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 Gb/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
Flow Classification and Reordering:
• Inter-flow classification: sort streams according to their TCP 4-tuple
• Intra-flow reordering: sort same-stream affiliated packets by their sequence number
Packet Normalization: drop duplicate packets and merge the overlapping payload
Stream Processing: keep the internal states between consecutive batches

Key Functions
Flow Classification
- raw packet
- packets in flow and sequence order
- flow identifier
- state transition automation

Stream Normalization
- next packet array
- bytes of overlapping data (prefer new data)
- Parallel sorting for intra-flow classification and packet reordering

Intra-flow: AC automaton
- Keywords: X = {s, h1, s2, h2, h3}
- One thread per packet
- Each thread scans extra n bytes towards its consecutive packet, where n is the maximum branch length of the AC tree

Inter-flow: AC automaton & AC-suffix automaton
- Out-of-order Packets
- AC-Suffix-Tree

Key Mechanisms

Challenges

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 processing: 2120 string extracted from Snort rules

Raw Processing Throughput

End-to-End Throughput

Comparison to Existing Tools

Conclusion
• Develop a highly efficient packet/flow analysis tool that 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