IEEE/ACM International Symposium on Quality of Service (IWQoS) 2020

Application-Layer DDoS Defense with Reinforcement Learning

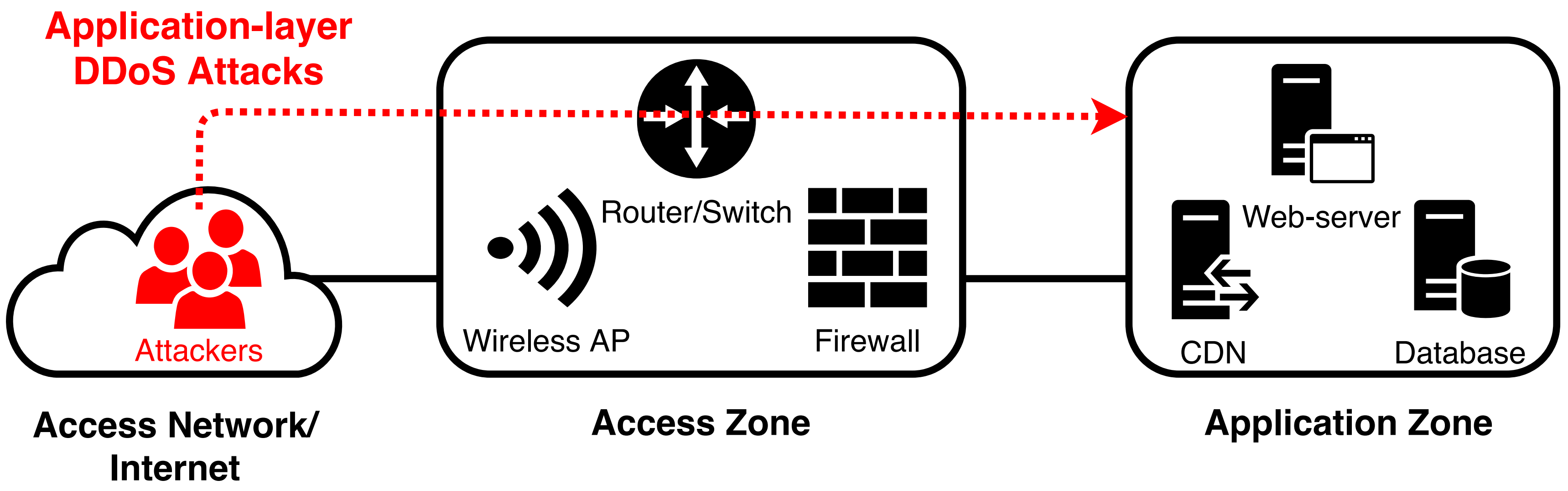
June 17, 2020.

Yebo Feng, Jun Li, Thanh Nguyen
Computer and Information Science Department
University of Oregon

Application-layer (L7) DDoS Attacks

L7 DDoS represents a type of malicious behaviors intended to target the application layer in the network model,

- where Internet application actions such as HTTP GET and SMTP LOGIN occur.
- It utilizes application messages to overwhelm certain components of the victim server.



Mechanisms L7 DDoS Attacks

- Request Flooding Attack.
  - Overwhelm the victim system by sending a large amount of application-layer requests at a high rate from different IP addresses.
  - HTTP flood, SMTP flood, SIP call-control flood, etc.
- Leveraged Attack.
  - Leverages the flaws of the victim system to amplify the threat.
  - Low and slow attack, AIP-based attack, etc.
- Lethal Attack.
  - The attacker first scans the victim system to pinpoint the performance bottlenecks (e.g., I/O, memory space, or database server).
  - Sends a large amount of related requests to overwhelm the bottlenecks.

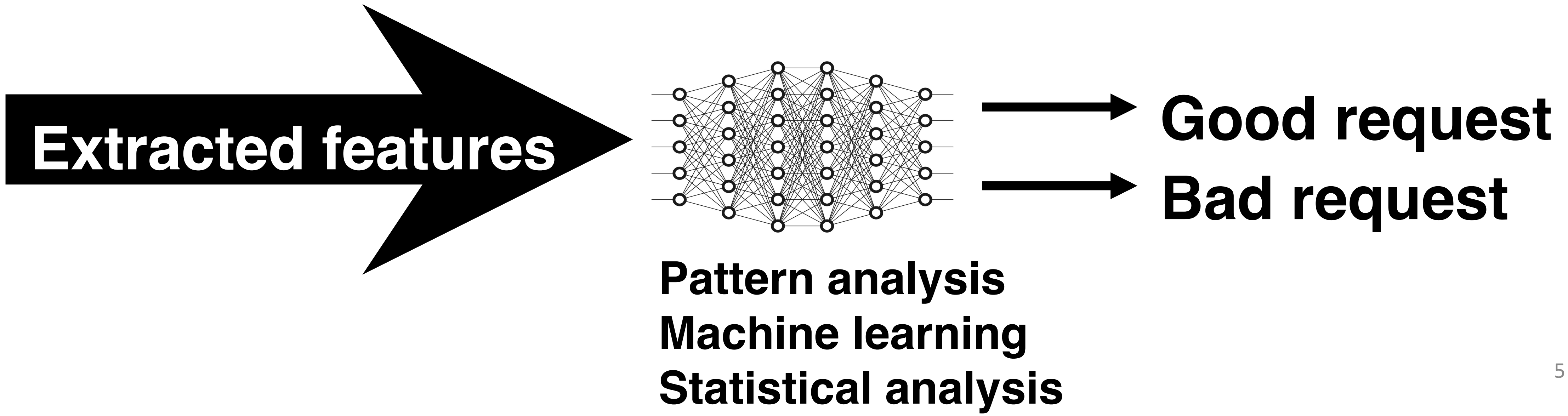
Impacts & Characteristics

- Application-Layer DDoS is on the rise.
  - It occupied about 38% of the total DDoS attacks during 2018[1].
- L7 DDoS attack poses threats to:
  - CPU load, Disk I/O, and memory space of the victim servers;
  - accessibilities of web application service.
- It is hard to detect and mitigate.
  - Traffic patterns are legitimate in network and transport layers.
  - Difficult to choose suitable mitigation methods.

[1]: https://securelist.com/

Design Philosophy of Existing Methods

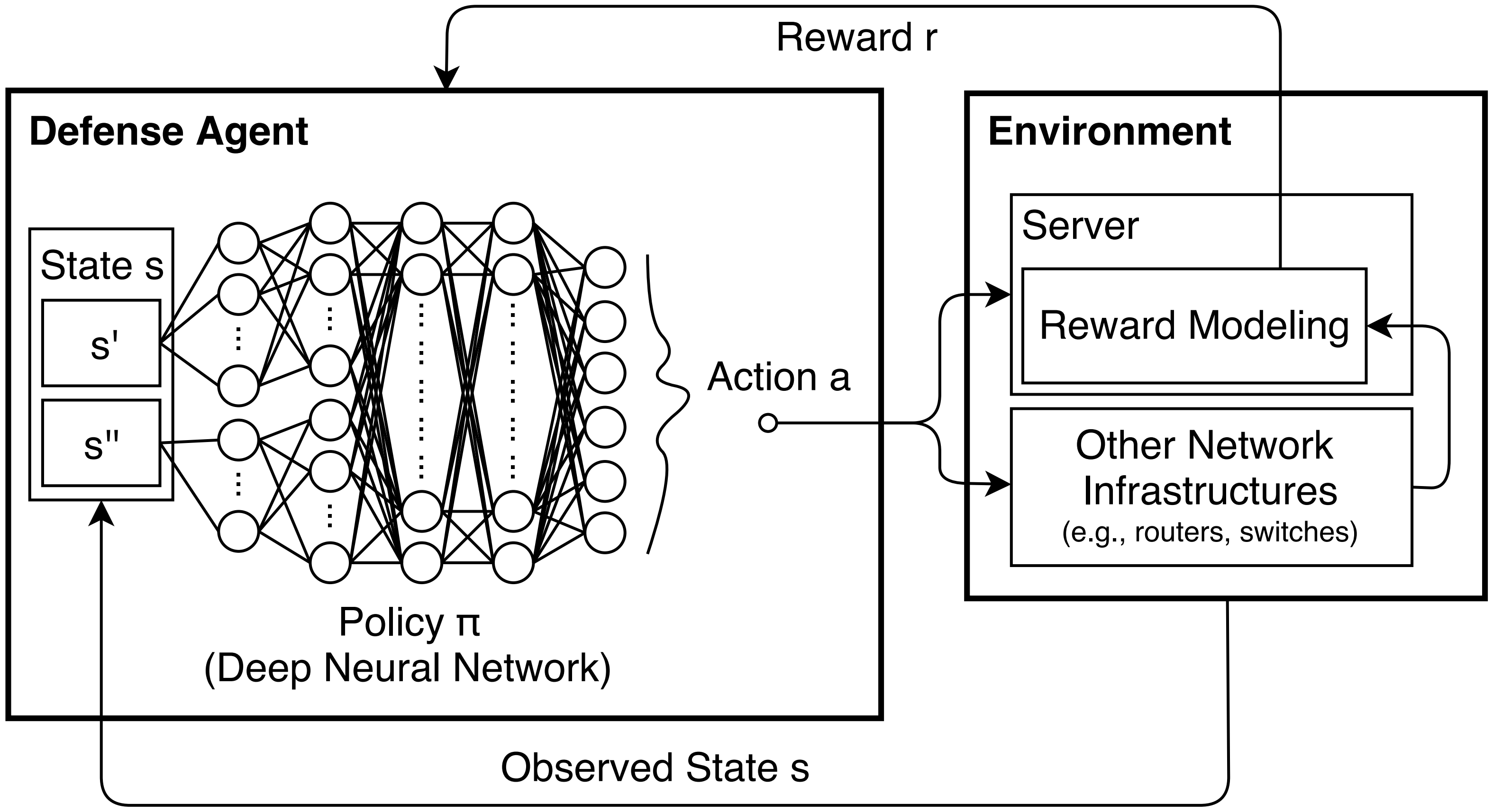
- Traditional defense model:
  - Information extraction → detection & classification → labels → mitigation
- Problems:
  - Application messages with malicious and legitimate intentions sometimes are identical, it is hard to perform precious classification.
  - Different attacks have different suitable mitigation methods.



Goals of Proposed Design

- Skip the detection & classification steps, directly generate the mitigation actions.
- Mitigate as many threatening requests as possible during severe attacks.
  - This might sacrifice a little bit false positive rate.
- Minimize collateral damage during server's routine operation.
  - This might sacrifice a little bit true positive rate.
- Ensure the efficiency during the defense process.

Reinforcement-learning-based L7 DDoS Defense Solution



State

Each state is represented by a state vector s, which has 12 dimensions. Each dimension of the state s is a value that represents a feature.

s', message-related states:

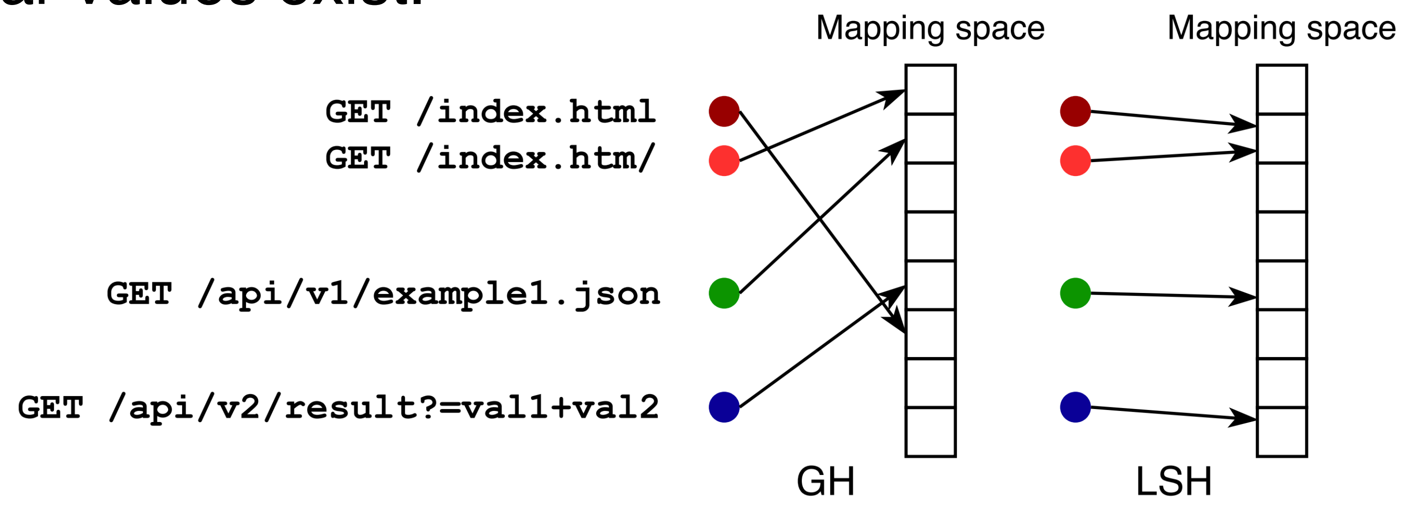
- Traffic size from the IP block
- Average behavior interval
- Interval deviation
- Request size
- Number of requests within 5 seconds
- Number of similar requests within 5 seconds
- Request consumption
- Ratio of incoming traffic size to outgoing traffic size

s'', server-related states:

- CPU utilization
- Memory utilization
- Link utilization
- Expected link utilization

Number of similar requests

1. Collect all possible typical request payload, normalize them as strings.
2. Use the string set to build a locality-sensitive hashing (LSH) query function.
3. For new request, go through the same normalization process, get the hashing value.
4. Count how many similar values exist.



Action a

Single-targeted

Multiple-targeted

- Scheduling actions:
  - a_i: let the server process the application request.
  - a_{ii}: let the server postpone the application request.
- Mitigation actions:
  - a_{iii} : drop application request.
  - a_{iv} : drop all the application requests that have contents similar to the current request.
  - a_v : blocking all the traffic from the IP address of the current application request.
  - a_{vi} : blocking all the traffic from the IP block of the current application request.

Reward Function When γ < α

When γ < α, we set the reward function for single-targeted actions as follow:

$$R_1(a(m)) = \begin{cases} -2 & for\ false\ positive\ sample \\ 1 & for\ true\ positive\ sample \\ -1 & for\ false\ negative\ sample \\ 0 & for\ true\ negative\ sample \end{cases}$$

We design the reward function for multiple-targeted actions as:

$$\begin{aligned} R_2(a(M)) &= \eta \sum_{i=1}^{n} R_1(a(m_i)) \\ &= \eta(-2|fp| + |tp| - |fn|) \end{aligned}$$

a(m): Conduct action a on message m.

α : The policy transition threshold value.

γ : The current system load.

η: The reward multiples for adjustments of the reward function, which is a value larger than 1.

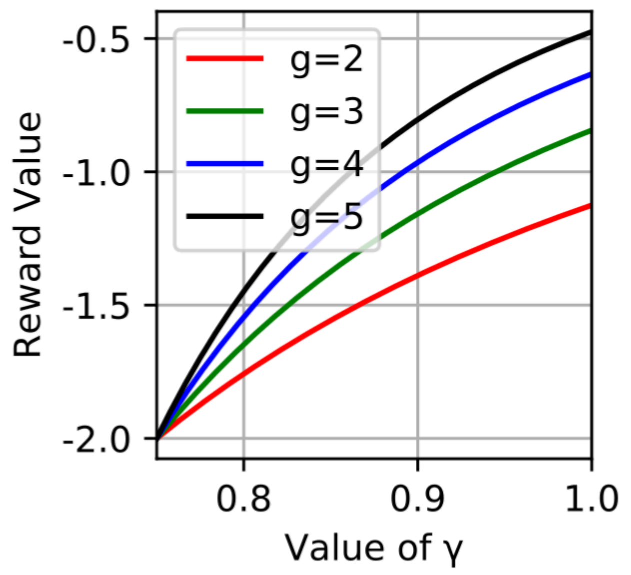
Reward Function for single-targeted action (When γ ≥ α)

When γ ≥ α, the victim system is heavily loaded, we set the reward function for single-targeted actions as follow:

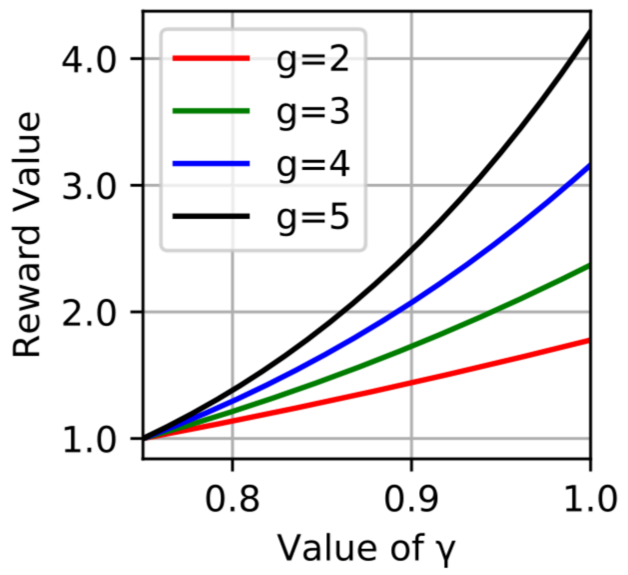
$$R_3(a(m)) = \begin{cases} -\frac{2}{(\frac{\gamma}{\alpha})^g} & for\ false\ positive\ sample \\ (\frac{\gamma}{\alpha})^g & for\ true\ positive\ sample \\ -(\frac{\gamma}{\alpha})^g & for\ false\ negative\ sample \\ 0 & for\ true\ negative\ sample. \end{cases}$$

Where g is the hazard index, an input parameter that determines how eager the victim wants the attack to be mitigated.

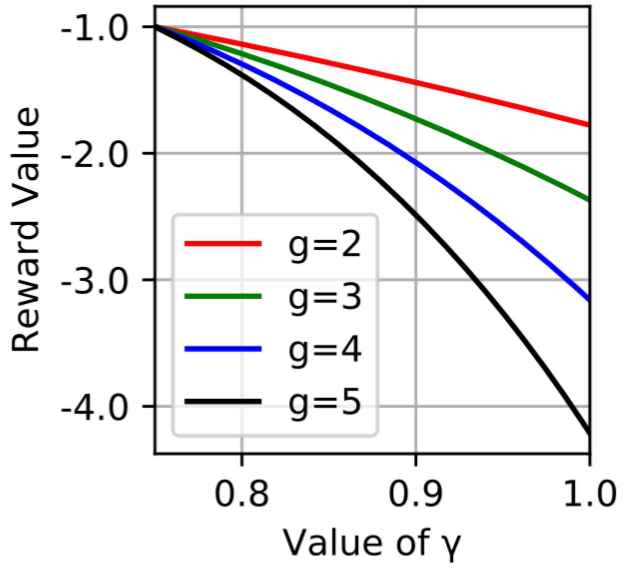
Reward Function for single-targeted action (When γ ≥ α)



(a) For false positive samples



(b) For true positive samples



(c) For false negative samples

Reward Function for multiple-targeted actions (When γ ≥ α)

In the scenario that the agent is making multiple-targeted actions, and γ ≥ α, we set the reward function as follow:

$$\begin{aligned} R_4(a(M)) &= \eta \sum_{i=1}^{n} R_3(a(m_i)) \\ &= \eta(-\frac{2}{(\frac{\gamma}{\alpha})^g}|fp| + (\frac{\gamma}{\alpha})^g|tp| - (\frac{\gamma}{\alpha})^g|fn|) \end{aligned}$$

α : The policy transition threshold value.

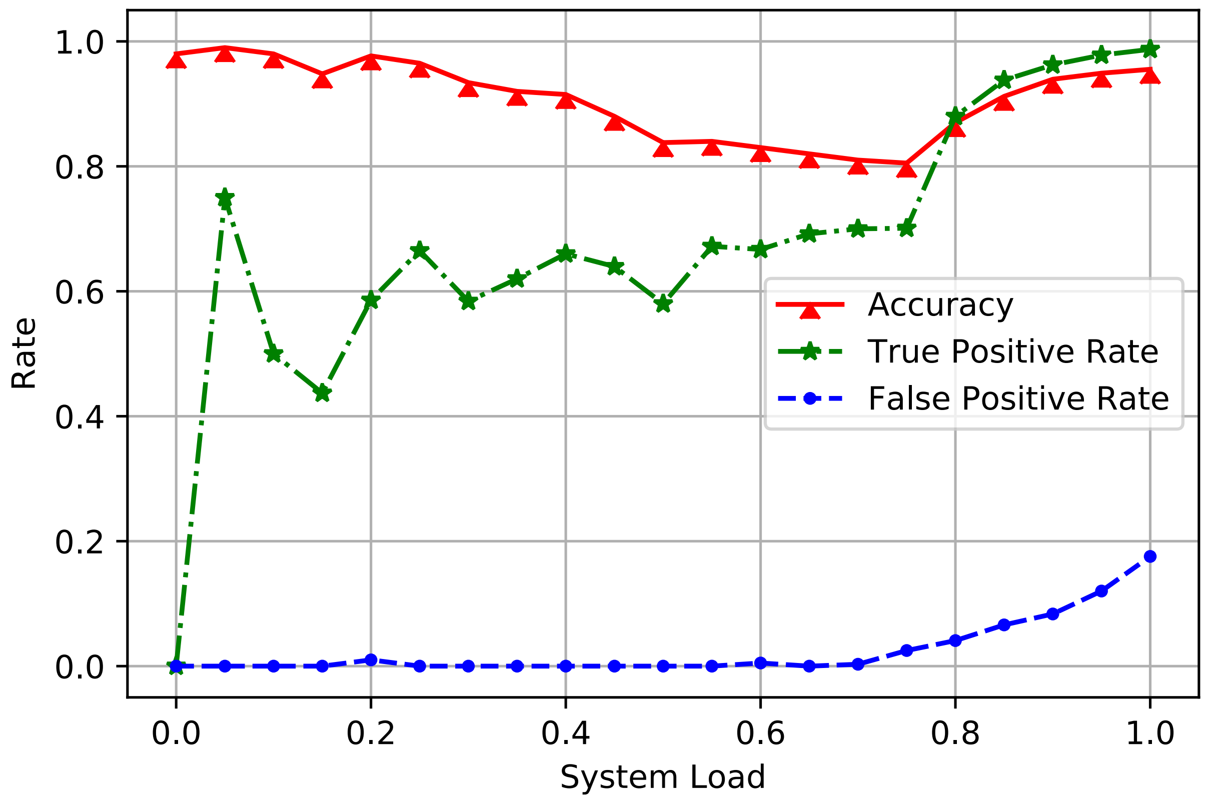
γ : The current system load.

η: The reward multiples for adjustments of the reward function, which is a value larger than 1.

Ability of Mitigating Attacks

Mitigation accuracies during different system load (α = 0.75)

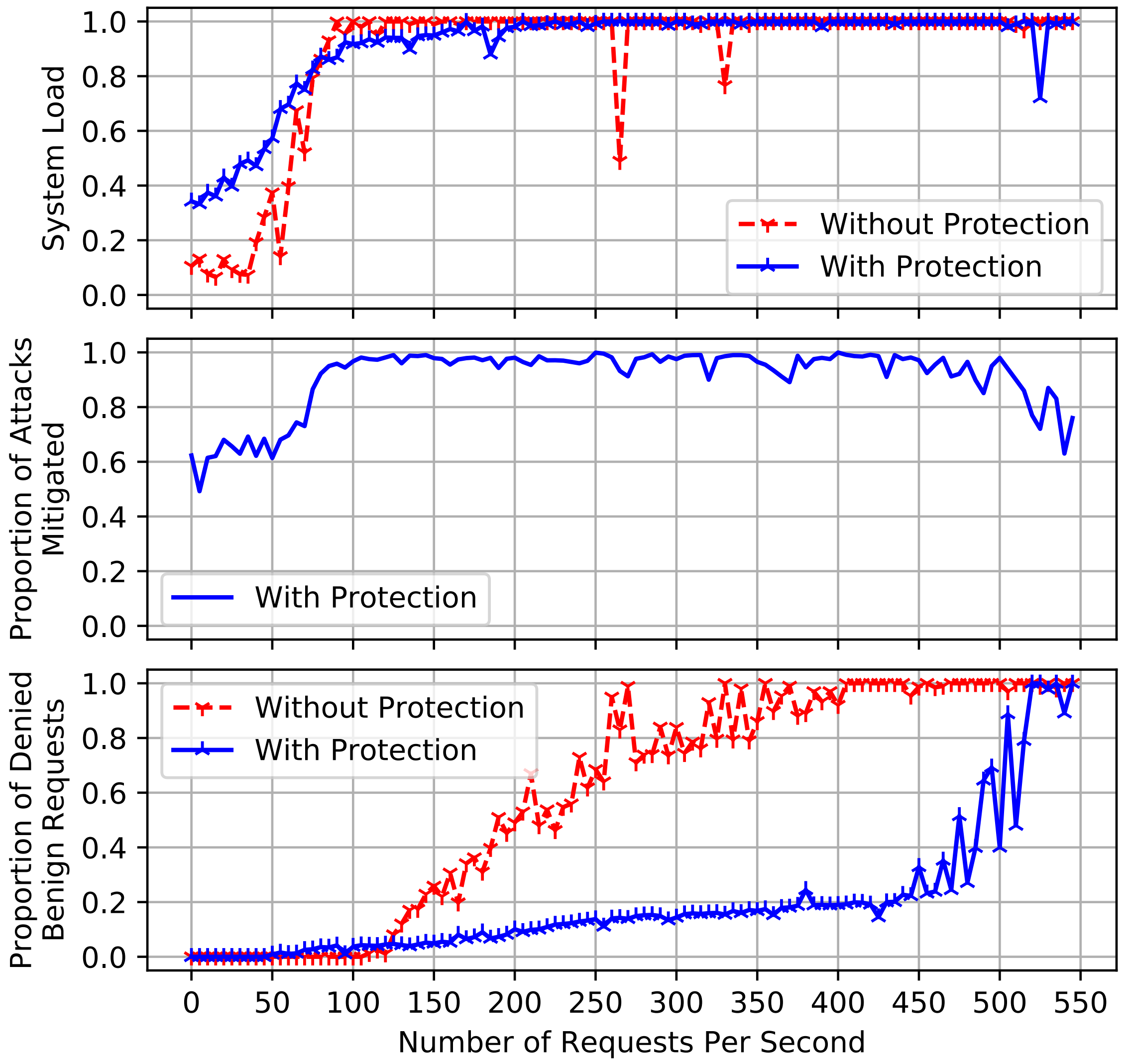
- Attack traffic: legitimate traffic = 4:1
- The accuracy transition comes when the system workload is at 0.75.
- When the system load stabilizes at 100%, the accuracy, true positive rate, and false positive rate are 0.9553, 0.9873, and 0.1756 respectively.



Mitigation Efficacy

If we assume that a server is considered to be proper functioning when the deny rate of legitimate requests is lower than 20%, the capability of the server without protection is 140 requests per second.

While for the server with protection, the deny rate of legitimate requests goes higher than 20% after the number of requests per second hitting 440, which is 3.15 times the capability of the unprotected server.

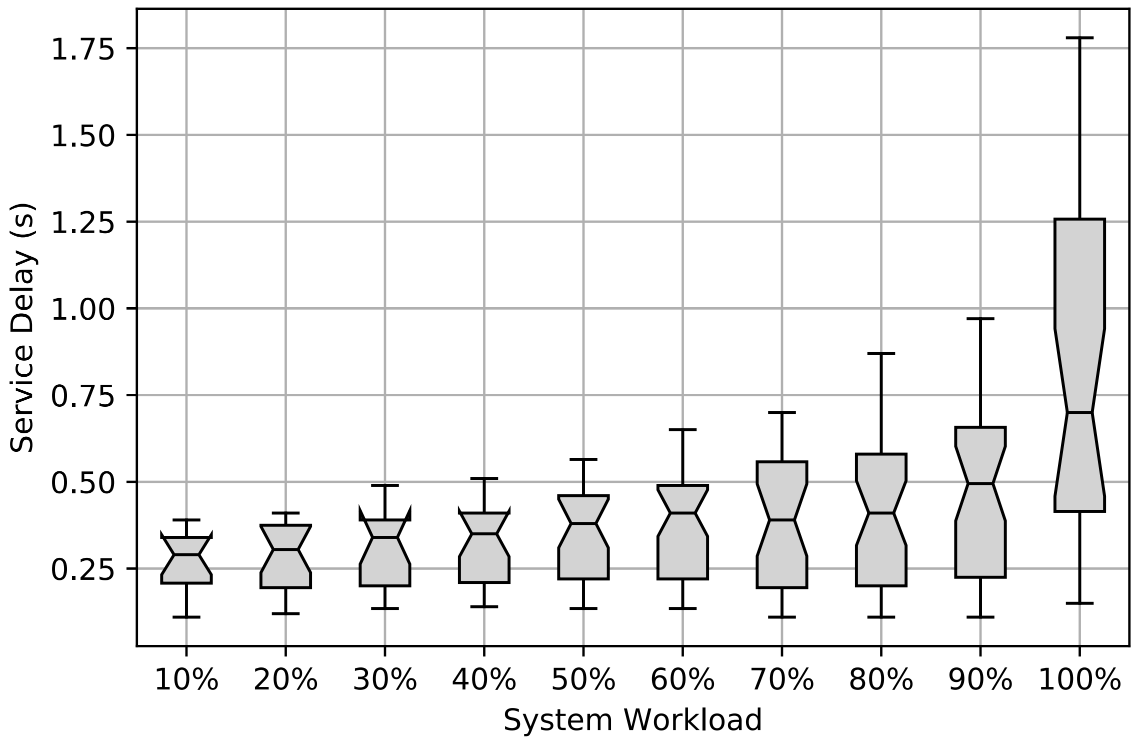


Service Delay

The delay is defined as the time duration from sending out a request to receiving the whole reply.

The delay without any defense approaches implemented is around 0.25 seconds.

The average delay time for the server with protection remains under 0.5 seconds when the system workload is less or equal to 90%.



Conclusion

- This project proposes the first reinforcement-learning-based L7 DDoS defense approach.
- It continuously monitors and analyzes a variety of metrics related to the server's load, the dynamic behaviors of clients, and the network load of the victim, to detect and mitigate L7 DDoS attacks.
- This approach employs a new multi-objective reward function that minimizes false positive rate to avoid collateral damage when the victim system load is low and maximizes the true positive rate to prevent the server from collapse when the victim system load is high enough.
- Performs the mitigation with acceptable delay; detects 98.73% of the L7 DDoS traffic flows at the peak system load, mitigating most of them.

Thanks for listening!

Q&A

Acknowledgment

The authors would like to thank **Derek Strobel** from the University of Oregon for his proofreading and constructive suggestions on this work.