STONE: A Stream-based DDoS Defense Framework

Vincenzo Gulisano
Chalmers University of Technology

Chalmers University of technology

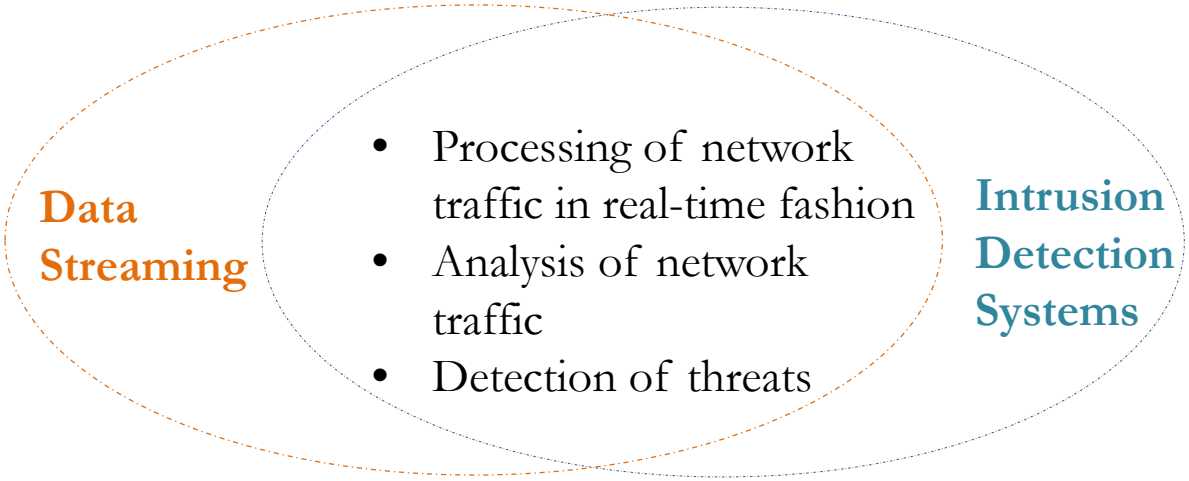
Technical University of Madrid

Agenda

- Introduction
- STONE architecture
- Evaluation
- Conclusions

STONE - Introduction

- DDoS Defense Framework
  - Monitor the traffic of an entity
    - Detect packet-flooding threats
  - When mitigating:
    - Maximize the percentage of legitimate traffic forwarded to the entity

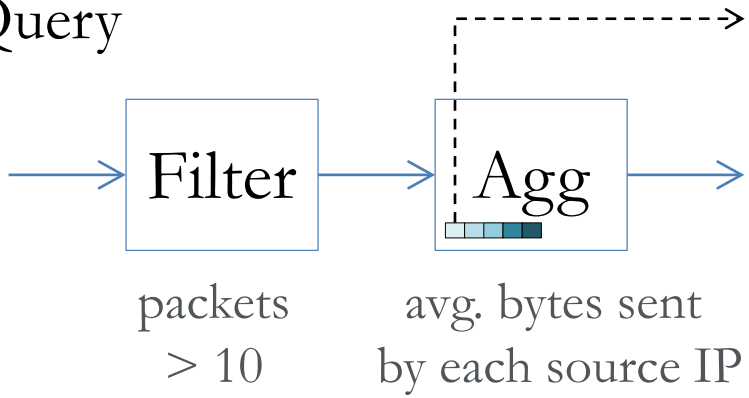


Data Streaming - System Model

- Data Stream
  - unbounded sequence of tuples sharing the same schema

$$<IP_A, IP_B, packets, bytes, T_{start}, T_{end}>$$

- Operator
  - Stateless: 1 input tuple → 1 output tuple
  - Stateful: 1+ input tuple(s) → 1 output tuple
- Continuous Query



Stateful operators perform their computations on **Sliding Windows**

- **Time-based**
- **Tuple-based**

Intrusion Detection Systems

- State of the Art approaches:
  - Misuse based:
    - Check each packet and decide whether to forward it.
    - New threats need new signatures
  - Anomaly based:
    - Spot deviations between current and reference traffic behavior
    - More challenging due to complex analysis for profiling

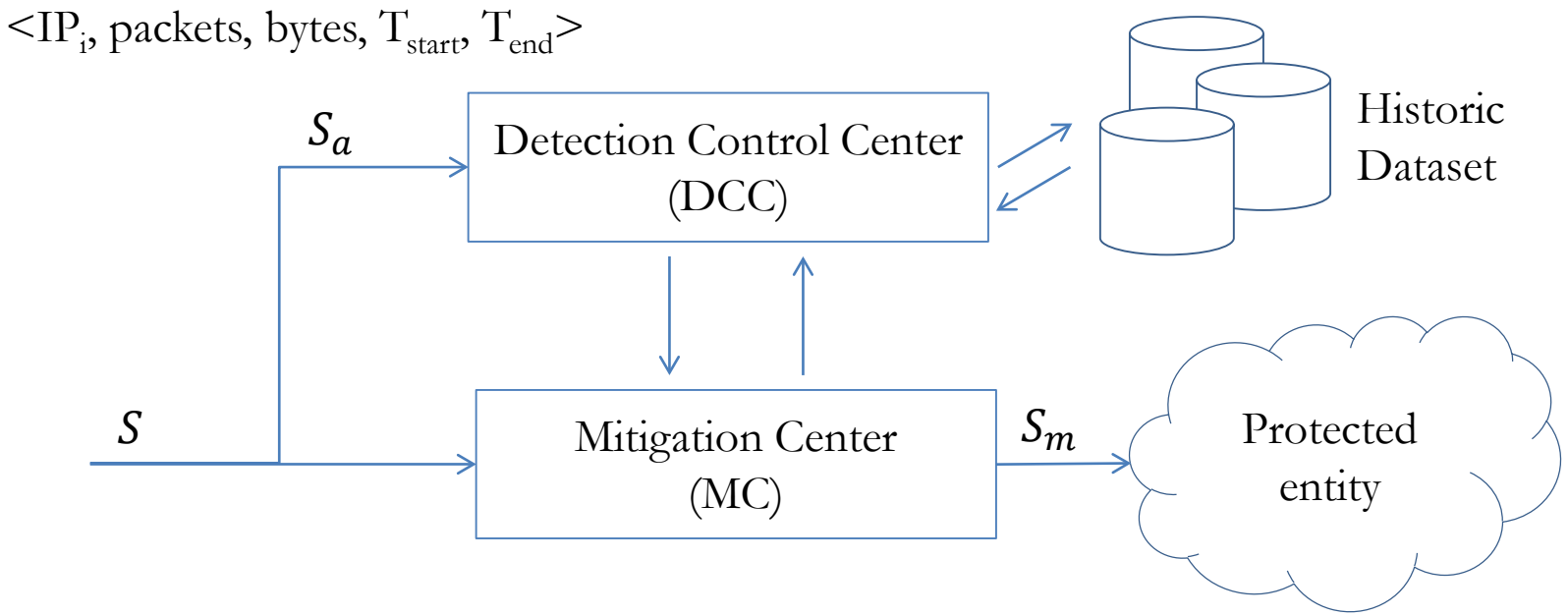
Agenda

- Introduction
- STONE architecture
- Evaluation
- Conclusions

System Model

- Network model:
  - Protected entities
  - Legitimate hosts
  - STONE machines
  - Bots
- Adversary model
  - Packet-flooding DDoS Attacks
  - No knowledge about traffic characteristics (e.g., distribution of source addresses)
  - Cannot modify nor pollute reference information

STONE Architecture



Detection Control Center

- If profiles are built on a per-source IP basis:
  - Possibly impractical due to huge amount of IPs
  - Predictability of individual IPs might be unreliable

- **Prefix level aggregation** of flow tuples into **source clusters**

Detection Control Center

Prefix level aggregation, IPs to source clusters:

<199.10.2.x_1, packets, bytes, T_{start}, T_{end}>

< 199.10.2.x_2, packets, bytes, T_{start}, T_{end}>

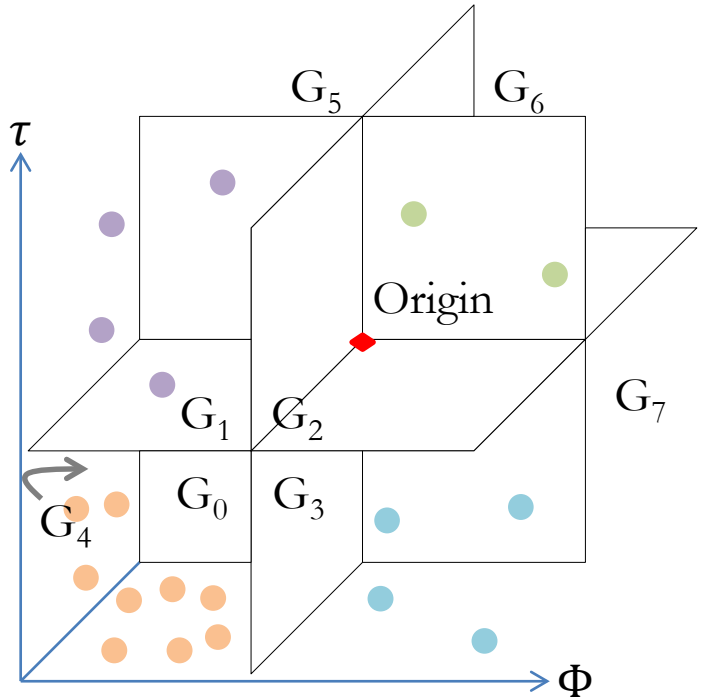
< 199.10.2.x_3, packets, bytes, T_{start}, T_{end}>

< 199.10.2.x_n, packets, bytes, T_{start}, T_{end}>

$srcCL_i$
- Φ_i average packets per flow
- ω_i average bytes per flow
- τ_i average duration per flow

(based on time-based sliding windows)

Detection Control Center

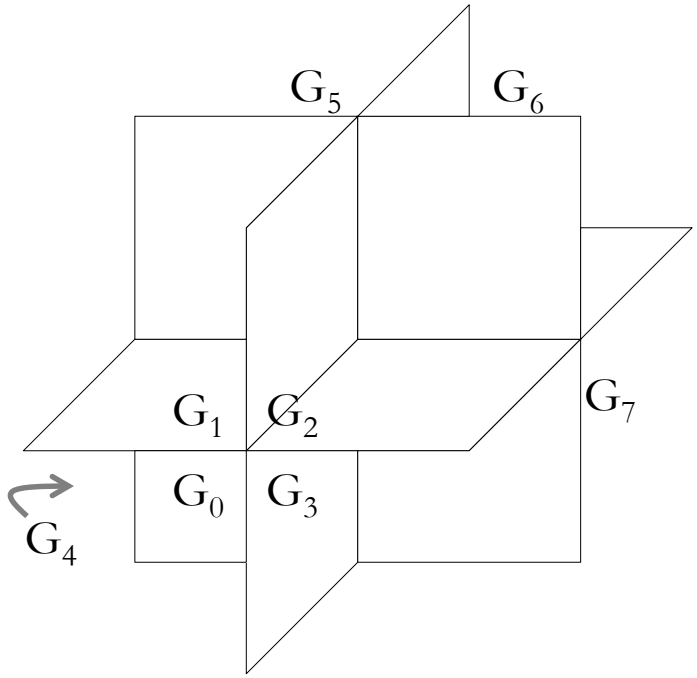


Φ: avg. packets / ω: avg. bytes / τ: avg. duration per flow

- Origin is chosen as 0.95-quantile
- Distribution of source clusters to groups is stable
- Maintain:
  - Current ratio $\hat{r}_i, i = 0 \ldots 7$
  - Reference ratio $r_i, i = 0 \ldots 7$
- Detection if
  - $max_i |r_i - \hat{r}_i| \geq tol$

Mitigation Center

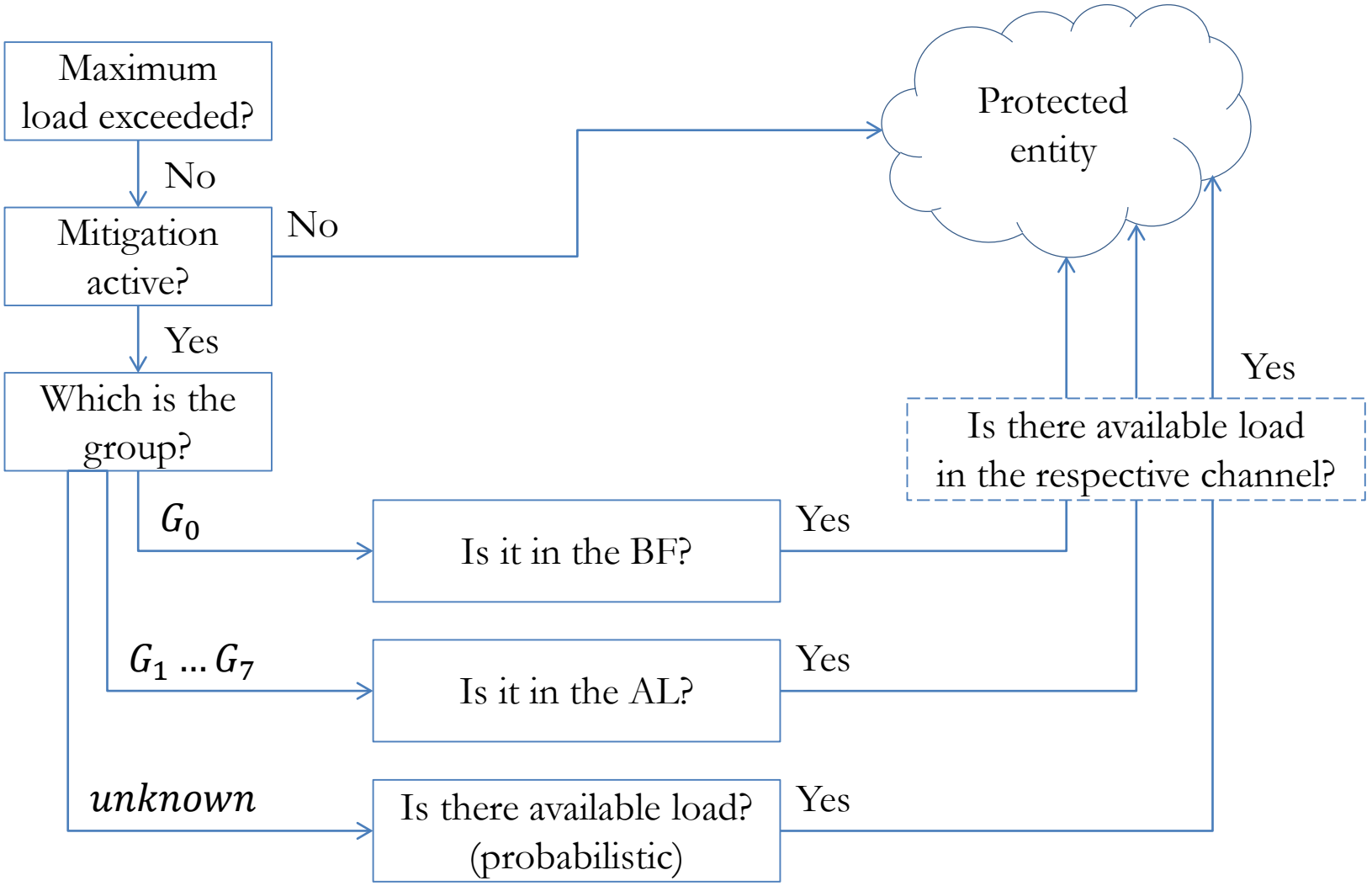
- Being L the maximum load of a protected entity
- Filtering is applied if traffic $\geq \alpha L$



- $G_1 \ldots G_7$: source clusters having at least 1 feature $\geq$ 0.95-quantile
  → **Acquaintance List (AL)**
  <$srcCL_i$, forward prob.>

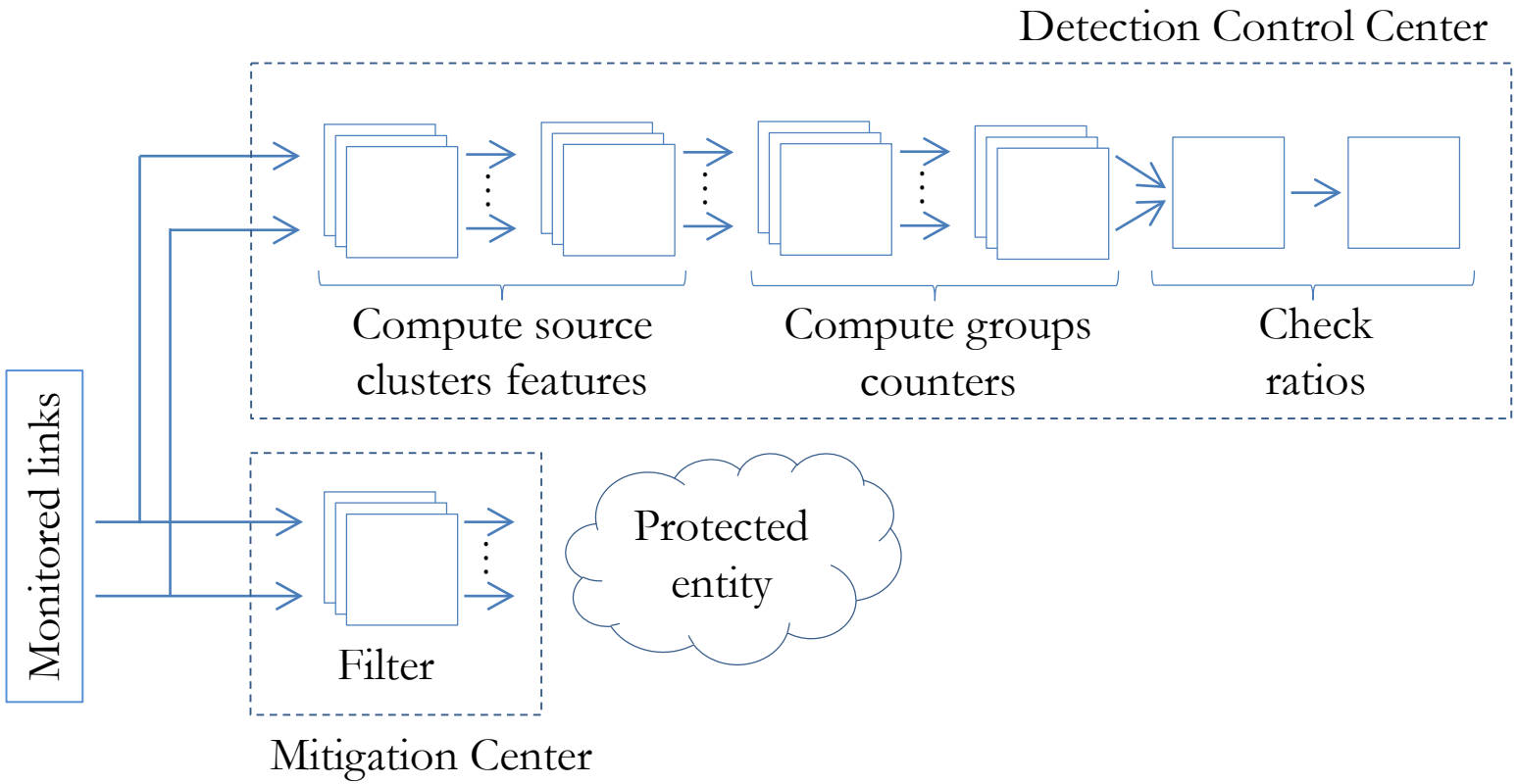
- G_0: source clusters having all features < 0.95-quantile
  → **Bloom Filter (BF)**
  $srcCL_i$ communicating during 5 minutes preceding the attack

Mitigation Center



Implementation of STONE

- Continuous query on top of StreamCloud



Agenda

- Introduction
- STONE architecture
- Evaluation
- Conclusions

Evaluation setup

- Legitimate traffic
  - Anonymized data traces from 10Gbits/s backbone link of OptoSUNET
  - Excerpts of traffic happening on Thursdays, 11:00 to 12:00 during 9 weeks in 2010

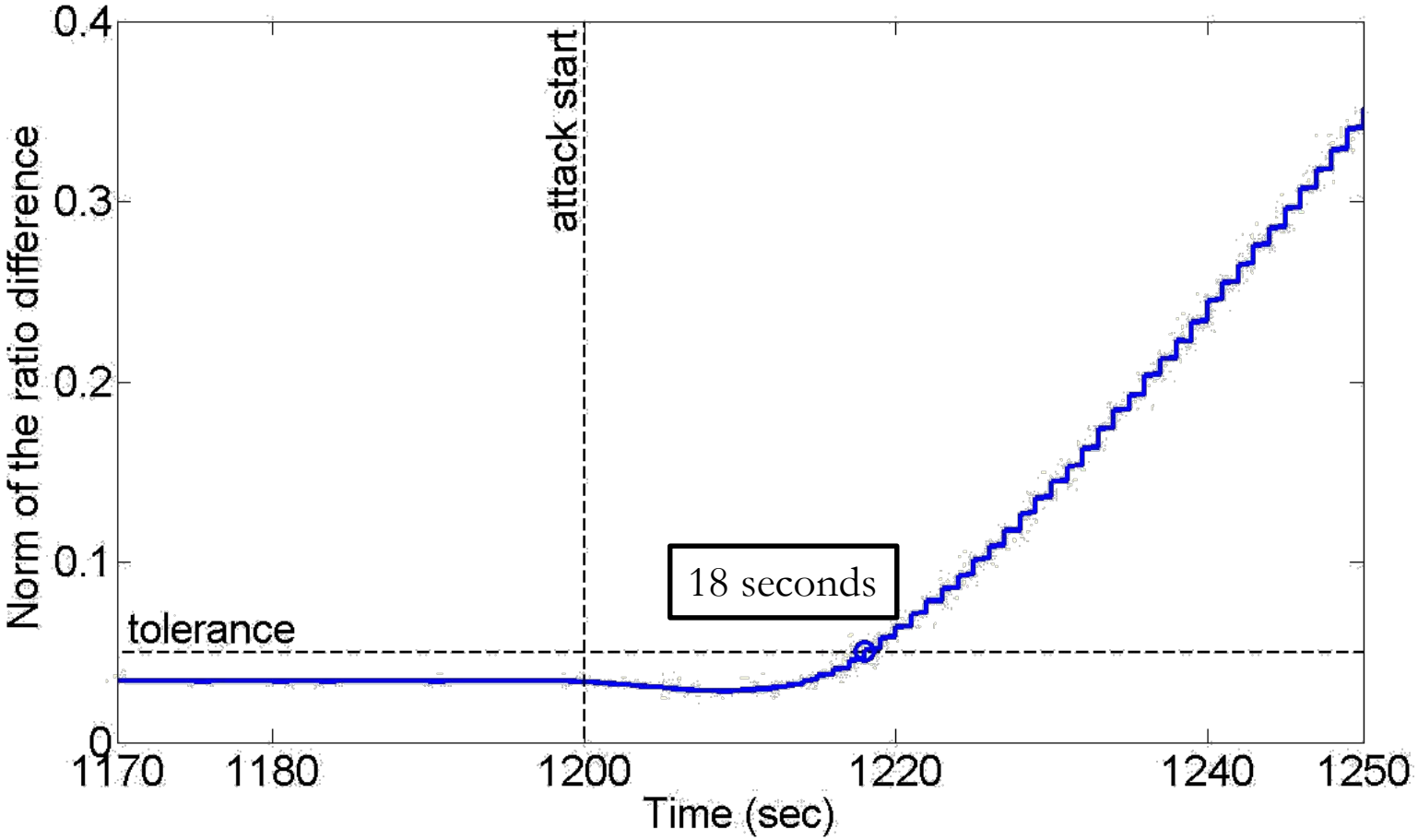
- Illegitimate traffic
  - Anonymized data traces from a DDoS attack (CAIDA)

Evaluation metrics

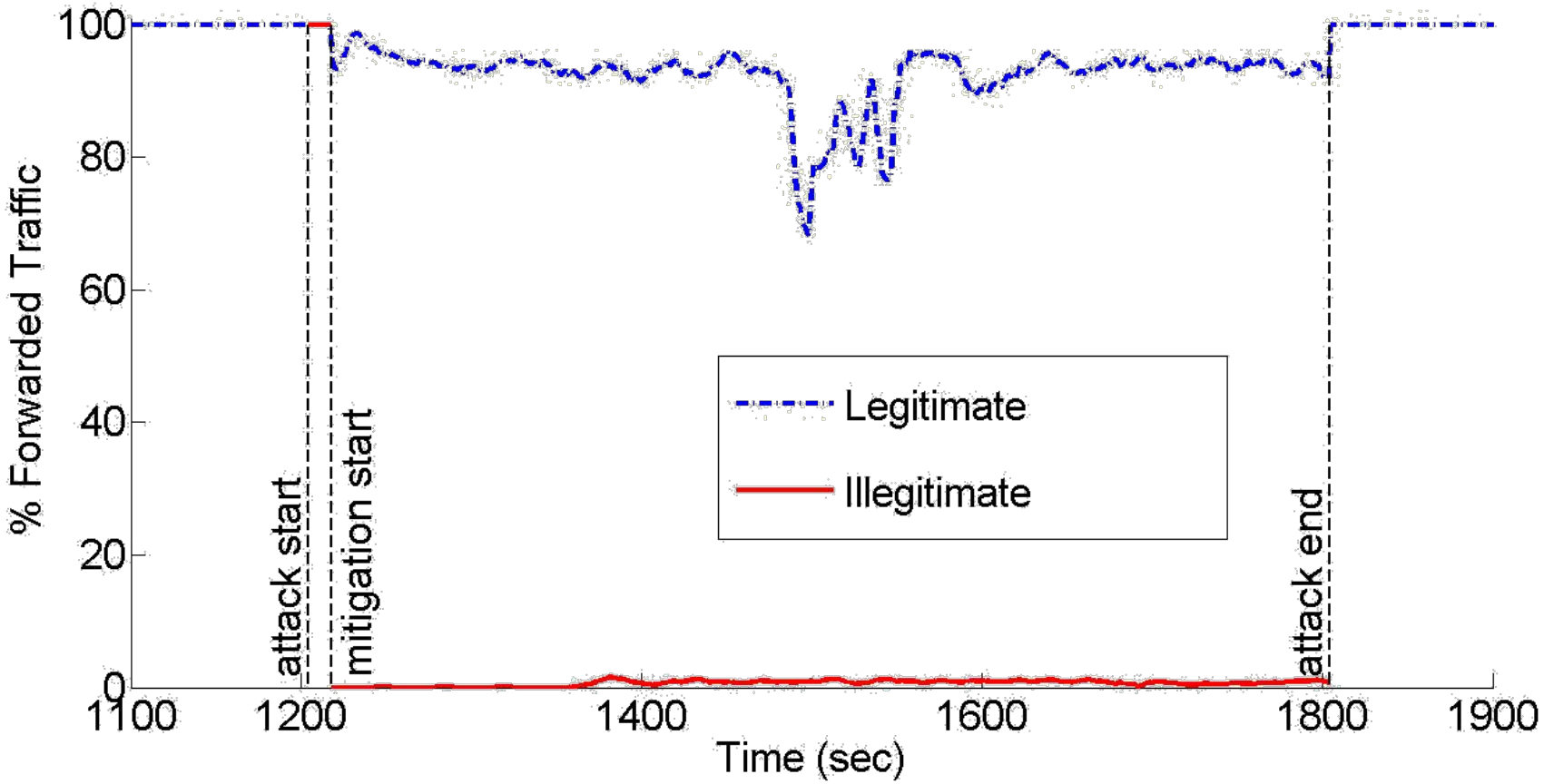
- While processing legitimate traffic we inject attack packets and evaluate:

- Detection time
  - Time elapsed between the attack start and the detection
- Mitigation precision
  - Degradation of legitimate user traffic
- Traffic volume shaping
  - Amount of traffic discarded during the attack

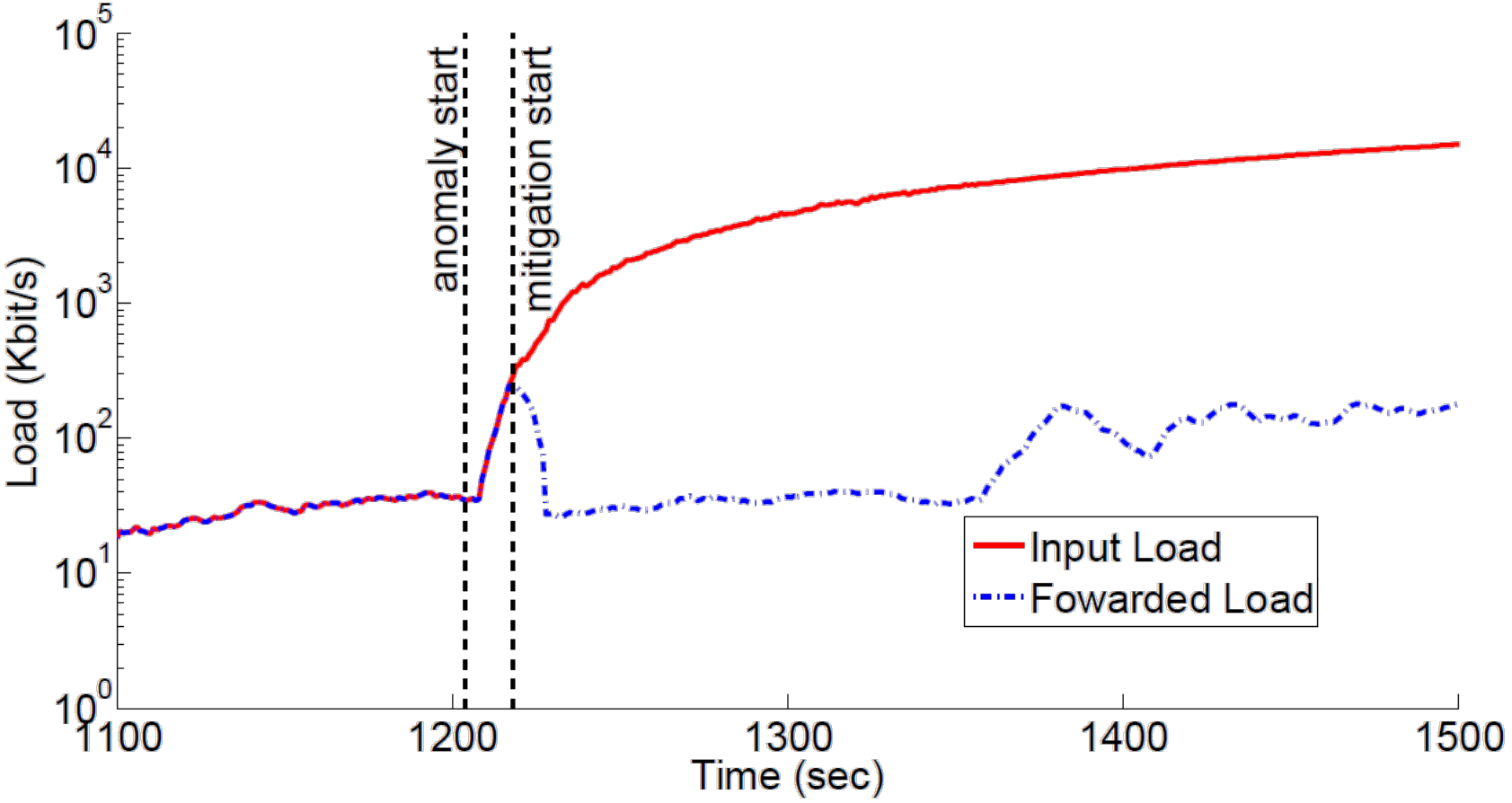
Detection Time



Mitigation precision



Traffic volume shaping



Agenda

- Introduction
- STONE architecture
- Evaluation
- Conclusions

Conclusions

- STONE: A Stream-based DDoS Defense Framework
- Anomaly-based defense that provides both detection and mitigation
- Traffic analysis based on the data streaming paradigm
- Evaluation based on real prototype (StreamCloud) and conducted using real traffic traces