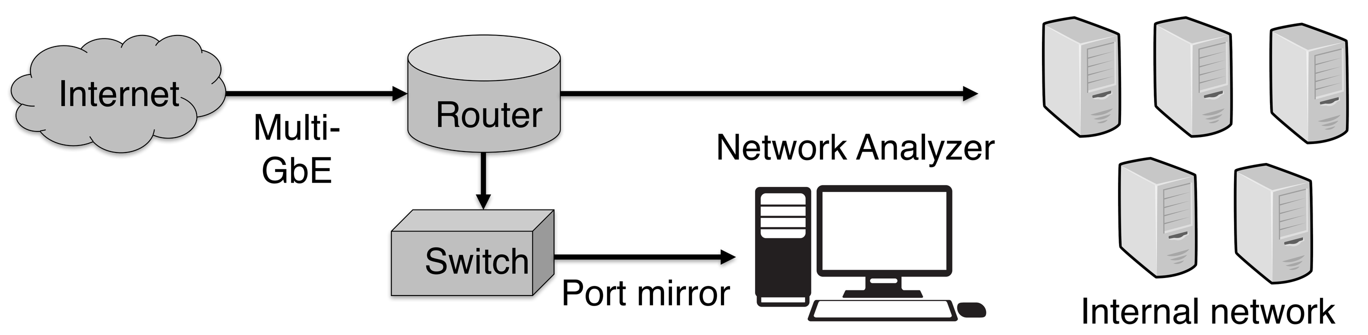
Deep Packet/Flow Analysis using GPUs

Qian Gong, Wenji Wu, Phil DeMar (Fermilab)

Motivation

- Network packet monitoring and deep packet analysis (DPA) are widely used to support applications such as intrusion detection, surveillance, traffic control and statistics gathering.

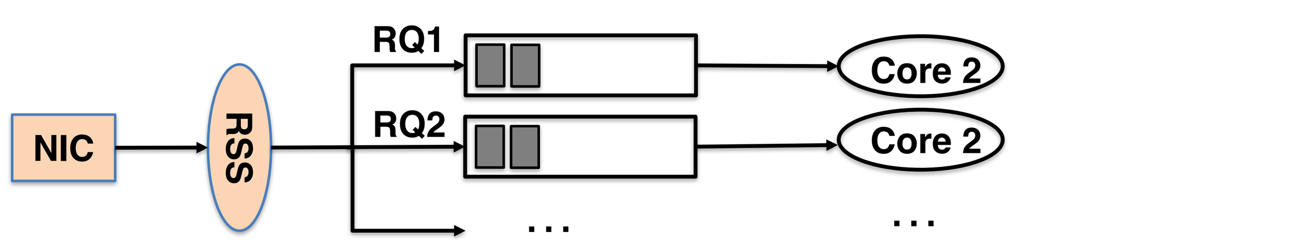


- The growing network traffic rate and rising sophistication in the type of network attacks are pushing for faster and more complex (e.g., flow-based) packet processing systems.
- GPUs work exceptionally well for network packet analysis; but a solid flow-based GPU deep packet inspection tool is missing.

Challenges

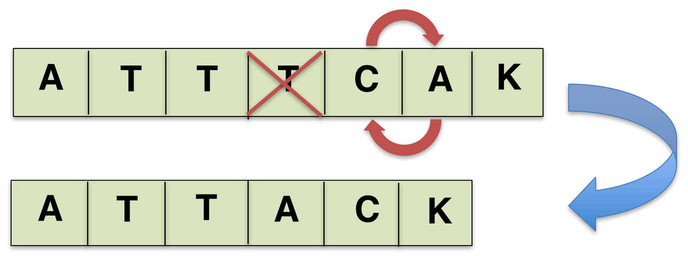
Challenges in Packet Capturing

- Single-threaded packet analysis system has limited performance, e.g., Snort: 0.2 Gbit/s when perform DPA
- Multi-queue NICs and multicore systems provide opportunities.

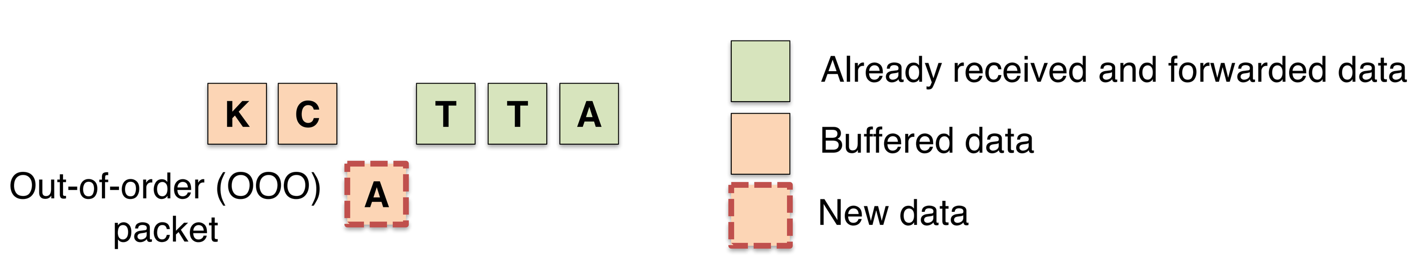


Challenges in Flow-based DPA

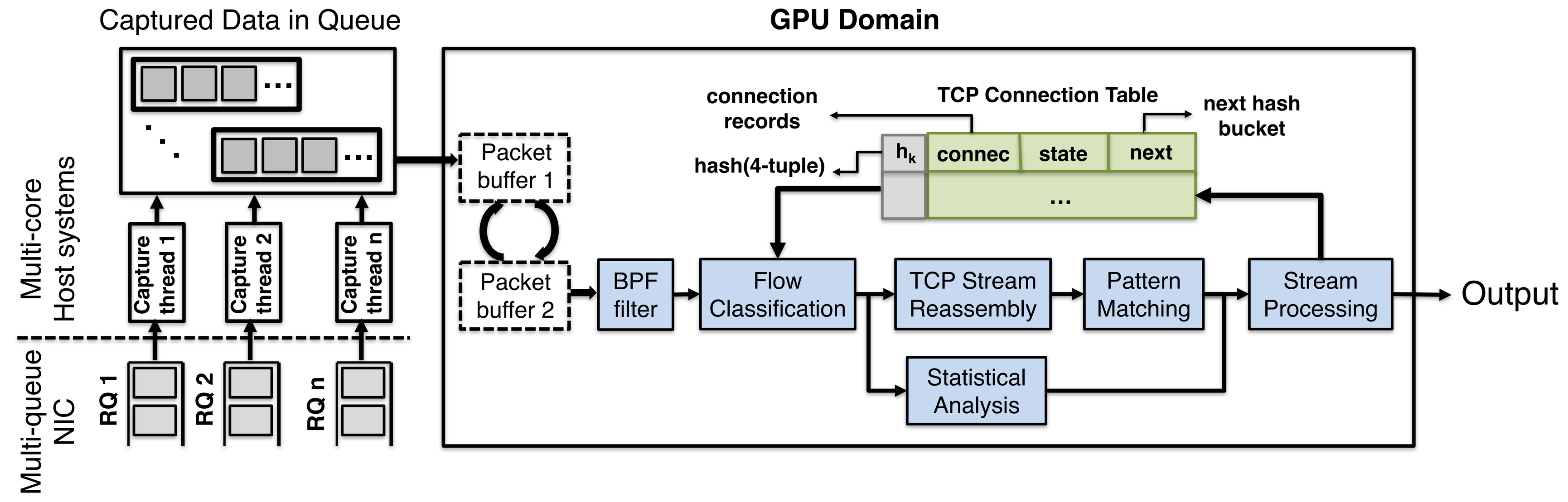
- For TCP traffic, payload of packets affiliated to the same TCP stream need to be assembled before matching against pre-defined patterns.



- Conventional packet normalizers have to buffer all packets following a missing packet, until they become in-sequence again, to prevent TCP fragmentation evasion attacks.



Framework



Key Functions

Flow Classification and Reordering:
- Inter-flow classification: sort streams according to their TCP 4-tuple
- Intra-flow reorder: sort same-stream affiliated packets by their sequence number

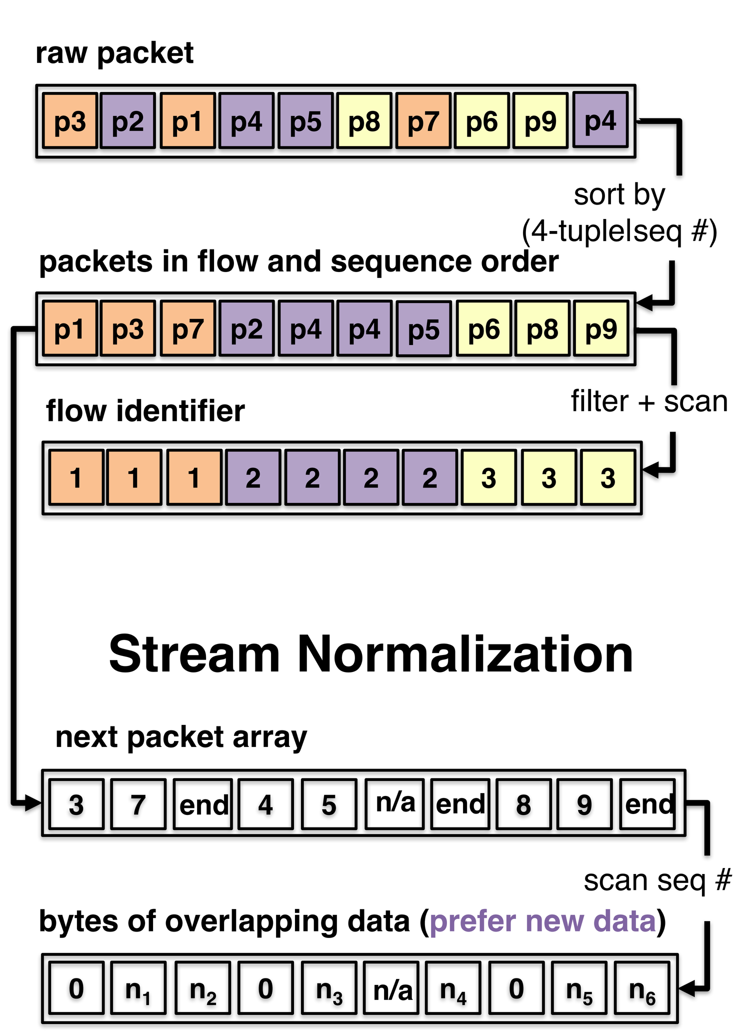
Packet Normalization: drop duplicate packets and merge the overlapping payload

Pattern Matching: match the payload against fixed strings

Stream Processing: keep the internal states between consecutive batches

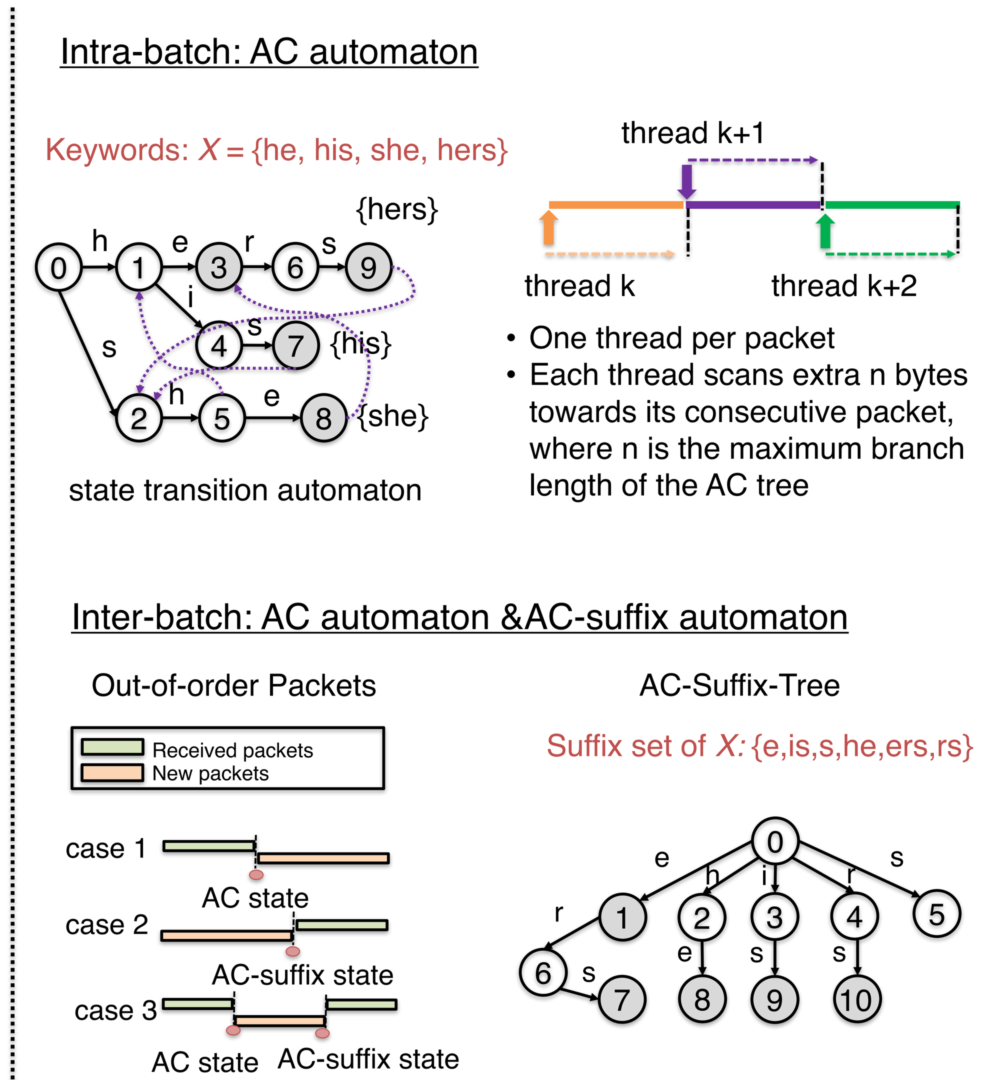
Key Mechanisms

Flow Classification



Parallel sorting for intra-batch flow classification and packet reordering.

Pattern Matching



AC and AC-suffix automaton to preserve the inter-batch pattern matching states.

Performance

Traffic source: real traffics mirrored from the Fermilab border router, with a mean packet length 1042-byte.

Base system: Intel Xeon CPU E5-2650 v3, NVIDIA GPU K40

Pattern for searching: 2120 string extracted from Snort rules

Raw Processing Throughput

	GPU	CPU
TCP Reassembly	552.28 Gbit/s	4.55 Gbit/s (Libnids)
Pattern Matching	62.53 Gbit/s	0.27 Gbit/s (Snort)

End-to-End Throughput

GPU (8-capture-threads)		CPU (single-threading)		CPU (8-threads)
w/ pkt transfer	w/o pkt transfer	Snort[1]	CPU-AC-suffix[2]	CPU-AC-suffix
31.95 Gbit/s	54.61 Gbit/s	0.25 Gbit/s	0.61 Gbit/s	3.58 Gbit/s

Comparison to Existing Tools

	Snort[1]	AC-suffix	GASPP[3]	Ours
Computing platform	CPU	CPU	GPU	GPU
Methods	Stream Reassembly	Split detection	Intra-batch stream reassembly	In-batch stream reassembly + inter-batch split detection
Detection over OOO packets	✔	✔	limited	✔
Resistance to fragmentation flood	N	Y	N	Y
Throughput	Low	Low	High	High

[1] Roesch, Martin. "Snort: Lightweight intrusion detection for networks." *Lisa*. Vol. 99. No. 1. 1999.
[2] Chen, Xinming, et al. ``AC-suffix-tree: Buffer free string matching on out-of-sequence packets." Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on. IEEE, 2011.
[3] Vasiliadis, Giorgos, et al. "Design and Implementation of a Stateful Network Packet Processing Framework for GPUs." *IEEE/ACM Transactions on Networking* (2016).

Conclusion

- Develop a highly efficient packet/flow analysis tool hat fully utilizes the parallelism of multicore systems, NICs, and GPUs.
- Present a novel GPU-centric TCP state management and stream reassembly framework.
- Perform on-the-fly DPA without requiring dropping or buffering the OOO packets by implementing an AC-suffix-tree method on GPU.

Selective Prior Work

[1] Wenji Wu, et al. "Deep Packet Analysis using GPUs". GTC 2017.
[2] Wenji Wu, Phil DeMar "Wirecap: a novel packet capture engine for commodity NICs in high-speed networks". Proceedings of the 2014 Conference on Internet Measurement Conference. ACM, 2014.