Randomized session-memory purging in Internet routers

Gunwoo Nam, Pushkar Patankar, George Kesidis, and Chita Das

Research Goal

•**Problem**: How to protect routers maintaining per-flow state from excessive I/O operations in the presence of very large numbers of flows or SYN flooding?

•Any performance degradation of the router could adversely affect the entire network connecting to the Internet through it

•An attack response strategy to SYN floods may be more effectively and feasibly enacted by intermediate routers instead of those very close to the victim or attacking end-systems

•We can avoid excessive I/O upon mass deletion by using logical swapping of the flow's enable and touch bits or by using randomized replacement of flow entry

•As Intrusion Detection, QoS assurance, and billing motivate stateful inspection of TCP flows, modern network devices such as routers and IDSs have the capability to monitor tens of thousands of active TCP sessions

•TCP session termination may occur upon receipt of RST/FIN or time-out

•Trade-off in the selection of the time-out thresholds:

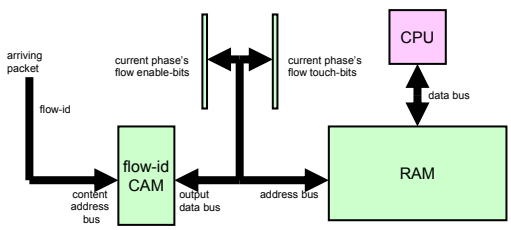
- false-positives if threshold is too small
- false-negative if threshold is too large

•Choice of time-out threshold may effect how many TCP sessions simultaneously expire

A Deterministic Approach

•Time-outs can be managed within each session's FSM using "touch" flags or at the session/flow level to remove or disable entire FSMs "stale" (not recently touched) flows from memory

•Phase-time T_ϕ *(30 sec usually)* to time-out flows
- After every T_ϕ time, all untouched flows are purged



•A flow touch/enable-bit is maintained in a separate CAM

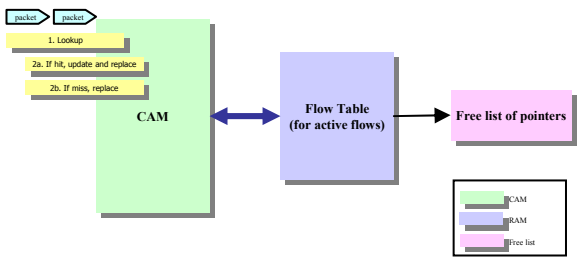
•No free list of pointers to the RAM

•Logical swapping of the flow *enable* and *touch* 1-bit CAMs upon conclusion of a phase time:

- All flows whose touch-bit = 0 have "timed-out"
- Vector of touch-bits in RAM → vector of enable-bits
- Reset the touch-bit to 0 for all flows whose touch-bit =1

•It is possible *all* expired flows (involving tens of thousands of flows) are to be deleted upon conclusion of the current phase, and such mass purging needs to be done effectively

A Randomized Framework



•In this system, an active flow may have its flow-id resident in the CAM multiple times

•A free list of pointers to the CAM is maintained

•Each FSM-RAM entry has a linked list of addresses in the flow-id CAM where the corresponding flow-id is currently stored (can thereby detect "heavy hitter" flows)

•Time-out parameters are *not* explicitly employed

•Upon arrival of a packet:
- If flow-id "hits" in the CAM, update state in RAM
- If flow-id "misses" in CAM (may be a SYN packet of a new flow or a packet of a flow whose RAM entry was over-written), select a CAM entry at random and replace it with new flow-id with a probability r

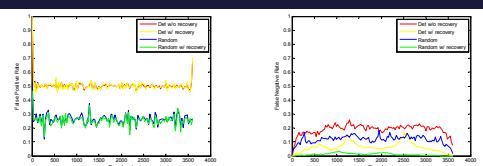
•Differential handling of SYN packets: By monitoring SYN/non-SYN ratio, the probability of acting on SYN packets can be adjusted

•Performance parameters

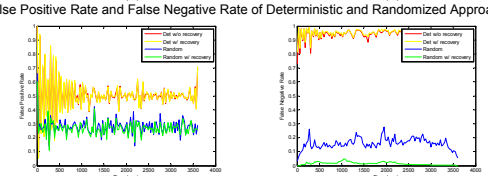
- False Positive Rate = $\frac{False\ Positive}{False\ Positive + True\ Negative}$
- False Negative Rate = $\frac{False\ Negative}{False\ Negative + True\ Positive}$

-True Positive: Active flows
-False Positive: Staled flows at next phase (T_ϕ)
-True Negative: Deleted flows + Stale flows in the current phase
-False Negative: Wrongly removed flows

Performance Evaluation



False Positive Rate and False Negative Rate of Deterministic and Randomized Approaches



False Positive Rate and False Negative Rate of Deterministic and Randomized Approaches under SYN flooding

•The experiment is done by real Internet traffics captured by Lawrence Berkeley National Laboratory

Conclusion

•Proposed a randomized framework for mass purging of deemed-stale (per-flow) TCP session states in Internet routers

•Our approaches can accommodate limited hardware and software resources in addition to very high data rates by avoiding excessive I/O to free-list memories

•Adaptation and recovery methods are being explored to enhance performance

Sponsored by NSF and Cisco Systems, Inc.