Online Detection of Anomalies in Distributed Streams

Sneha Aman Singh

Srikanta Tirthapura (faculty advisor)

Electrical & Computer Engg., Iowa State University
Example: Botnet Attack Detection

• Data from Intel Labs:
  – Endhosts Monitored for many weeks within a domain
  – End hosts that were comprised by botnets
    • Contacted the C&C center regularly (period unknown)
    • Very little traffic each contact
    • Ex: IRC.Zapchast-11 contacted its C&C center every 10 hours, 1.4 connections per contact

• Giroire et al. “Exploiting Temporal Persistence to Detect Covert Botnet Channels” (RAID 2009)
Which items are persistent?
What are Frequent request types?

Distributed Streams Model

Server 1 (Indiana)
Server 2 (Italy)
Server 3 (Ames)

Master Server (Ames)

Traffic

Low bandwidth communication
Persistent Items in a Stream

Persistence of \(a\) = 4 out of 4 slots = 100 %

Persistence of \(d\) = 2 out of 4 slots = 50 %

In botnet detection, look for destinations with high persistence

Periodic connections to an online advertisement indicates click fraud

Persistent Elements need not contribute to traffic volume: not “heavy hitters”
Research Task

• Design Techniques for Monitoring Over the Union of Distributed Streams
  – Persistent Elements
  – Entropy
  – Frequent Elements ("heavy hitters")
Task I: Tracking Persistent Elements

• In a large distributed stream of network traffic, track all items whose persistence exceeds a threshold

• Challenges:
  – Space: Cannot store entire stream in memory
  – Communication: Cannot forward all streams to a central server
Approach to Online Algorithm

• Design “sketches” of streams that are sent to a monitor, and are smaller than the stream

• Look for Space-efficient algorithm with relaxed guarantees:
  – If persistence > \( \tau \), then item is reported
  – If persistence < \( (\tau-\varepsilon) \), then item not reported

• Balance the false positive/negative rate versus space/communication cost
Our Prior Work: Centralized Scenario

• Lahiri, Chandrashekar, Tirthapura, “Space-efficient Tracking of Persistent Items in a Massive Data Stream”, Proc. ACM Conference on Distributed Event-Based Systems, 2011
Evaluation (Lahiri et al. 2011)

- Data: Trace of 885 million packets (source: CAIDA)
- 30-sec slots, 350 slots total
- Query windows: [1,100], [26,125], ..., [251,350]
- [1,100] window, ~570k distinct IPs, but ~500k of them occur in < 10 slots
  - Storing a counter for every distinct item wasteful
Evaluation (Lahiri et al. 2011)

- False Negative Rates < 5%, mostly
- False Positive Rate < 3%
- 85% Space saving compared to naïve
  ✓ 445 MB instead of 3 GB
Task II: Monitor Entropy of Network Traffic

• Entropy a measure of “randomness” in traffic

• Prior data (Lakhina et al.) has shown that large changes in entropy are indicative of network anomalies

• Task:
  – Monitor entropy of union of distributed streams
  – Detect Large Changes in entropy
Summary

• Distributed Streams Monitoring
  – Patterns in the union of many physically distributed streams

• Monitoring Anomalies in Distributed Streams
  – Large Changes in Entropy
  – Large Changes in Set of Heavy Hitters