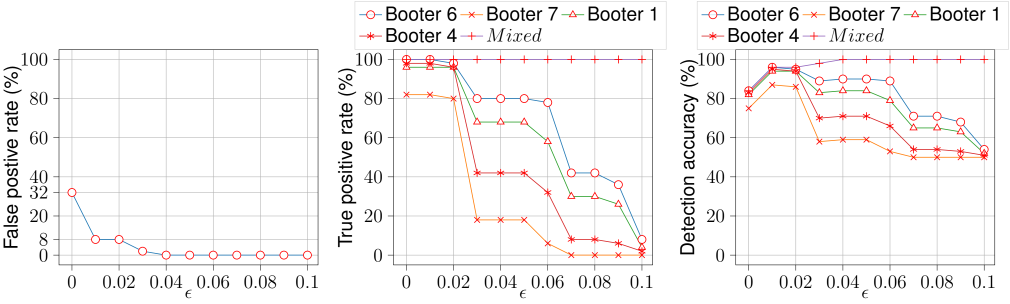


$$EWMA_{norm}^{50}(\alpha = 0.13)$$

$$\lambda_{norm}^{51} = EWMA_{norm}^{50} - \epsilon$$

Maximize DDoS detection accuracy

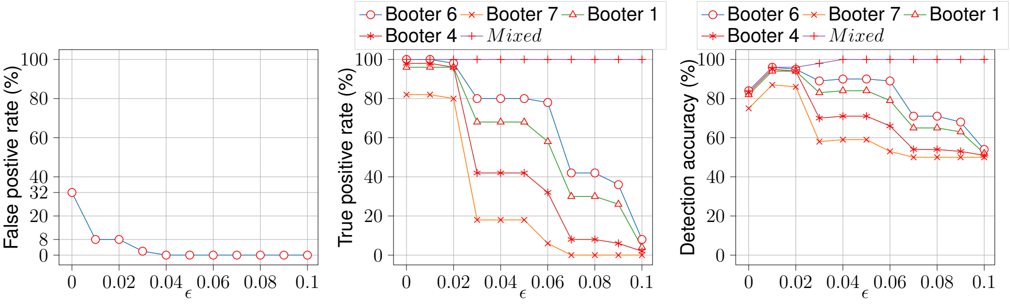
Configuring DDoS detection threshold



(a) Sensitivity to D_{fp} (b) Sensitivity to D_{tp} (c) Sensitivity to D_{acc}

- **Minimize** false positive rate D_{fp} ($\epsilon \in [0.01, 0.1]$)
- **Maximize** true positive rate D_{tp} ($\epsilon \in [0, 0.02]$)
- **Maximize** detection accuracy D_{acc} ($\epsilon \in [0.01, 0.02]$)

Configuring DDoS detection threshold

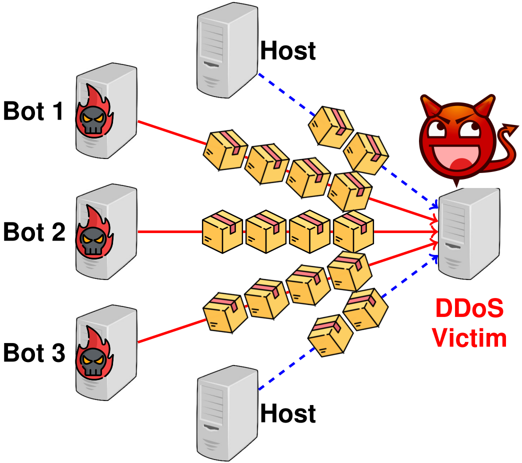


(a) Sensitivity to D_{fp} (b) Sensitivity to D_{tp} (c) Sensitivity to D_{acc}

- **Minimize** false positive rate D_{fp} ($\epsilon \in [0.01, 0.1]$)
- **Maximize** true positive rate D_{tp} ($\epsilon \in [0, 0.02]$)
- **Maximize** detection accuracy D_{acc} ($\epsilon \in [0.01, 0.02]$)

$$\epsilon = \mathbf{0.01}$$

State of the art (SOTA_DDoS)



Network traffic entropy
of **source IPs** H_{src} **increases** $\Uparrow$
OR
Network traffic entropy
of **destination IPs** H_{dst} **decreases** $\Downarrow$

Limitations:

- Spoofed source IPs
- Flow fluctuations
- Needs power-hungry TCAM memory to compute entropy

Lapolli, Angelo Cardoso, Jonatas Adilson Marques, and Luciano Paschoal Gaspary. "Offloading real-time ddos attack detection to programmable data planes." 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019.

Comparing to SOTA

Algorithm	False-positive rate D_{fp}	True-positive rate D_{tp} / Detection accuracy D_{acc}				
		Booter 6	Booter 7	Booter 1	Booter 4	Mixed
P4DDoS	8%	100% / 96%	82% / 87%	96% / 94%	98% / 95%	100% / 96%
SOTA_DDoS [3]	10%	100% / 95%	74% / 82%	100% / 95%	94% / 92%	100% / 95%

Booter name	PPS	Attack sources
Booter 6	$\sim$ 90000	7379
Booter 7	$\sim$ 41000	6075
Booter 1	$\sim$ 96000	4486
Booter 4	$\sim$ 80000	2970

[3] Lapolli, Angelo Cardoso, Jonatas Adilson Marques, and Luciano Paschoal Gaspary. "Offloading real-time ddos attack detection to programmable data planes." 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019.

Comparing to SOTA

Algorithm	False-positive rate D_{fp}	True-positive rate D_{tp} / Detection accuracy D_{acc}				
		Booter 6	Booter 7	Booter 1	Booter 4	Mixed
P4DDoS	8%	100% / 96%	82% / 87%	96% / 94%	98% / 95%	100% / 96%
SOTA_DDoS [3]	10%	100% / 95%	74% / 82%	100% / 95%	94% / 92%	100% / 95%

And ...

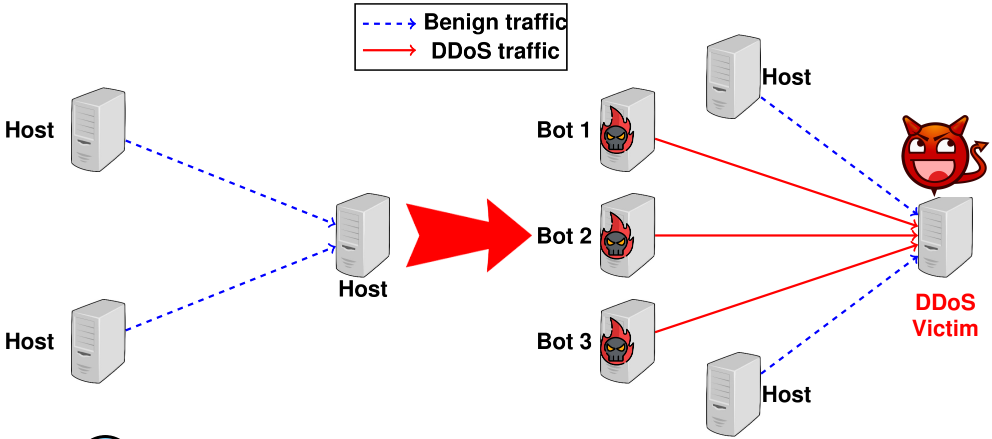
- No need to use power-hungry TCAM memory
  - Only relies on P4-supported operations
- Much simpler, i.e., lower implementation complexity
  - Only relies on normalized entropy of destination IPs
- Robust to the flow fluctuations in different time intervals
  - Normalized entropy instead of only entropy

Booter name	PPS	Attack sources
Booter 6	$\sim$ 90000	7379
Booter 7	$\sim$ 41000	6075
Booter 1	$\sim$ 96000	4486
Booter 4	$\sim$ 80000	2970

[3] Lapolli, Angelo Cardoso, Jonatas Adilson Marques, and Luciano Paschoal Gaspary. "Offloading real-time ddos attack detection to programmable data planes." 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019.

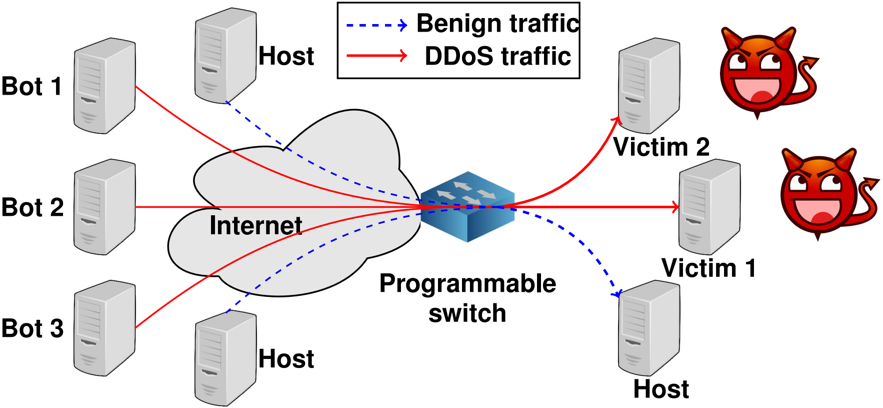
Per-flow cardinality-based DDoS detection

Property of volumetric DDoS attacks



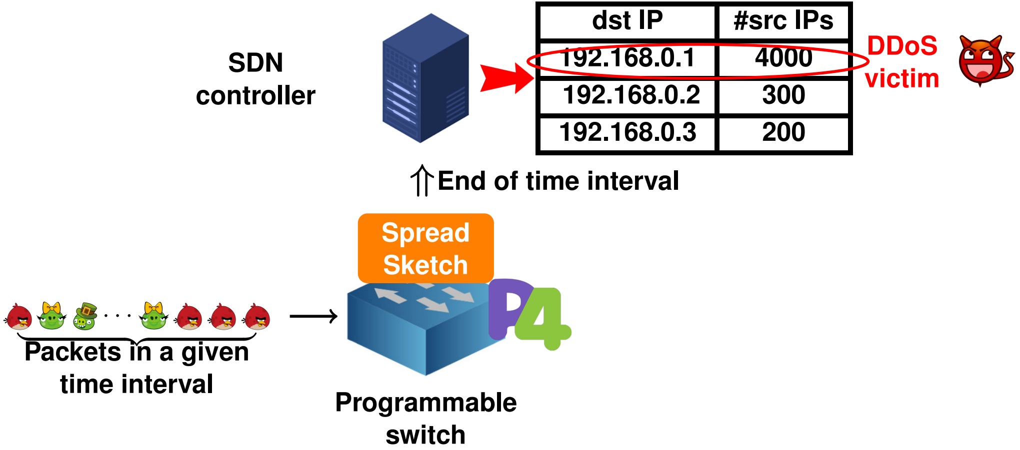
When a DDoS attack is taking place, the number of distinct flows (i.e. flow cardinality) significantly increases

Threat model and deployment scenario



A memory-efficient data structure to count the number of flows targeting different destinations in the programmable switch is necessary

Spread Sketch



dst IP	#src IPs
192.168.0.1	4000
192.168.0.2	300
192.168.0.3	200

Tang, Lu, Qun Huang, and Patrick PC Lee. "Spreadsketch: Toward invertible and network-wide detection of superspreaders." IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.

Spread Sketch

SDN controller

dst IP	#src IPs
192.168.0.1	4000
192.168.0.2	300
192.168.0.3	200

DDoS victim

⇑ End of time interval

Spread Sketch

Programmable switch

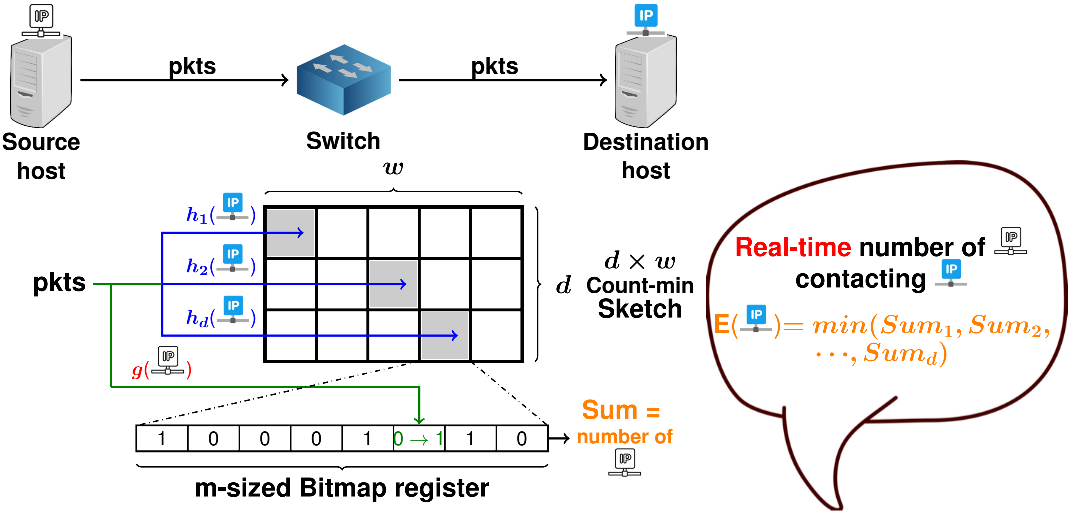
Packets in a given time interval

Limitations:

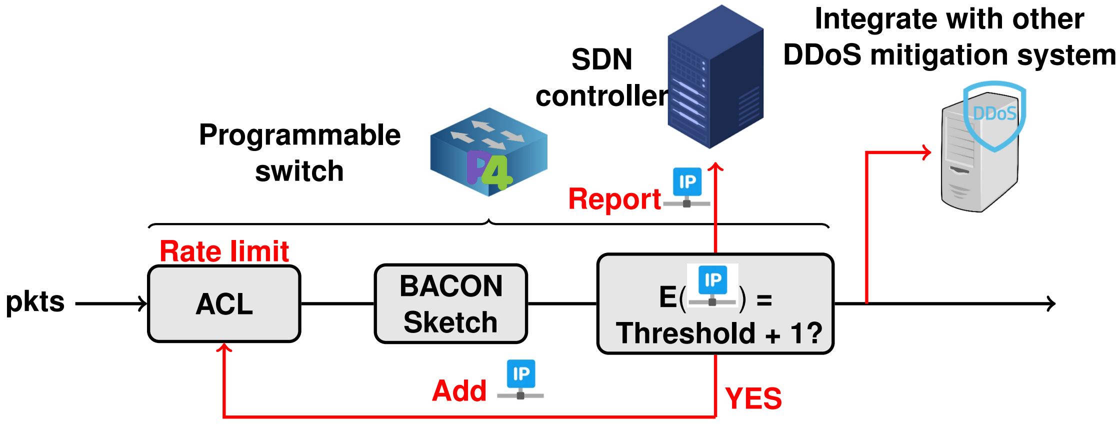
- Cannot estimate #flows in real time
- Cannot detect DDoS victims in the switch directly
- Incurs communication overhead and latency

Tang, Lu, Qun Huang, and Patrick PC Lee. "Spreadsketch: Toward invertible and network-wide detection of superspreaders." IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.

BACON Sketch



In-network DDoS victim identification (INDDoS)



Damu Ding, Marco Savi, Federico Pederzolli, Mauro Campanella, and Domenico Siracusa. *In-Network Volumetric DDoS Victim Identification Using Programmable Commodity Switches* IEEE Transactions on Network and Service Management (TNSM).

Programmable hardware switch

32x 100Gbps QSFP ports

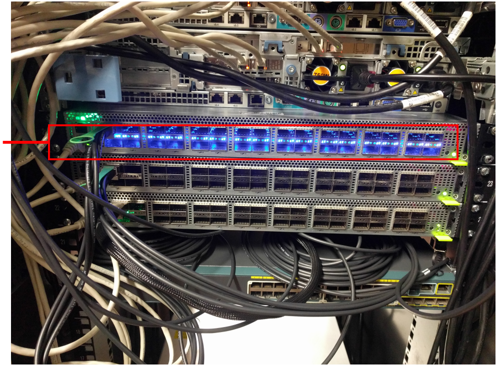


Figure: Edgecore Wedge-100BF-32X switch equipped with Barefoot Tofino ASIC in FBK's lab

Pros:

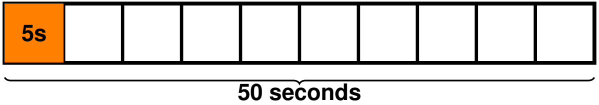
1. Higher monitoring throughput

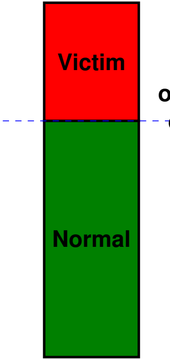
Cons:

1. Limited hardware resources
2. Computational constraints

Evaluation settings

Normal traffic
(CAIDA 2018)





$\geq 0.5\%\cdot$ overall number of source IPs in the time interval

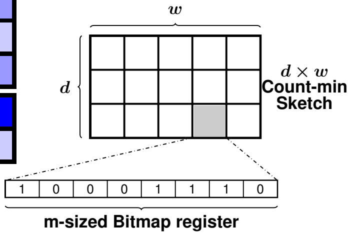
$$Recall = \frac{TP}{TP+TN} \qquad Precision = \frac{TP}{TP+FP}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

$TP = Count_{DDoSvictim}^{detected/true}$, $FP = Count_{DDoSvictim}^{detected/false}$, $TN = Count_{DDoSvictim}^{undetected/true}$

Sensitivity analysis of DDoS victim identification

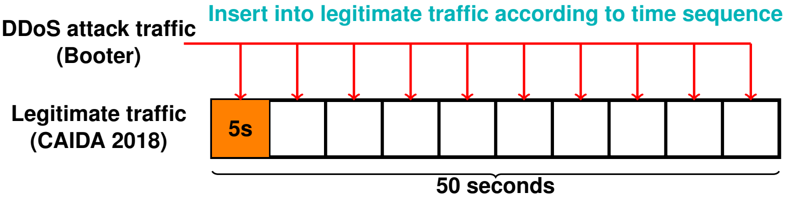
BACON Sketch size ($d \times w \times m$)	Recall	Precision	F1 score
$3 \times 1024 \times 1024$	0.96	0.99	0.97
$1 \times 2048 \times 1024$	0.98	0.54	0.70
$1 \times 1024 \times 2048$	0.94	0.38	0.54
$5 \times 1024 \times 512$	0.12	1.0	0.22
$5 \times 512 \times 1024$	0.96	0.89	0.92
Spread Sketch[4] size ($d \times w \times m$)	**Recall**	**Precision**	**F1 score**
$3 \times 1024 \times 1024$	0.92	0.94	0.93

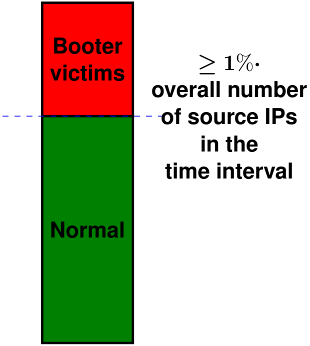


NB. Spread Sketch cannot be fully executed in programmable data planes

[4] Tang, Lu, Qun Huang, and Patrick PC Lee. "Spreadsketch: Toward invertible and network-wide detection of superspreaders." IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.

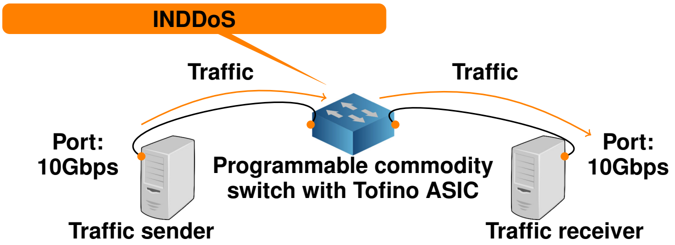
DDoS victim identification accuracy under Booter attacks





DDoS attack flow trace	Recall	Precision	F1 score
Booter 6	1.0 (1/1)	1.0 (1/1)	1.0 (1/1)
Booter 7	1.0 (1/1)	1.0 (1/1)	1.0 (1/1)
Booter 1	1.0 (1/1)	1.0 (1/1)	1.0 (1/1)
Booter 4	1.0 (1/1)	1.0 (1/1)	1.0 (1/1)
Mixed	1.0 (4/4)	1.0 (4/4)	1.0 (4/4)

Switch resource usage and processing time



Strategy	No. stages	SRAM	No. ALUs	PHV size	Throughput	Processing time w.r.t. Simple forwarding
Simple forwarding	16.67%	2.5%	4.2%	7.30%	10Gbps	-
INDDoS + Simple forwarding	100%	8.33%	56.25%	9.90%	10Gbps	100ns

Damu Ding damu.ding2@unibo.it

Lessons learned

	Per-flow cardinality-based DDoS detection	Network traffic entropy-based DDoS detection
High-packet-rate volumetric DDoS detection	✓	✓
Low-packet-rate volumetric DDoS detection	✓	✗
DDoS victim identification	✓	✗
Implementation complexity	Low	High

Research topics

Research topics	Contributions beyond SOTA	Publications
Network-wide heavy hitter detection	1. Used memory-efficient data structure to store flow statistics 2. Avoid packet double counting problem 3. More suitable threshold for network-wide heavy-hitter detection	1. **Damu Ding**, Marco Savi, Gianni Antichi, and Domenico Siracusa. *Incremental deployment of programmable switches for network-wide heavy-hitter detection.* IEEE Conference on Network Softwarization (NetSoft) 2019. 2. **Damu Ding**, Marco Savi, Gianni Antichi, and Domenico Siracusa. *An incrementally-deployable P4-enabled architecture for network-wide heavy-hitter detection.* IEEE Transactions on Network and Service Management (TNSM)
Normalized network traffic entropy-based DDoS detection	1. Only need P4-supported operations for DDoS detection 2. Lower implementation complexity 3. Robust to flow fluctuations in different time intervals	3. **Damu Ding**, Marco Savi, and Domenico Siracusa. *Estimating logarithmic and exponential functions to track network traffic entropy in P4* IEEE/IFIP Network Operations and Management Symposium (NOMS) 2020. 4. **Damu Ding**, Marco Savi, and Domenico Siracusa. *Tracking Normalized Network Traffic Entropy to Detect DDoS Attacks in P4* submitted to IEEE Transactions on Dependable and Secure Computing (TDSC)
Per-flow cardinality-based DDoS detection	1. Fully executed in hardware switch data plane 2. Detect DDoS attacks in real time 3. Low communication overhead	5. **Damu Ding**, Marco Savi, Federico Pederzolli, Mauro Campanella, and Domenico Siracusa. *In-Network Volumetric DDoS Victim Identification Using Programmable Commodity Switches* IEEE Transactions on Network and Service Management (TNSM).

Activities overview

- Training
  - Finished and passed Ph.D. courses (180 credits)
  - Attended Barefoot Academy “BA102: Introduction to data and control plane development with P4_16, Tofino ASIC and P4studio SDE”
- European project participation (GN 4-3 project[5])
  - Propose and develop new volumetric DDoS detection and mitigation strategies in programmable commodity switches for next-generation high speed ISP networks
  - Coordinate European collaborators for network performance evaluation
  - Publish project-related results in high-quality publications

[5] https://www.geant.org/Projects/GEANT_Project_GN4-3

Conclusion

- Offload monitoring tasks from SDN controller to data plane programmable switches leveraging various memory-efficient data structures
  - Count-min Sketch
  - LogLog counting
  - Count Sketch
  - and much more ...
- Focus on smart monitoring strategies in programmable data planes
  - Network-wide heavy-hitter detection
  - Normalized entropy-based volumetric DDoS detection
  - Per-flow cardinality-based volumetric DDoS detection
  - and much more ...
- Proved network monitoring performance using programmable switches
  - High monitoring accuracy
  - Low packet processing time for monitoring
  - Valid for high-throughput networks

Thank you!