SENSS Against Volumetric DDoS Attacks

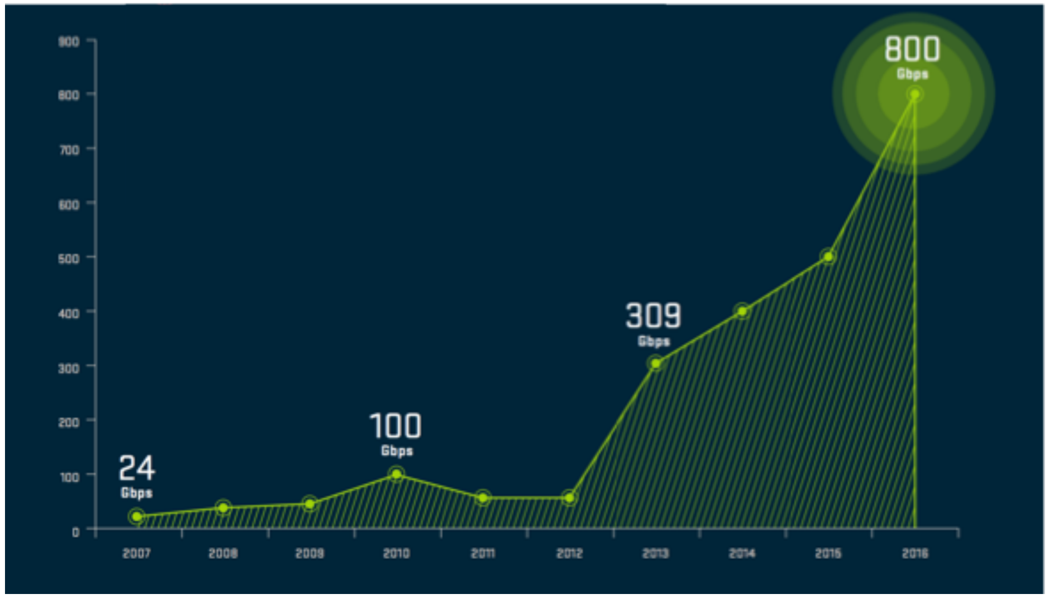
Sivaram Ramanathan[1], Jelena Mirkovic[1], Minlan Yu[2] and Ying Zhang[3]

[1]University of Southern California/Information Sciences Institute

[2]Harvard University

[3]Facebook

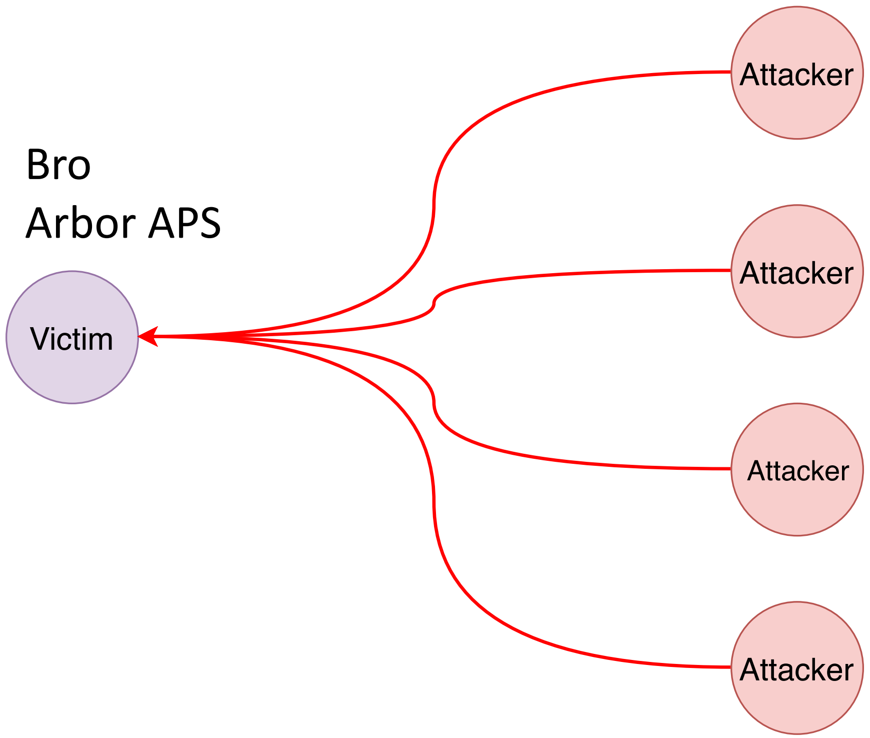
DDoS attacks



- Volumetric DDoS can overwhelm networks
- Such attacks are hard to mitigate by victim
  - Volume is too high for victim to handle – need help of upstream ISPs
  - Legit traffic mixed with attack traffic – need help to place imperfect filters near attack sources to minimize collateral damage
- Need collaborative, distributed response
- But today's internet lacks the infrastructure for victim to ask peers or remote networks for help

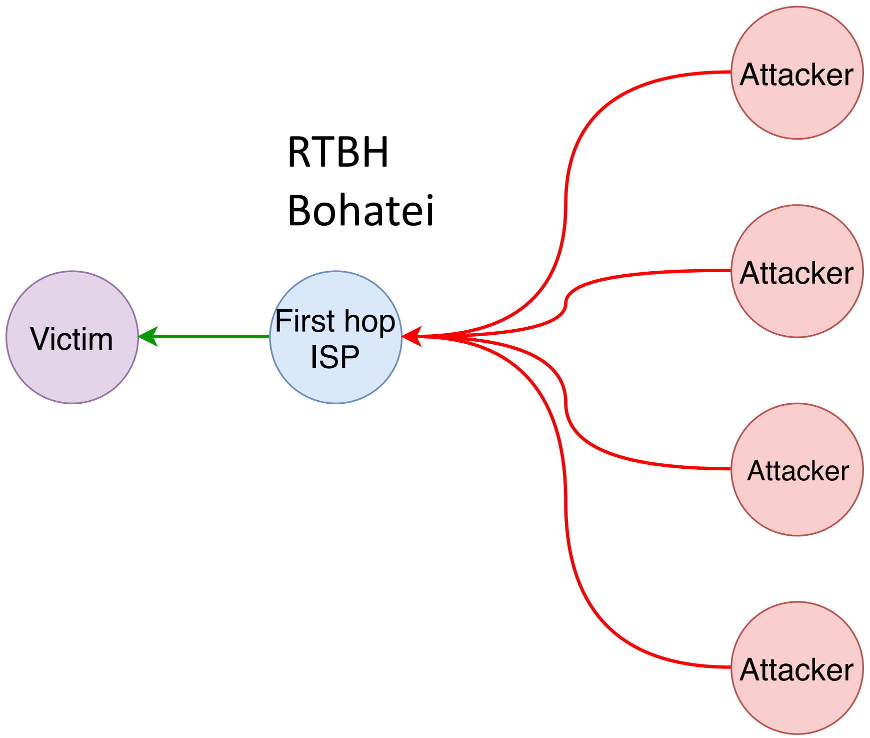
**Figure taken from Arbor Networks worldwide infrastructure security report, 2016*

Existing solutions at victim



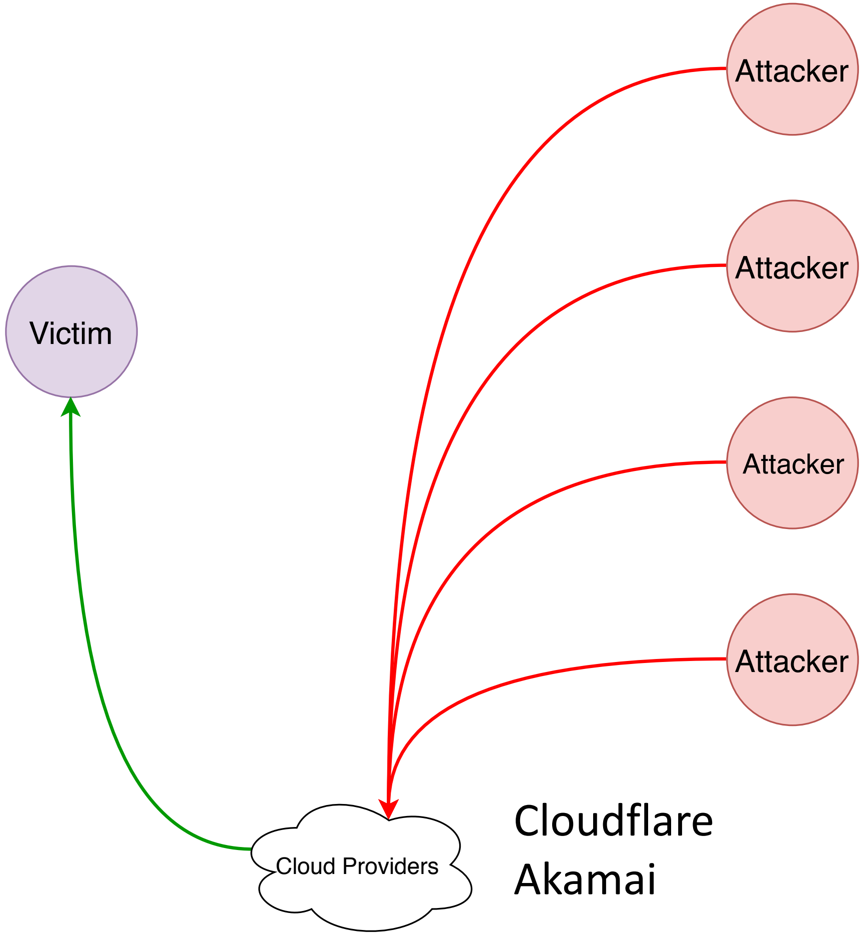
- Solutions such as Bro and Arbor APS deployed at victim
- Filters traffic based on inspection and rules
- Large attacks cannot be filtered as the origin of attack is upstream from victim

Existing solutions at first hop ISP



- Collaboration with ISP via human channels which are error prone and slow
- Crude filtering such as remotely-triggered blackhole saves ISP from attack but cuts victim from internet
- Bohatei uses SDN + NFV to scale defense on demand
- Provides a fine grained traffic control but is resource intensive

Existing solutions at cloud



Cloudflare
Akamai

- Cloud solutions are effective by diverting all victim's traffic towards themselves during an attack
- Apply scrubbing algorithms to remove attack traffic, send the rest to victim
- Ability to handle heavy attacks depends on extent of geo-replication, which is costly

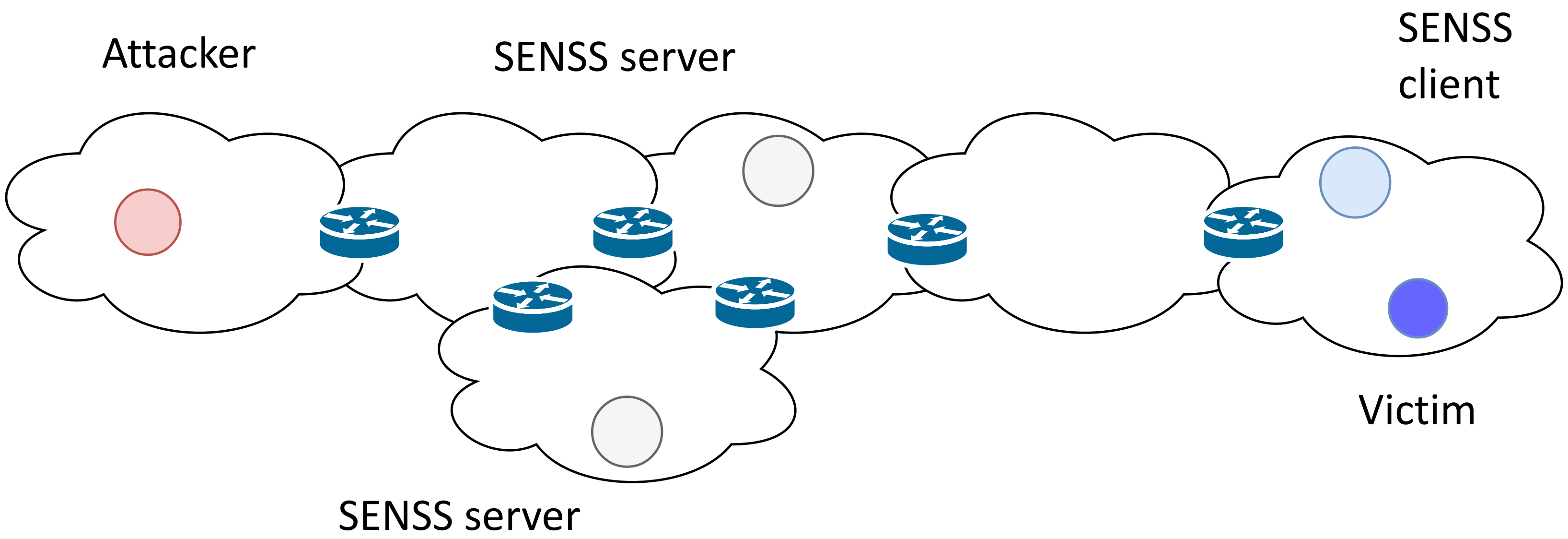
What do we provide?

- SENSS is a collaborative framework which allows victim under attack to communicate with peers or remote networks
- Design is simple
  - SENSS keeps the intelligence at the victim and has simple functionalities at ISP which can be easily implemented in current ISP infrastructure
  - Victim drives decisions to monitor and taking necessary actions to mitigate attacks
  - Victims can create versatile, evolvable and customizable defense for different types of DDoS flavors

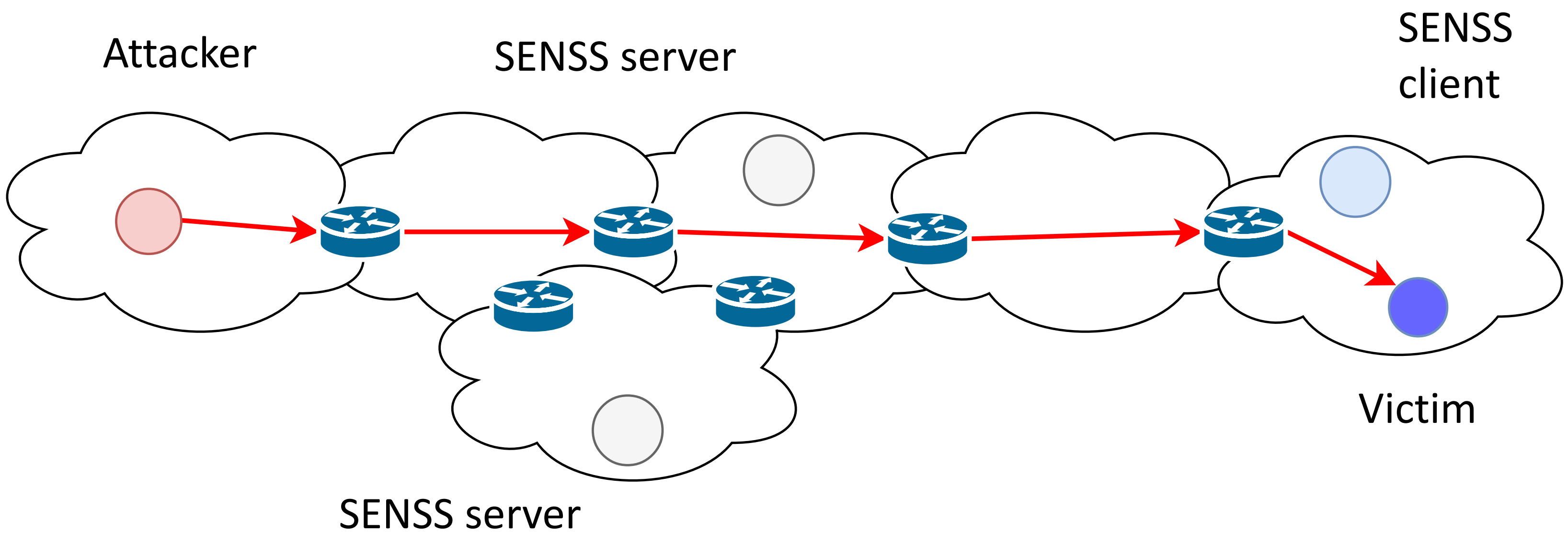
Overview

- Introduction
- SENSS
  - Architecture
  - SENSS API
- SENSS client programs
- Security and robustness
- Evaluation
- Conclusion

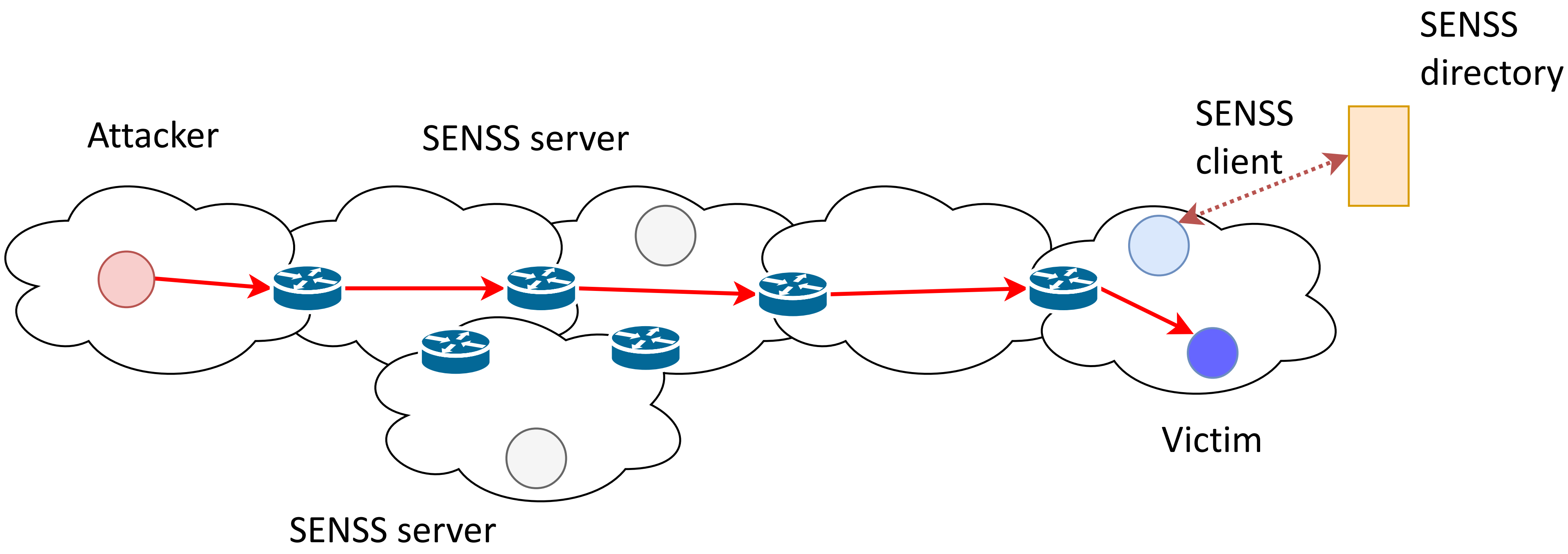
SENSS: Components



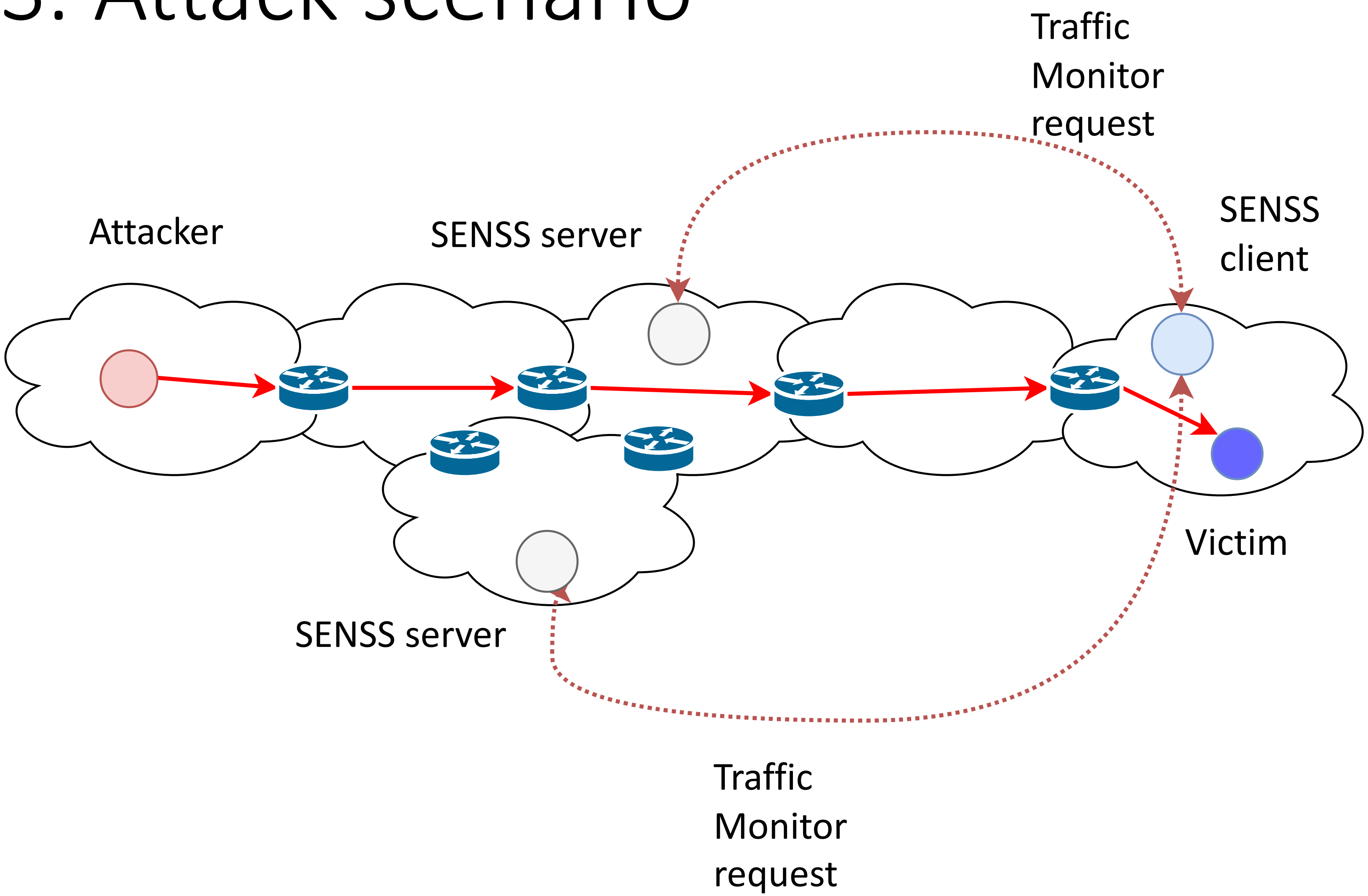
SENSS: Attack scenario



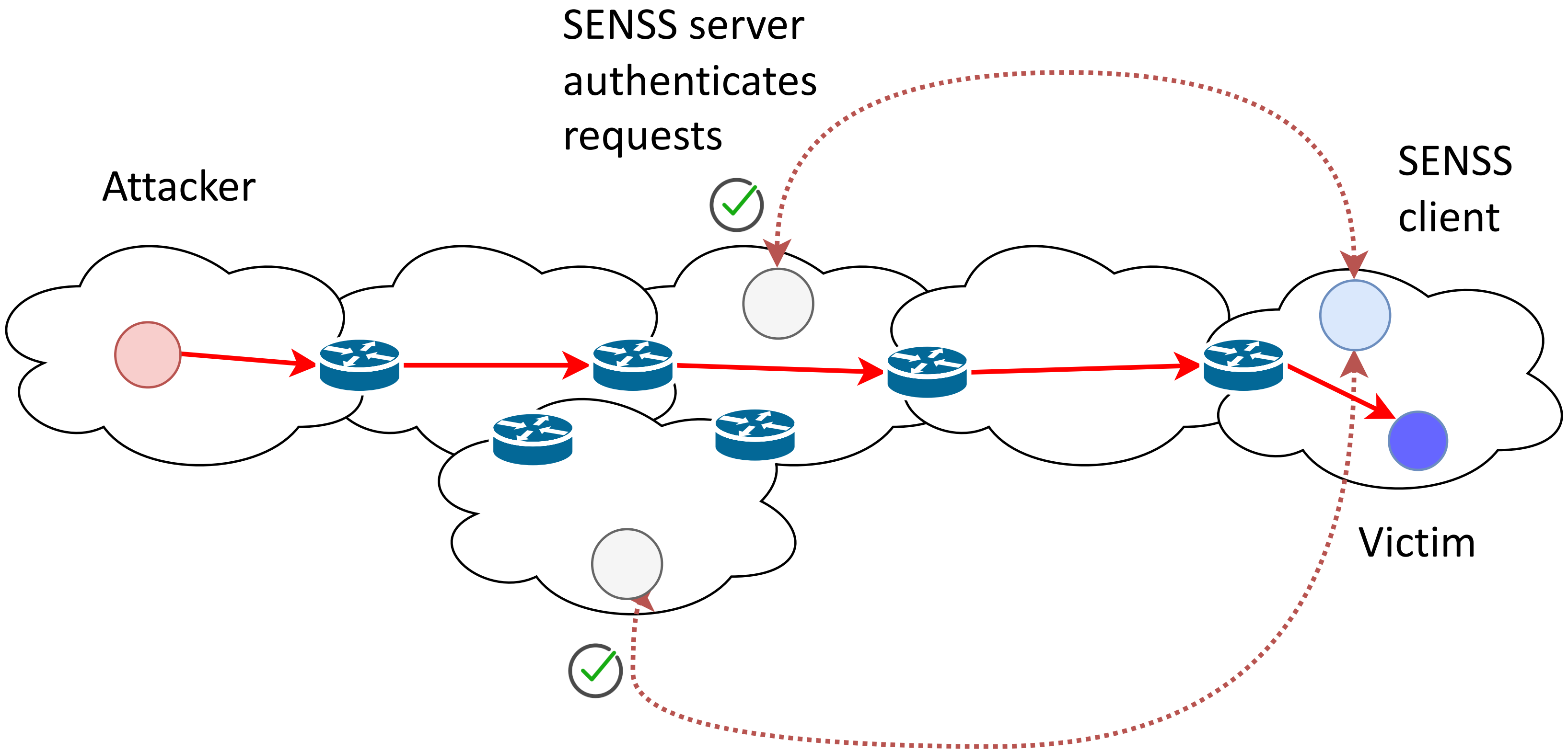
SENSS: Attack scenario



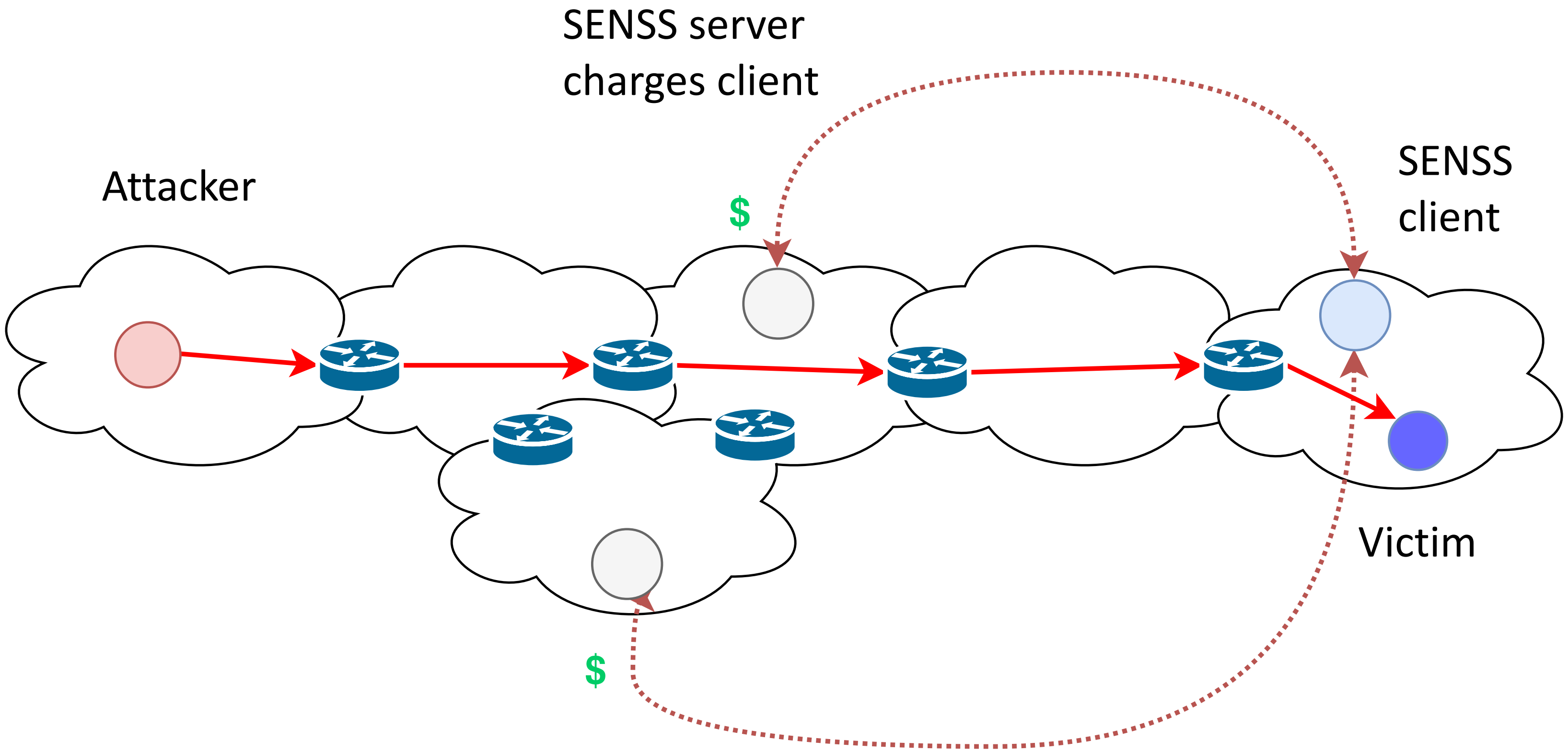
SENSS: Attack scenario



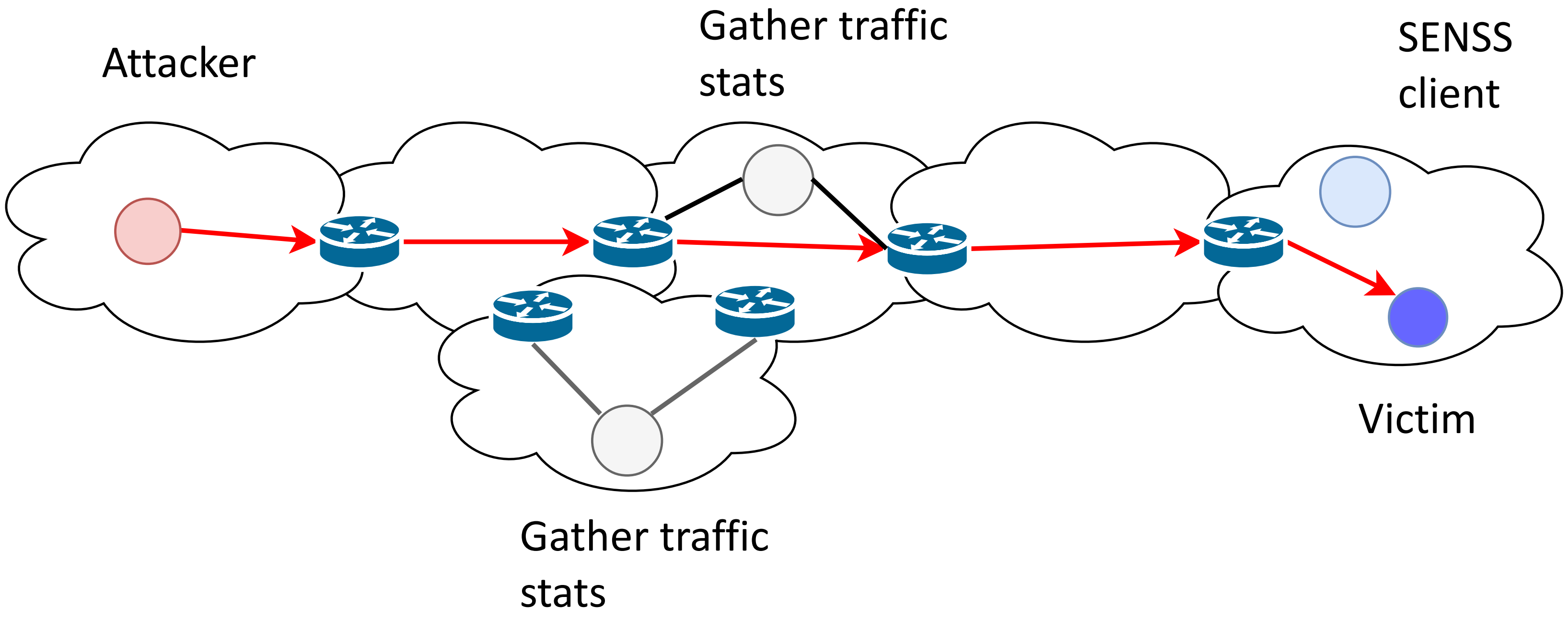
SENSS: Attack scenario



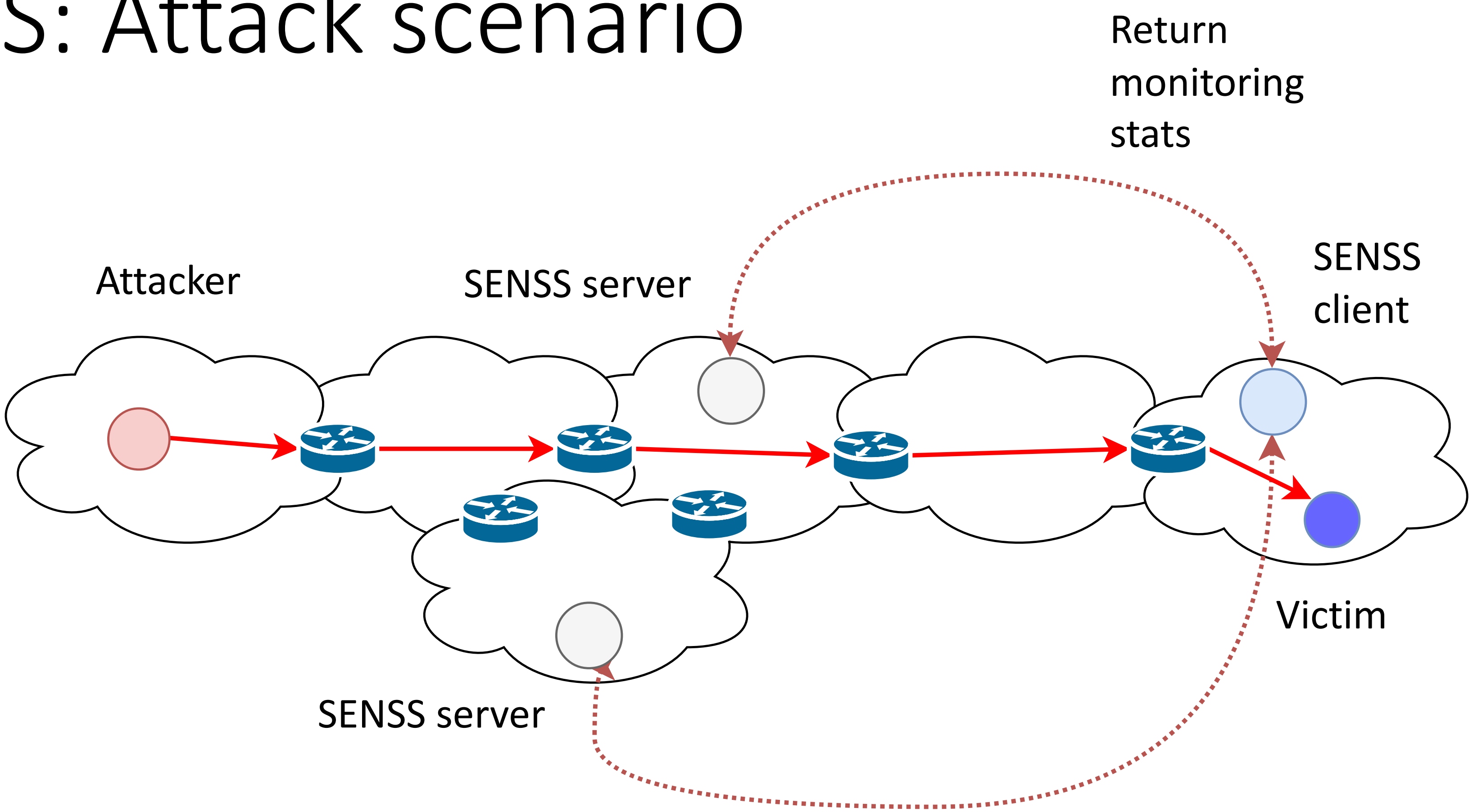
SENSS: Attack scenario



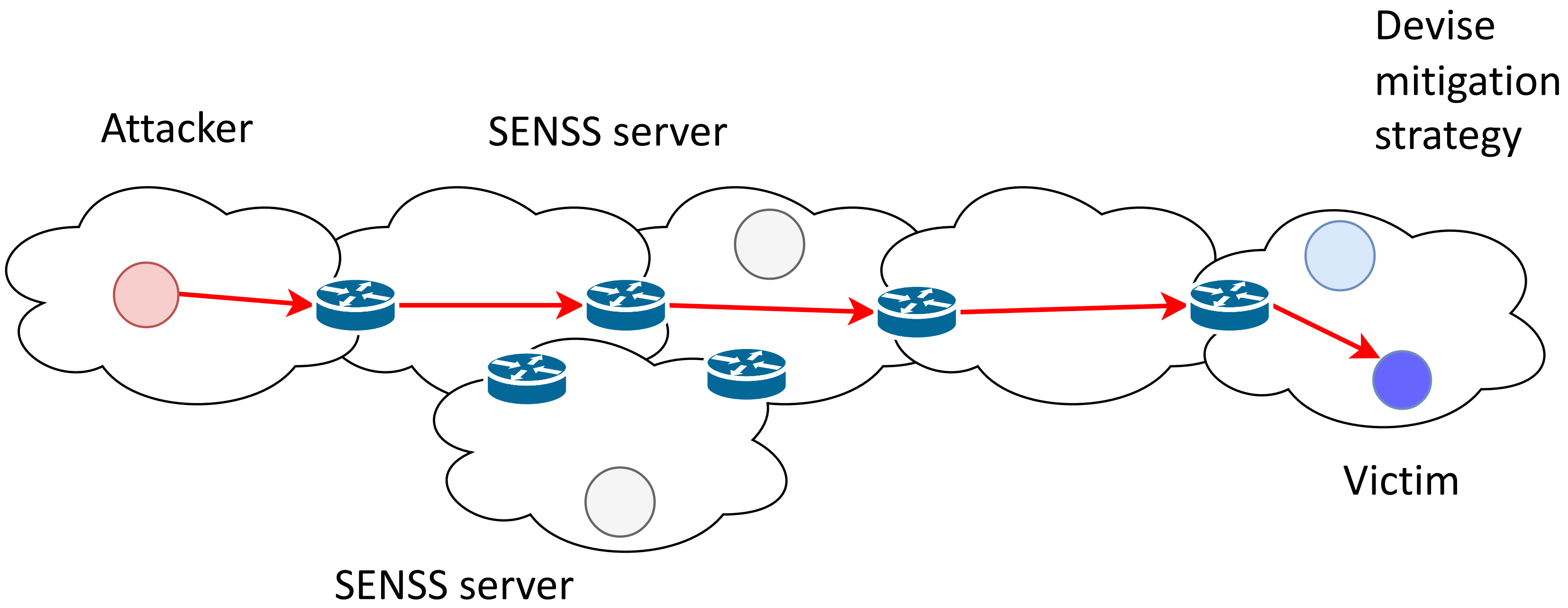
SENSS: Attack scenario



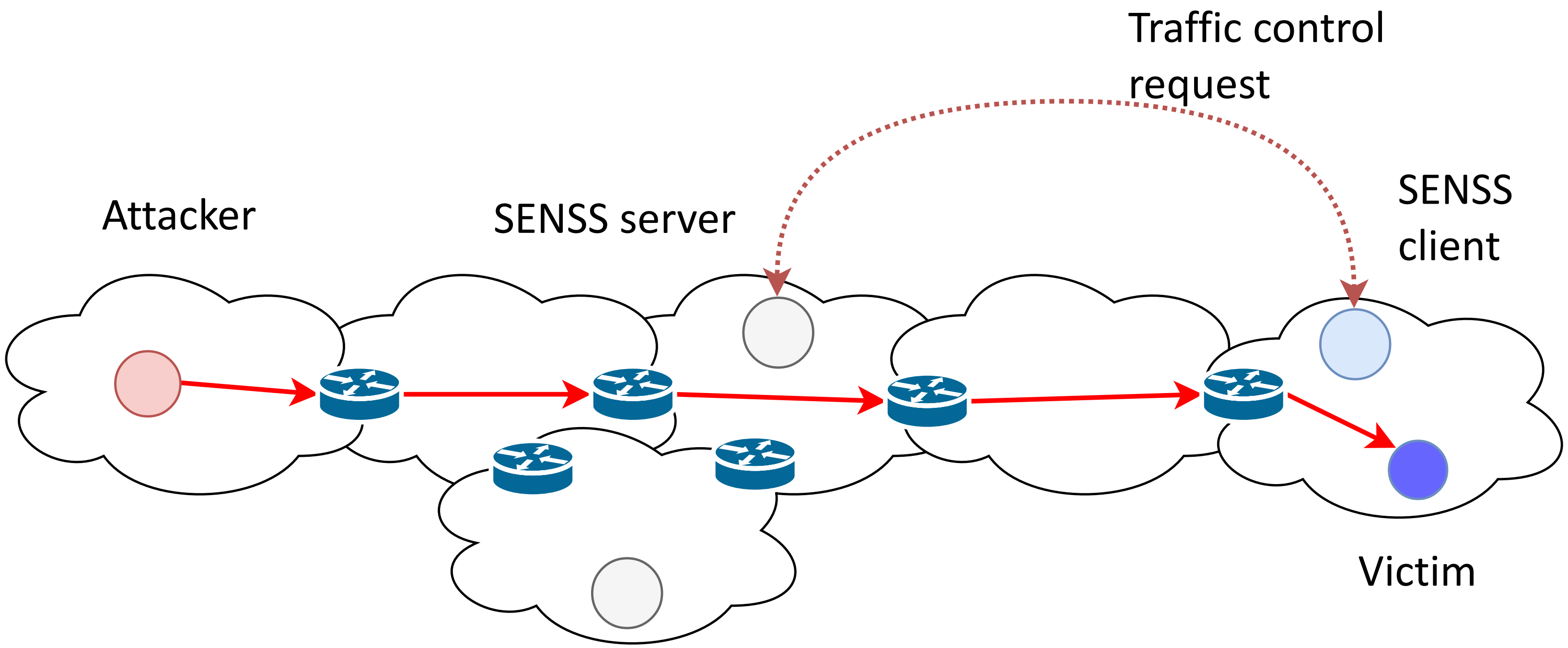
SENSS: Attack scenario



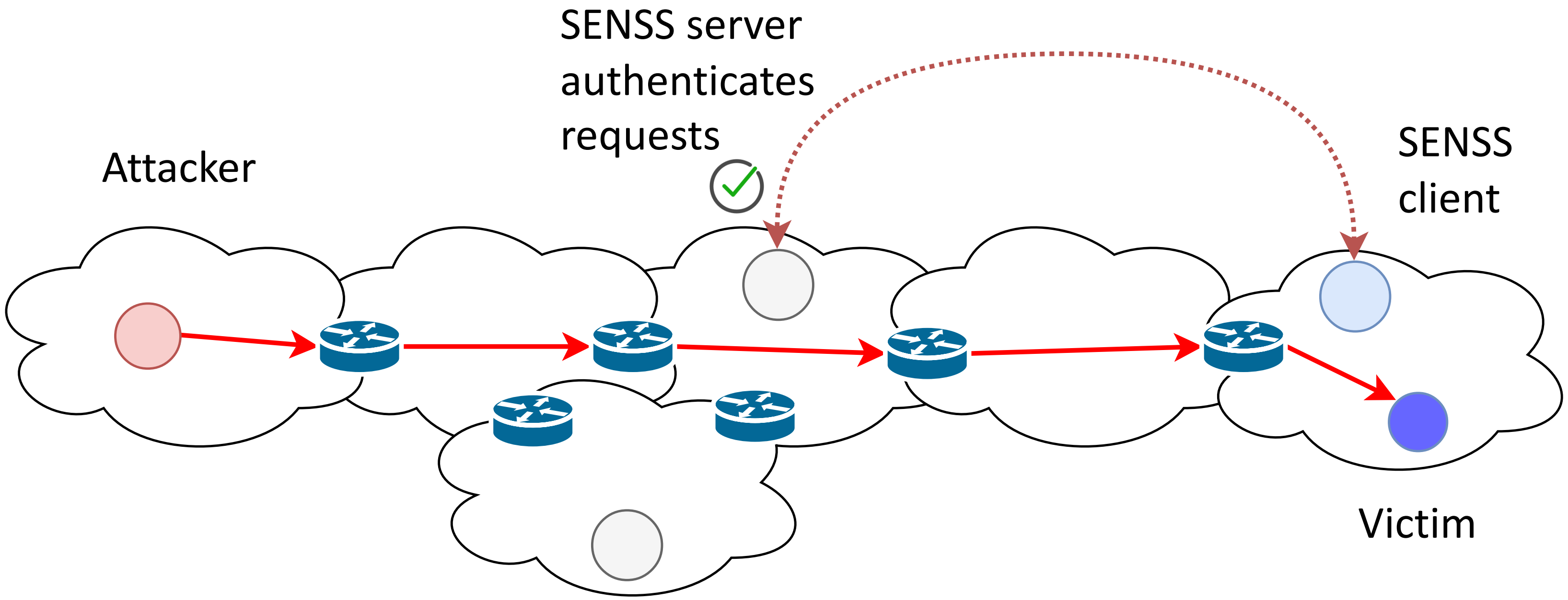
SENSS: Attack scenario



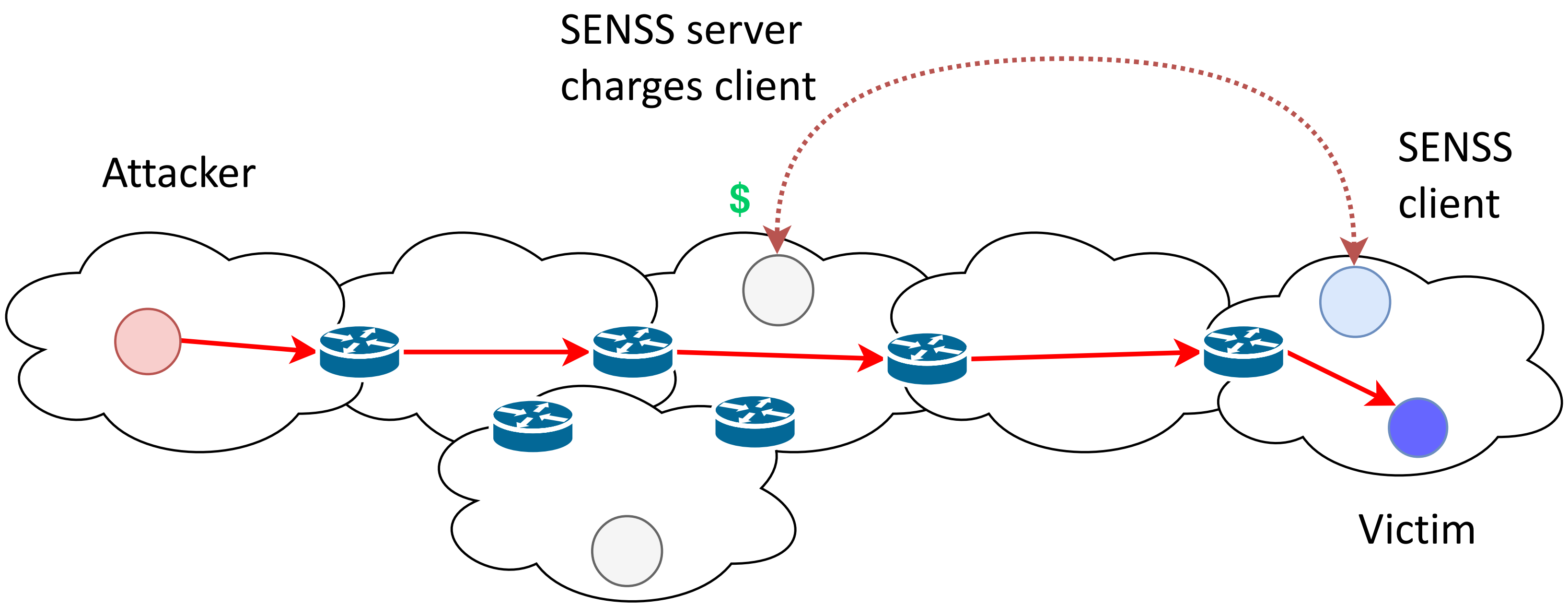
SENSS: Attack scenario



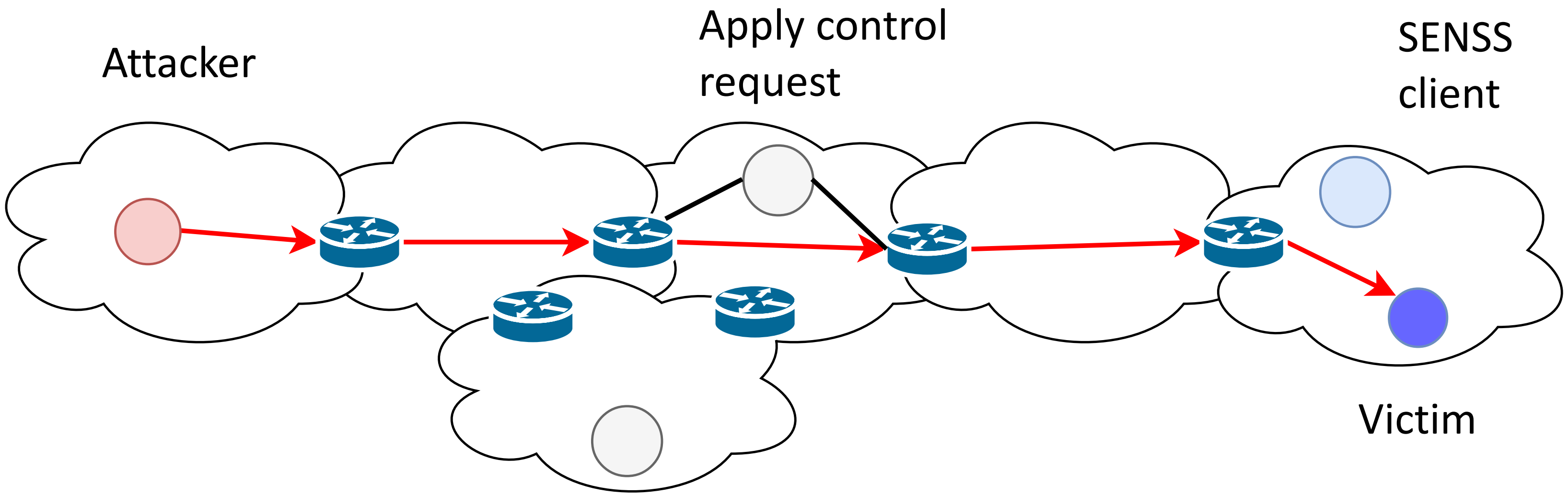
SENSS: Attack scenario



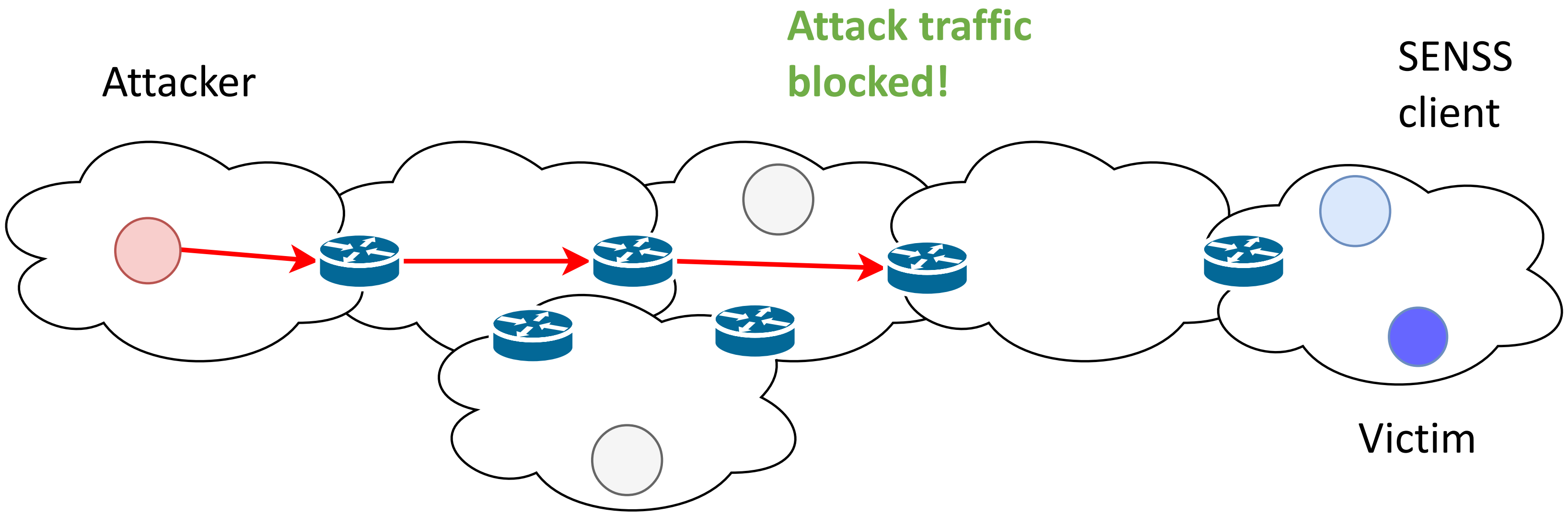
SENSS: Attack scenario



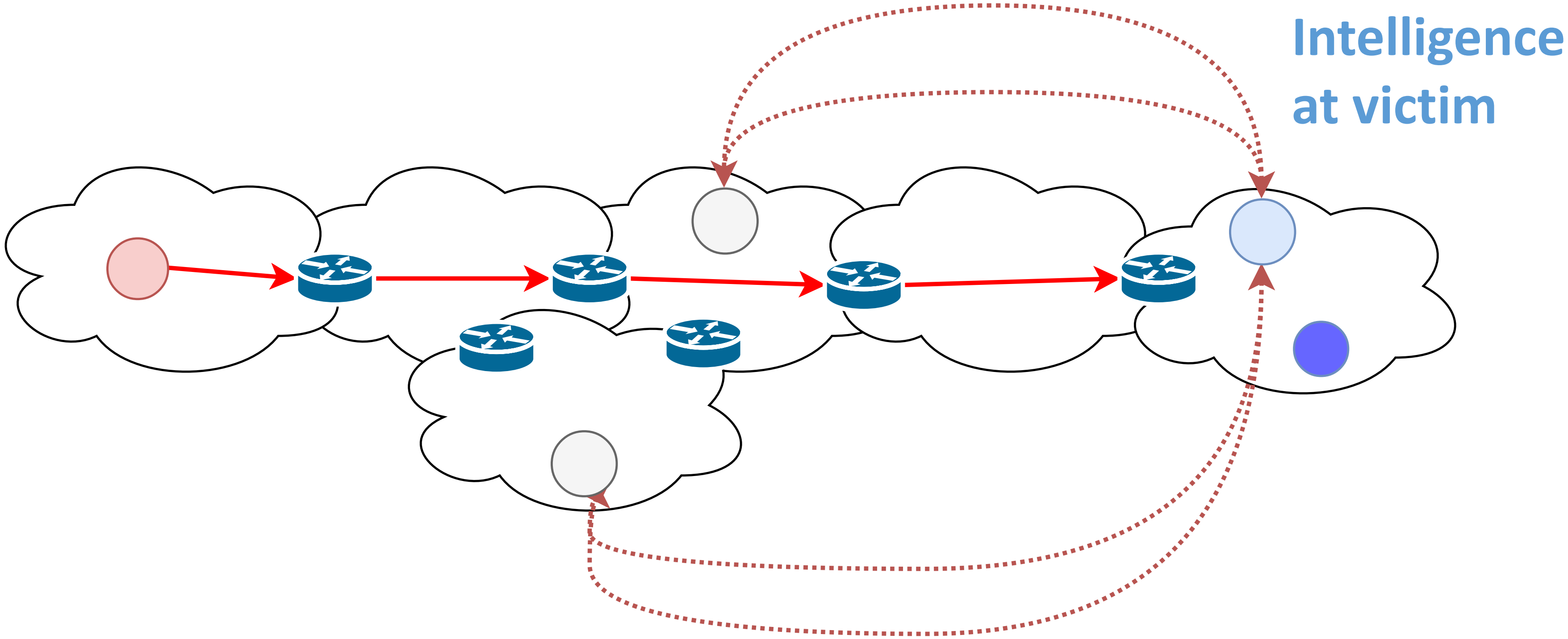
SENSS: Attack scenario



SENSS: Attack blocked

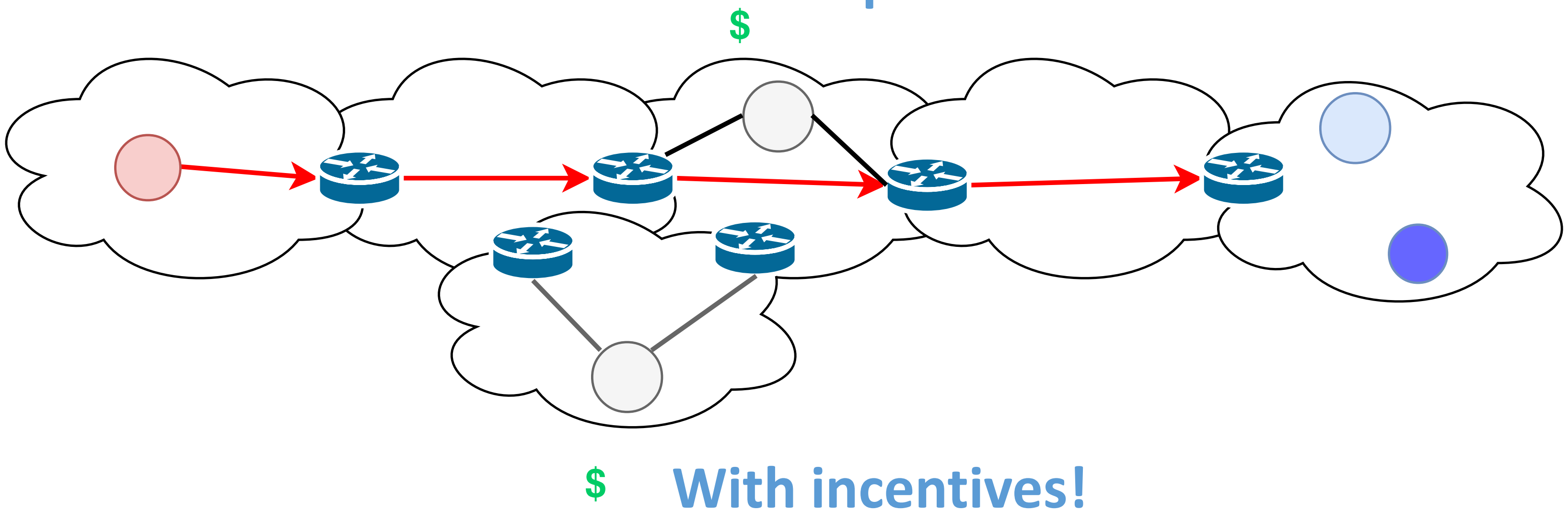


SENSS: Labor division



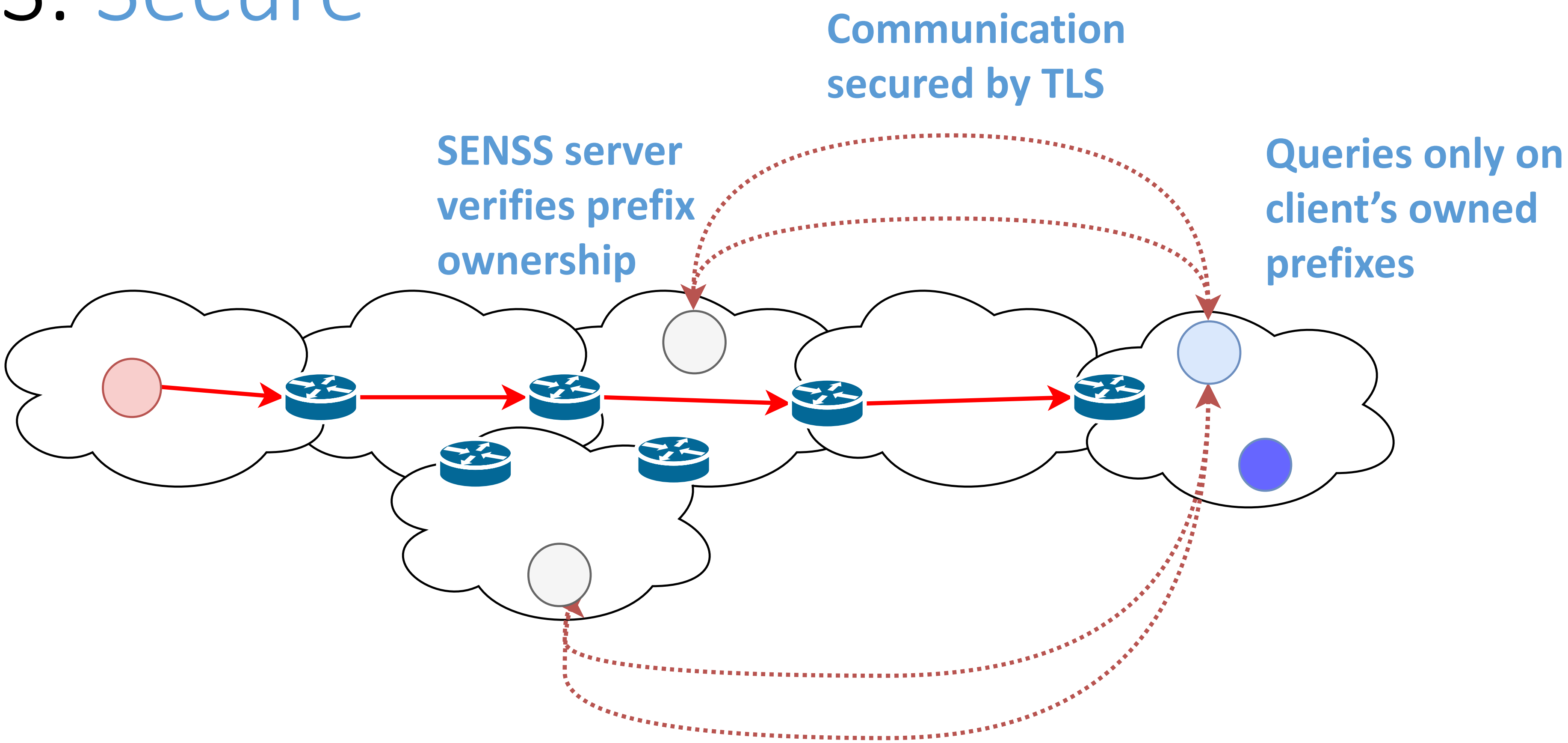
SENSS: Incentives for ISPs

Simple implementation at ISP



With incentives!

SENSS: Secure



SENSS API

Type	**Response from SENSS server**
Traffic Query	Traffic stats matching predicates

SENSS API

Type	Response from SENSS server
Traffic Query	Traffic stats matching predicates
Route Query	AS paths from SENSS server to prefix

SENSS API

Type	Response from SENSS server
Traffic Query	Traffic stats matching predicates
Route Query	AS paths from SENSS server to prefix
Traffic filter	Adds filter matching predicate

SENSS API

Type	Response from SENSS server
Traffic Query	Traffic stats matching predicates
Route Query	AS paths from SENSS server to prefix
Traffic filter	Adds filter matching predicate
Route demote	Demotes AS path from SENSS server to prefix with certain path segment

SENSS API

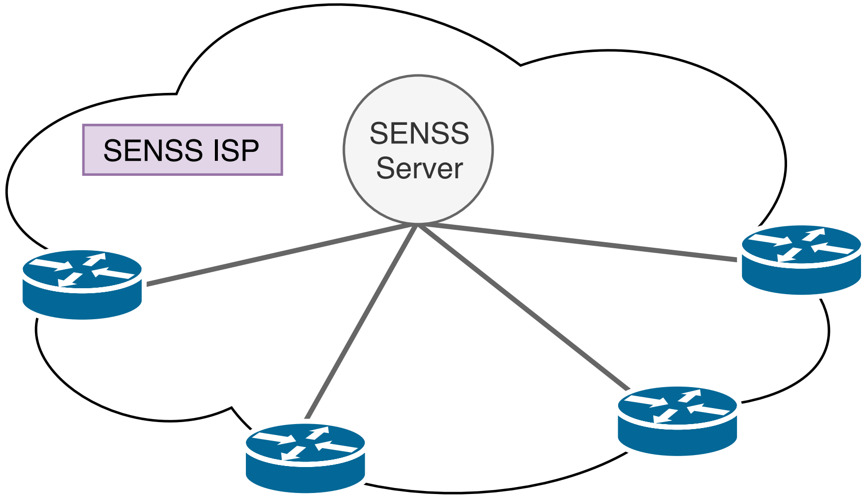
Type	Response from SENSS server
Traffic Query	Traffic stats matching predicates
Route Query	AS paths from SENSS server to prefix
Traffic filter	Adds filter matching predicate
Route demote	Demotes AS path from SENSS server to prefix with certain path segment

Each traffic query/control consists of a predicate matching flow(s)

- Supports various packet header fields
- Different packet header fields can be combined using negation, conjunction, disjunction and wildcard

SENSS Server Implementation

- Queries to SENSS server can be implemented using Openflow or Netflow+ACL
- SENSS server receives requests from clients, authenticates and sends appropriate replies
- SENSS server also co-ordinates with various border routers within the same ISP and gathers statistics

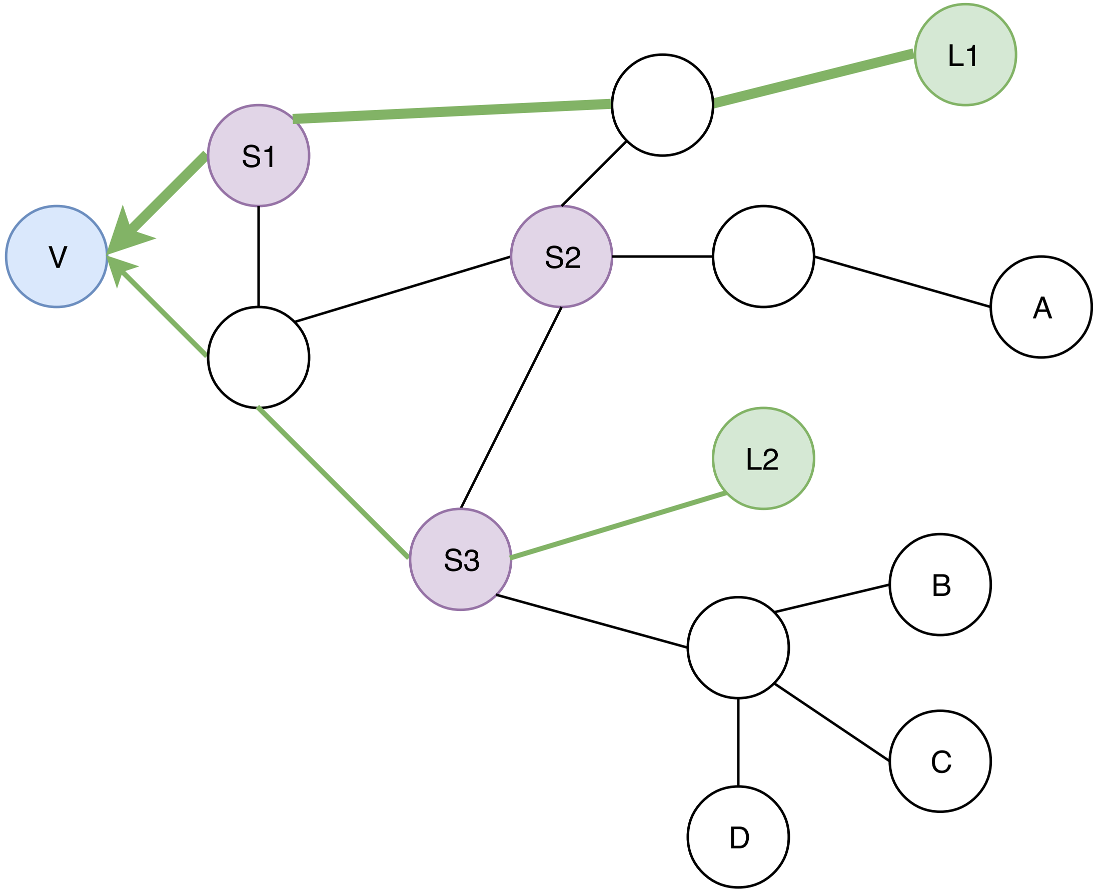


Overview

- Introduction
- SENSS
  - Architecture
  - SENSS API
- SENSS client programs
- Security and robustness
- Evaluation
- Conclusion

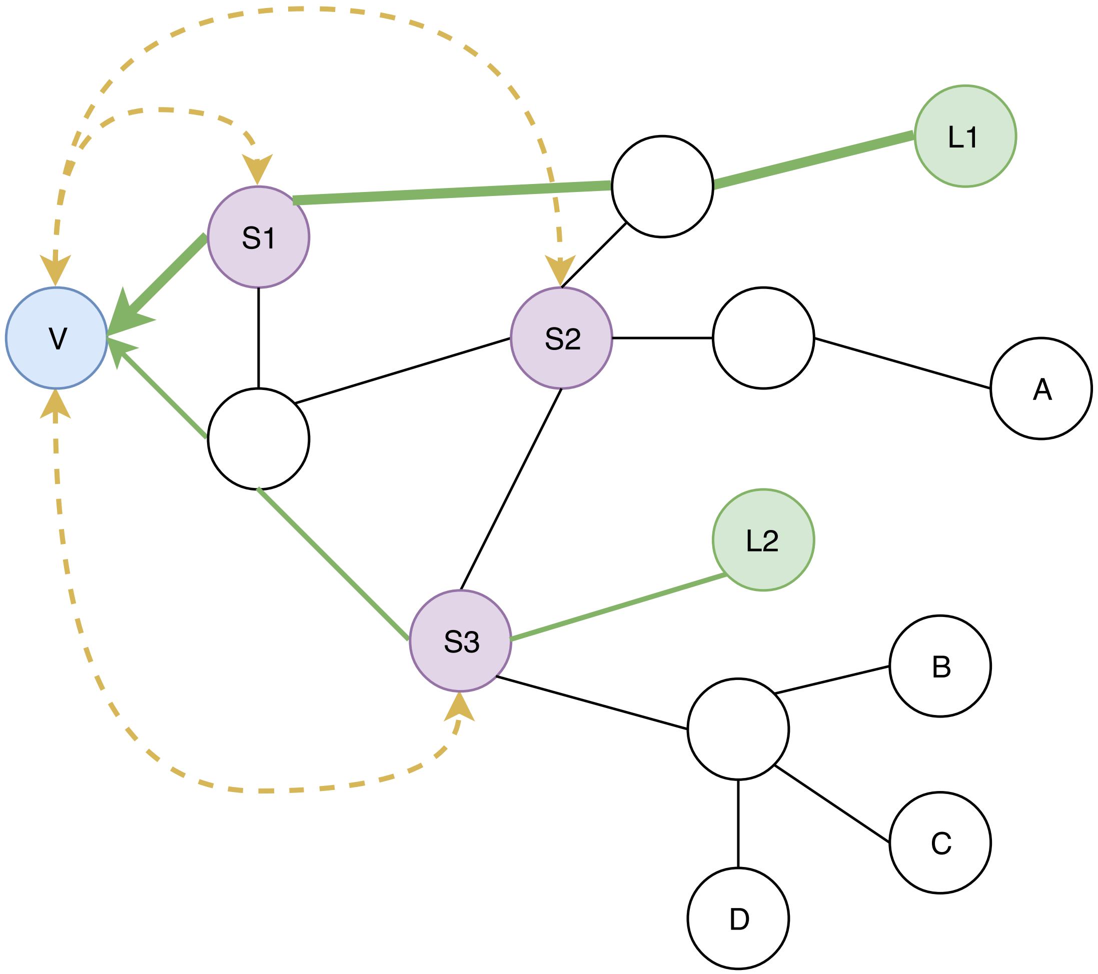
DDoS without signature attack

Legit traffic from L1
and L2



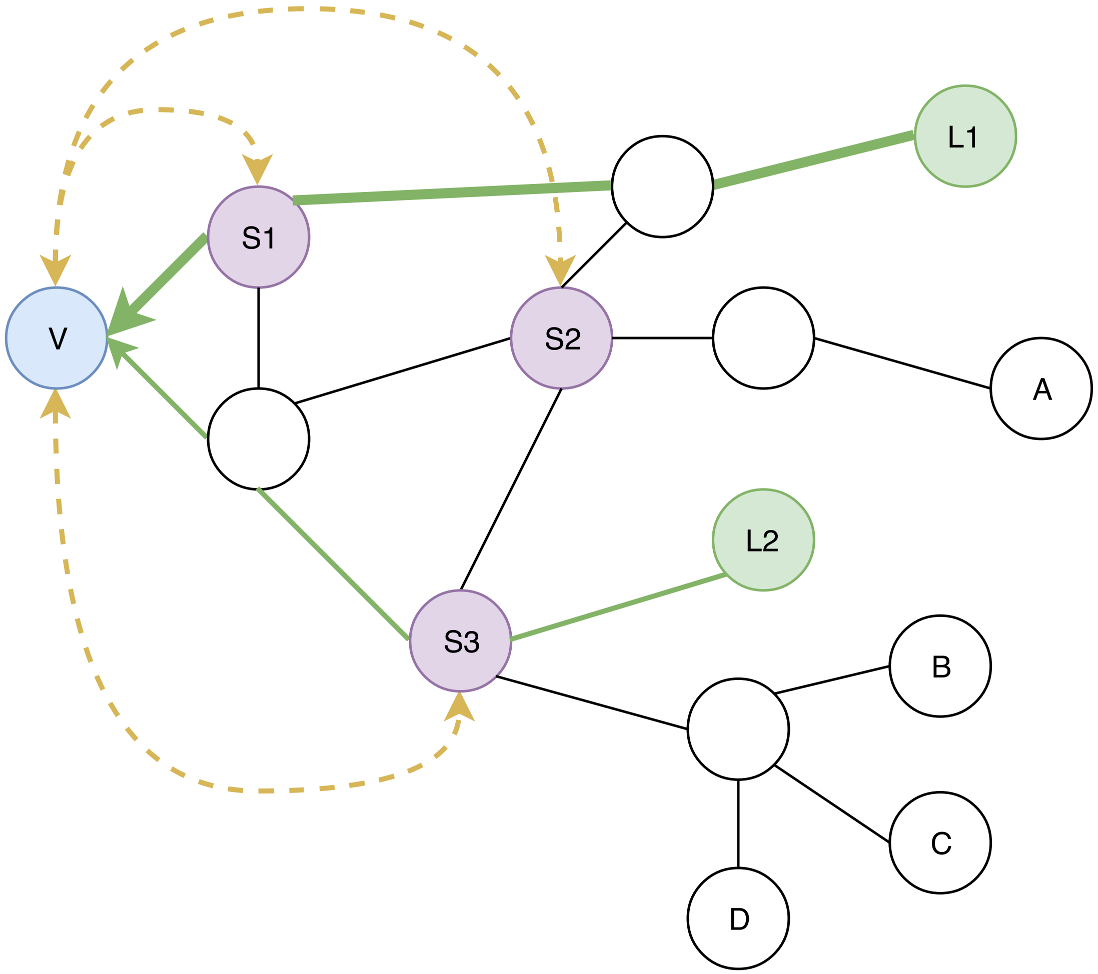
DDoS without signature attack

Periodic traffic query to S1, S2 and S3



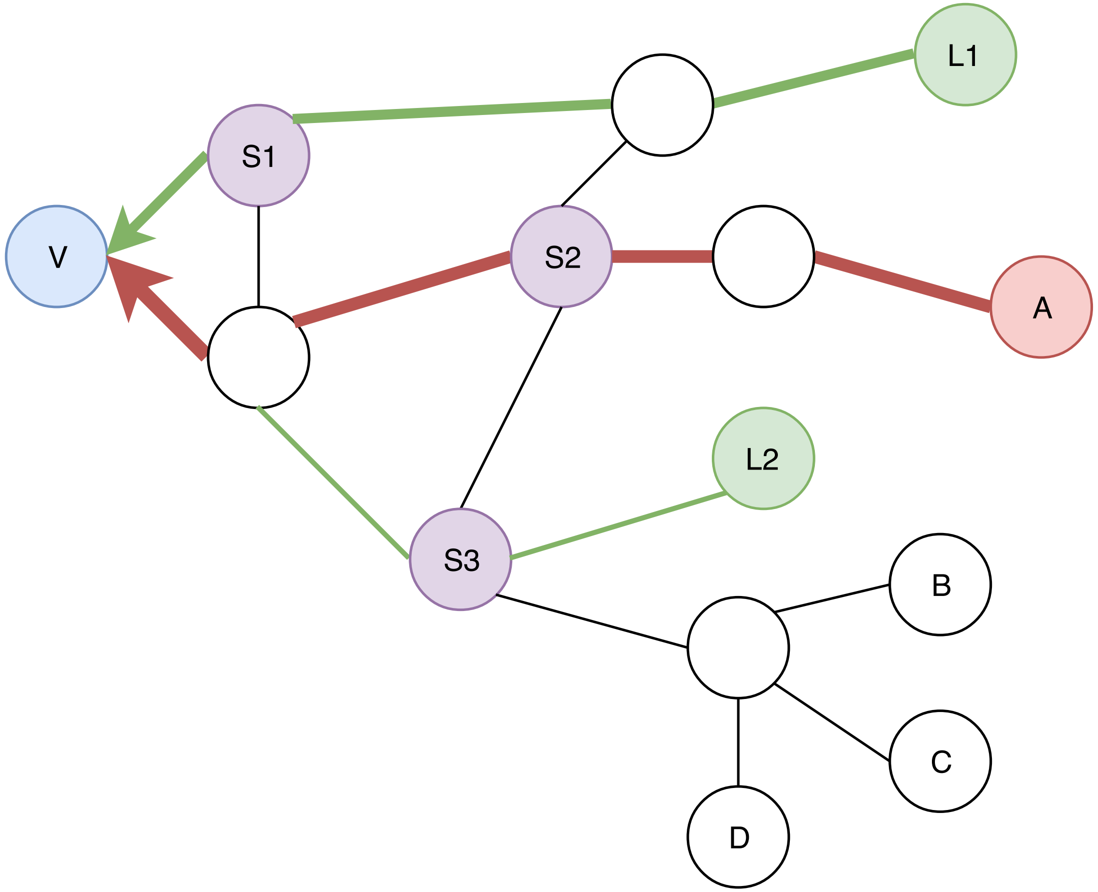
DDoS without signature attack

Replies
S1: 1000 Mbps
S2: 0 Mbps
S3: 400 Mbps

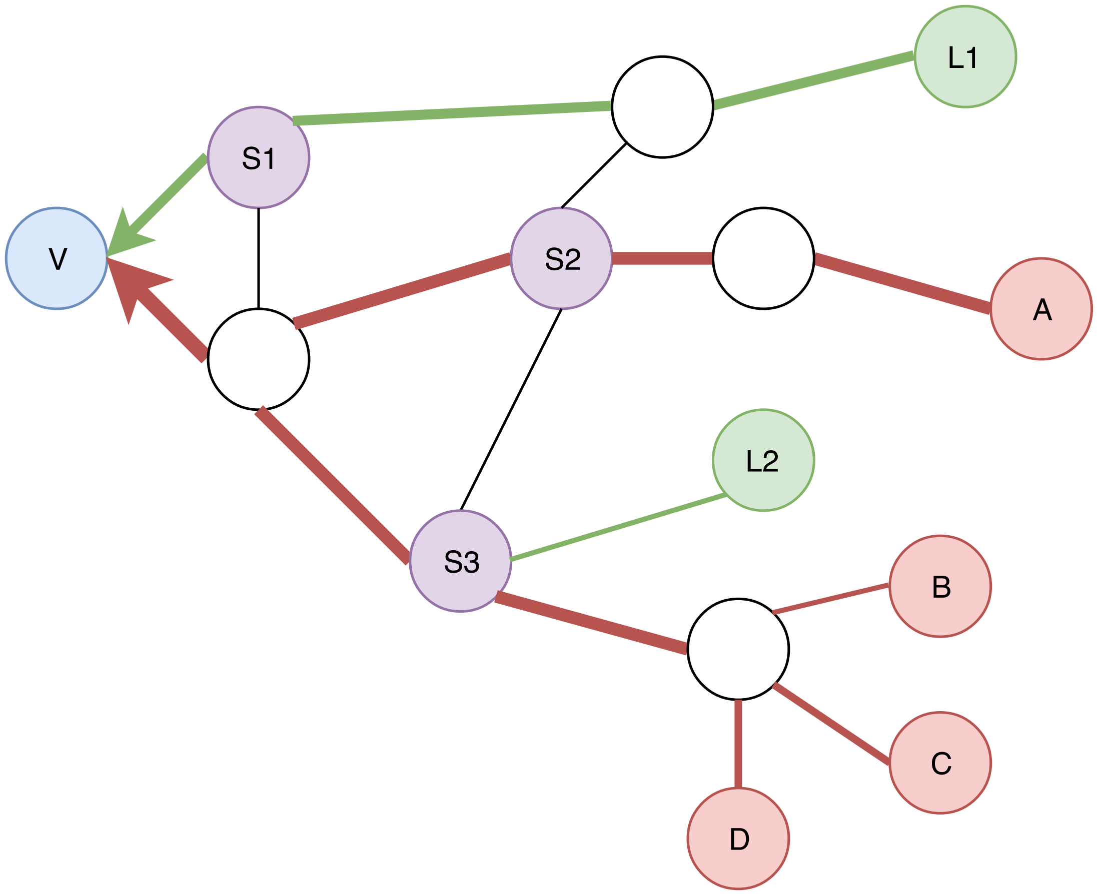


DDoS without signature attack

Attack from A



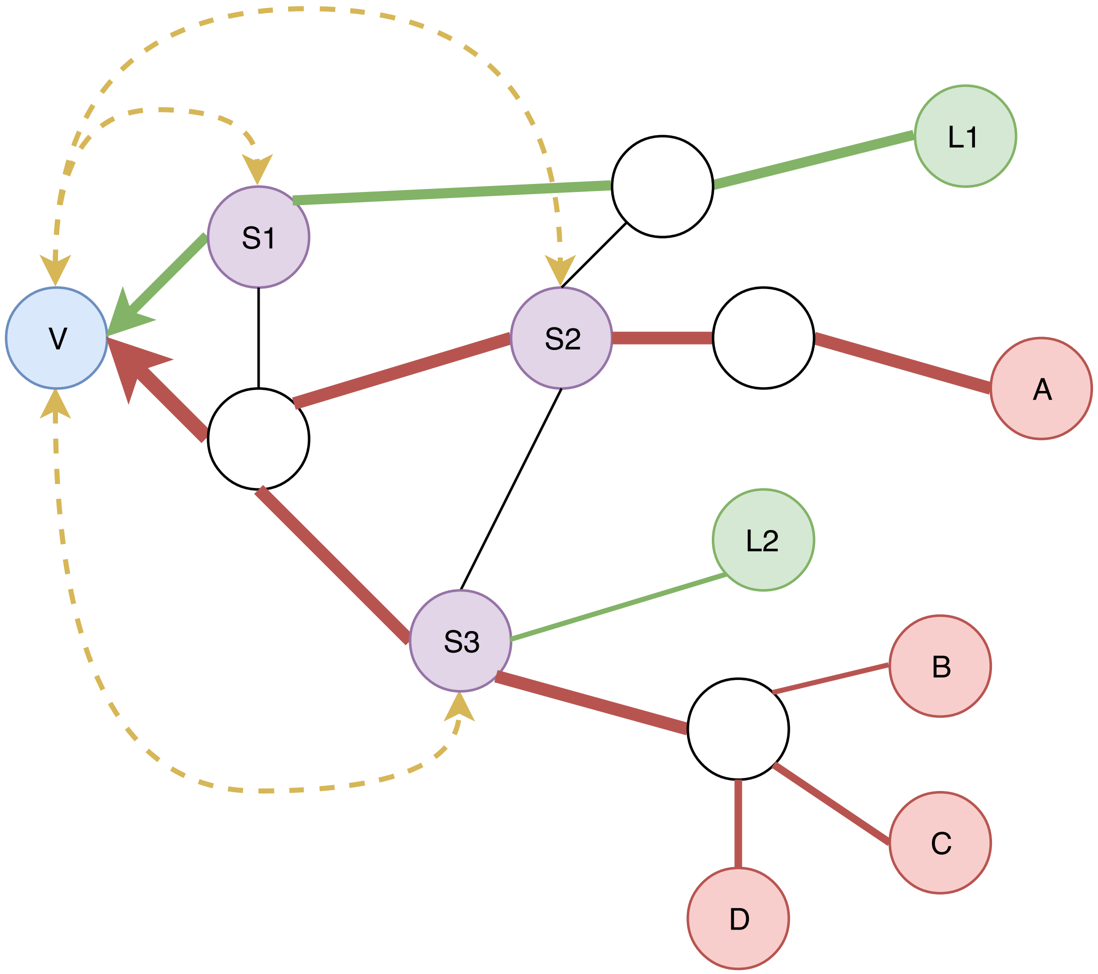
DDoS without signature attack



More attack from B, C and D

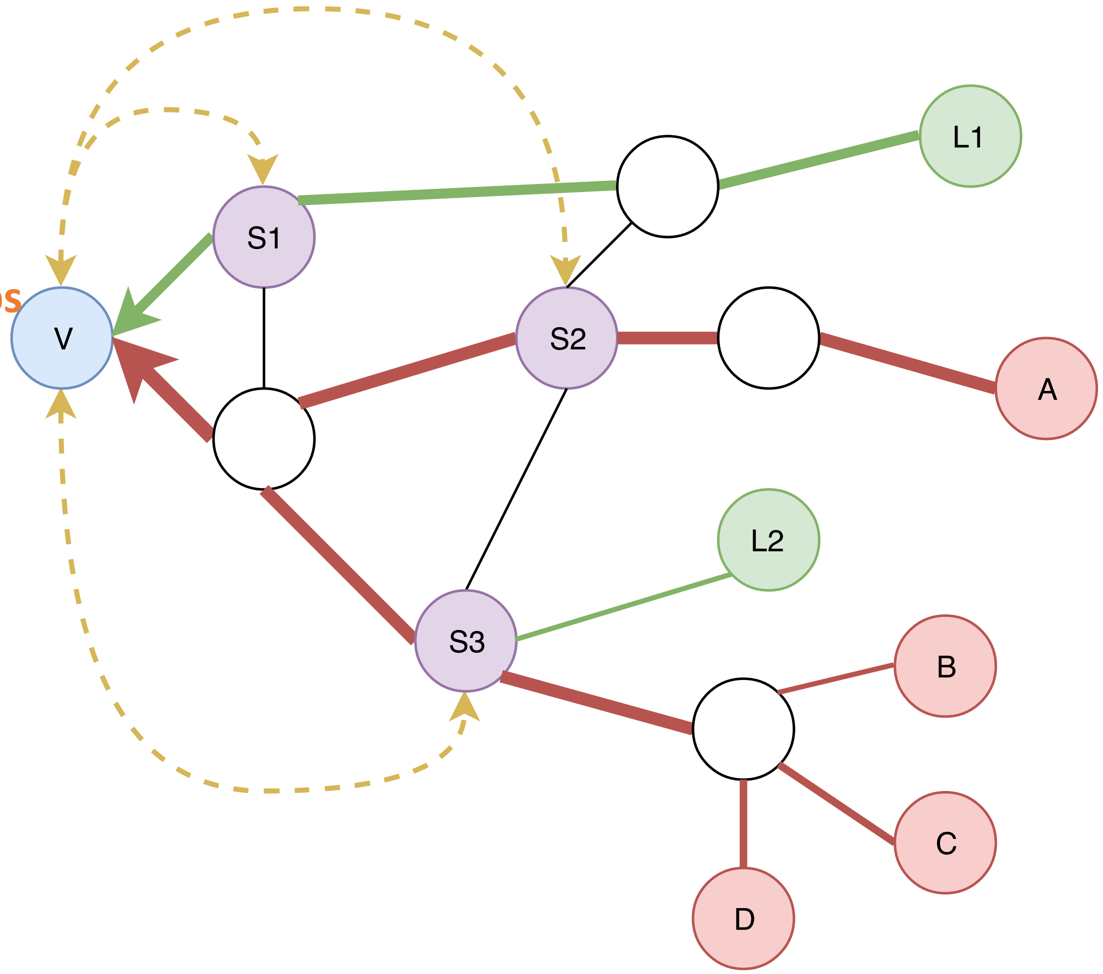
DDoS without signature attack

Periodic traffic query to S1, S2 and S3



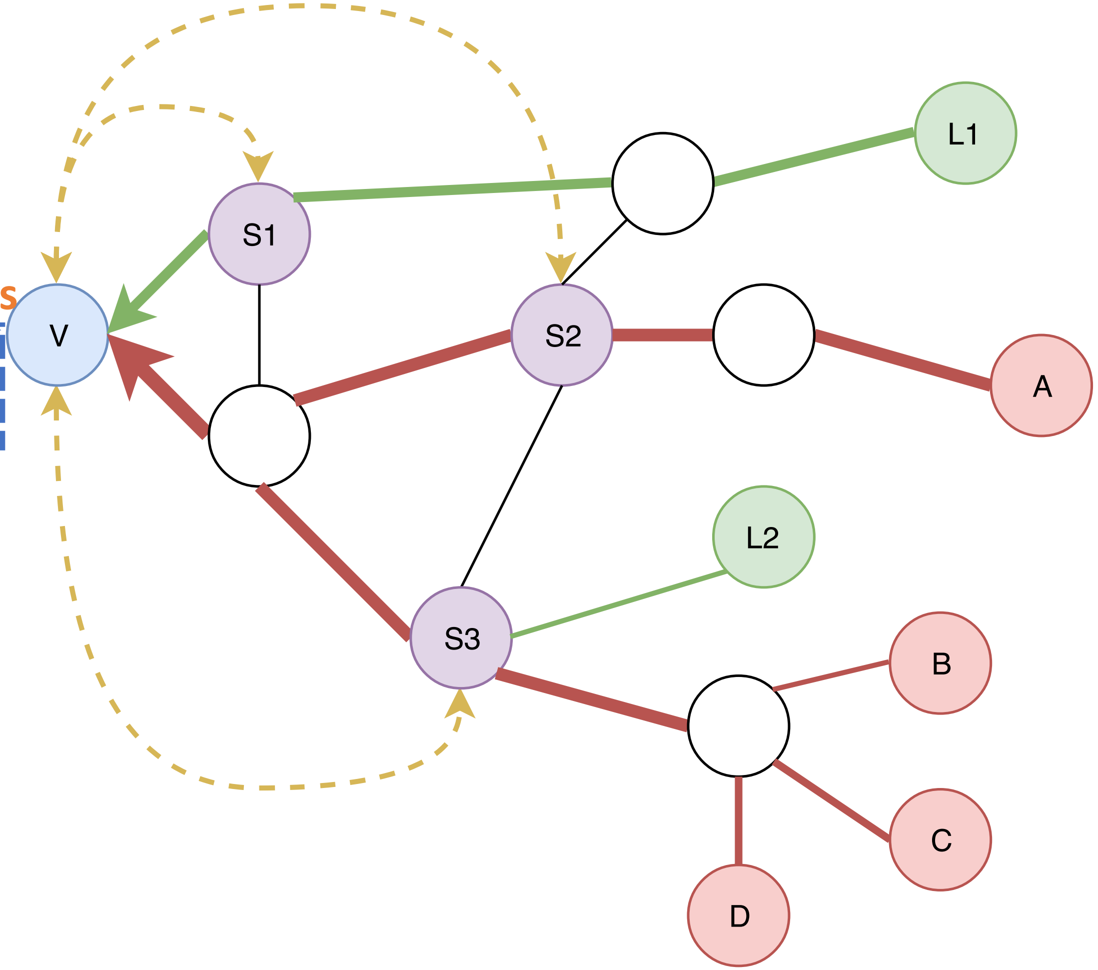
DDoS without signature attack

Replies
S1: 1000 Mbps
S2: 500 Mbps
S3: 750 Mbps



DDoS without signature attack

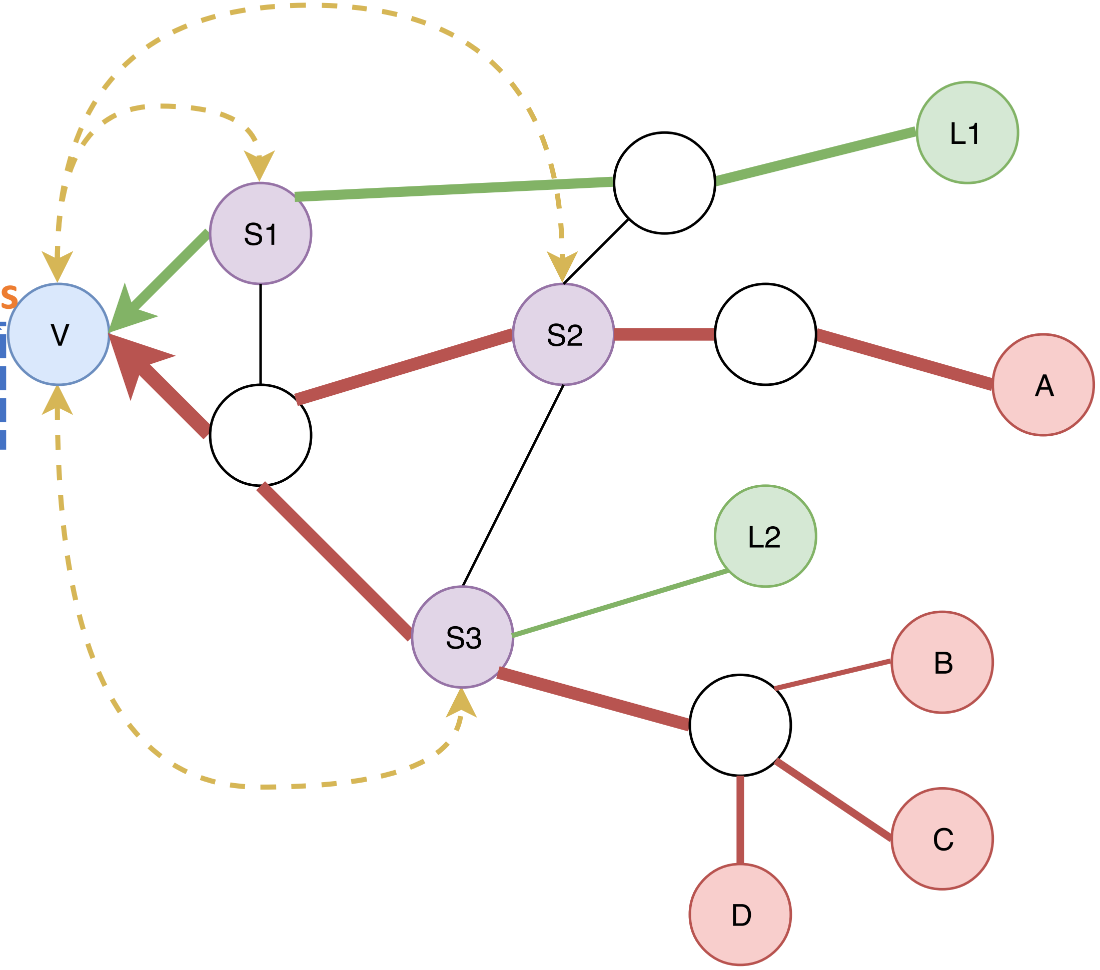
Replies	Replies
S1: 1000 Mbps	S1: 1000 Mbps
S2: 0 Mbps	S2: 500 Mbps
S3: 400 Mbps	S3: 750 Mbps



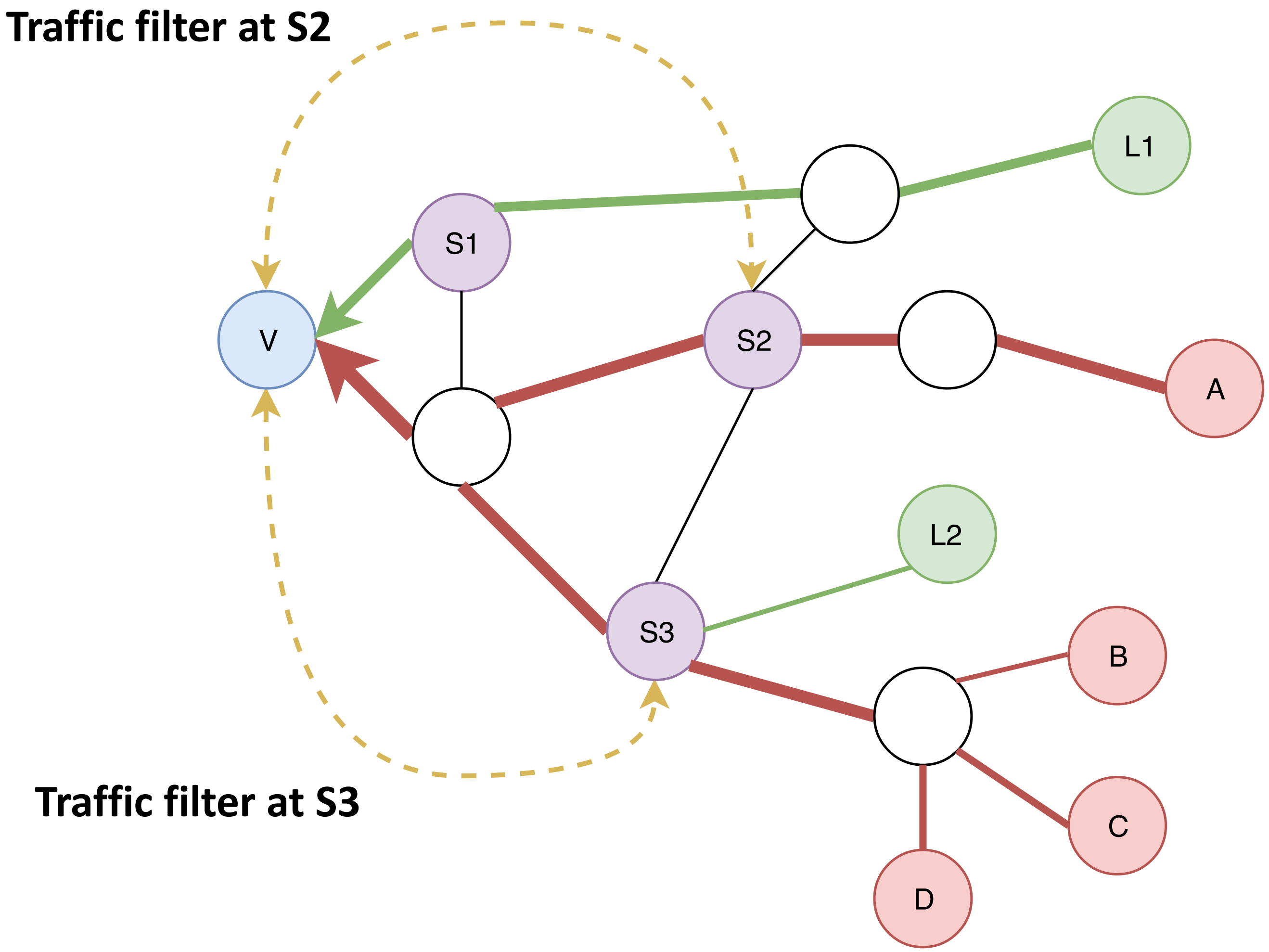
DDoS without signature attack

Replies	Replies
S1: 1000 Mbps	S1: 1000 Mbps
S2: 0 Mbps	S2: 500 Mbps
S3: 400 Mbps	S3: 750 Mbps

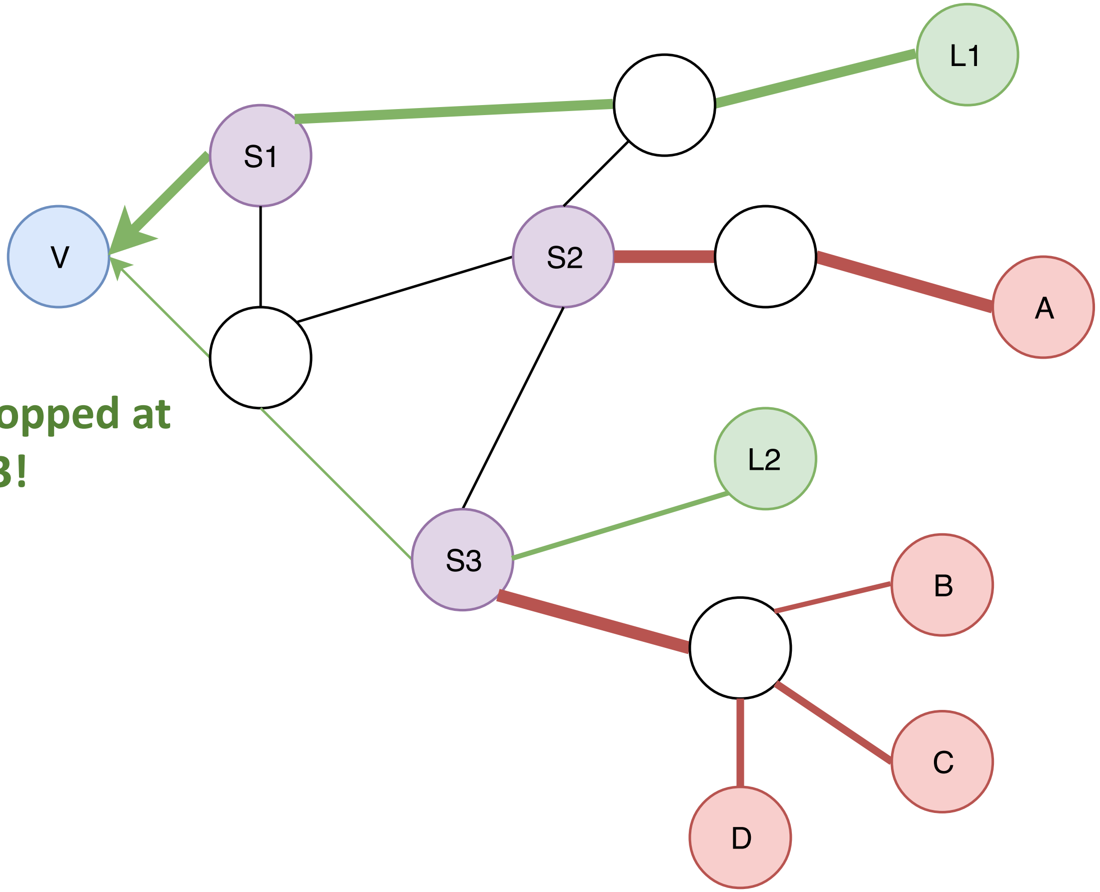
Unusual traffic
from S2 and S3



DDoS without signature attack



DDoS without signature attack



Attack stopped at S2 and S3!

Overview

- Introduction
- SENSS
  - Architecture
  - SENSS API
- SENSS client programs
- Security and robustness
- Evaluation
- Conclusion

Securing communication

- SENSS allows client to issue requests only to its own prefixes
  - SENSS client binds a proof of ownership certificate with every request
- Proof can be created using RPKI Route Origin Authorization (ROA) certificates
  - Alternatively we can issue custom certificates
- Communication between SENSS client and SENSS server is secured using TLS and occurs over HTTPS
  - If the privacy of key is compromised, SENSS server can purge all existing client requests

Challenges

- Router's TCAM space is limited
    - Coarse rules are enough to mitigate large volumetric attack
    - Finer rules can be prevented by SENSS ISP's or discourage users by charging higher prices
- ISP's privacy concerns
    - Traffic replies can contain anonymized ID's to cover neighboring peers
- ISP is in control
    - Can reject demote requests which may not be optimal

Handling misbehavior

- **SENSS clients** have low incentive to misbehave
  - Excessive requests are unlikely as clients need to pay for each request
  - Requests can be made only for their own prefixes
- **SENSS servers** could lie about observations and/or fail to implement control actions
  - Legacy: Lie about client’s traffic and make it look smaller, increasing the cost of client but does not drop traffic
  - Dropper: Lie about client’s traffic and make it look larger causing client to issue traffic control to drop traffic
    - But dropper liars are already on the path of traffic, SENSS does not make it worst

Overview

- Introduction
- SENSS
  - Architecture
  - SENSS API
- SENSS client programs
- Security and robustness
- Evaluation
- Conclusion

Evaluation objectives

- Extent of SENSS adoption by ISP required for effective protection?
- How will different customers benefit from SENSS adoption?
- SENSS comparison with existing cloud solutions

Evaluation objectives

- Extent of SENSS adoption by ISP required for effective protection?
  - 0.7—3.8% deployment of SENSS in large ISPs can protect most customers
- How will different customers benefit from SENSS adoption?

- SENSS comparison with existing cloud solutions

Evaluation objectives

- Extent of SENSS adoption by ISP required for effective protection?
    - 0.7—3.8% deployment of SENSS in large ISPs can protect most customers
- How will different customers benefit from SENSS adoption?
    - All direct single homed customers of SENSS ISPs are protected from direct floods and reflector attacks
    - 90% of direct multi homed or remote customers are protected from floods without signature and reflector attacks with just 1—3.8% of SENSS adoption
- SENSS comparison with existing cloud solutions

Evaluation objectives

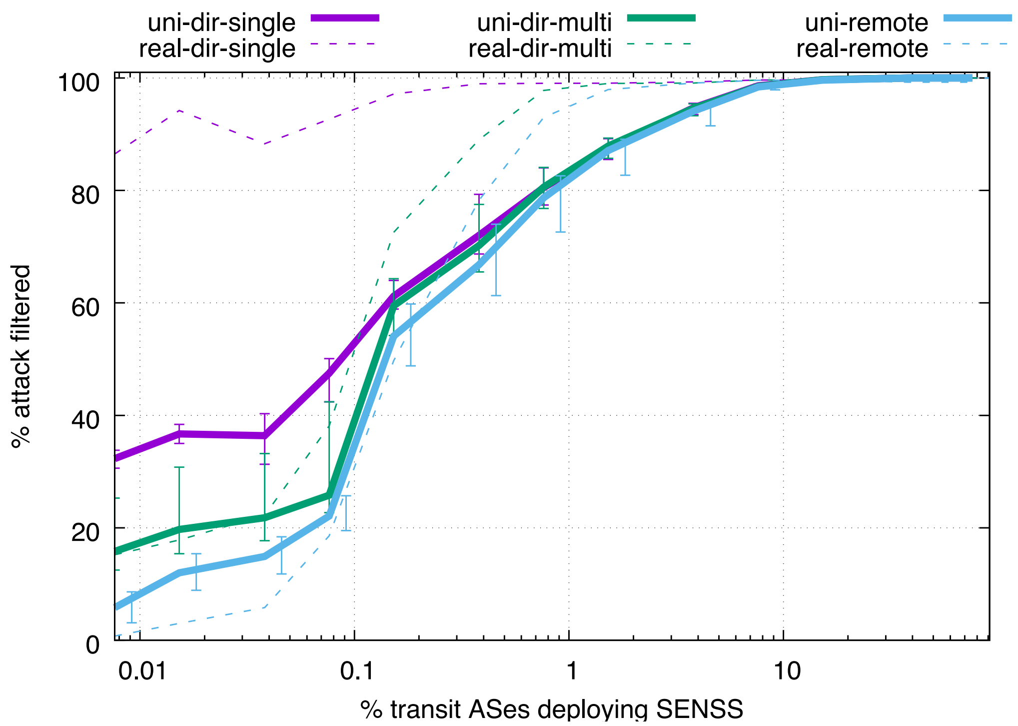
- Extent of SENSS adoption by ISP required for effective protection?
  - 0.7—3.8% deployment of SENSS in large ISPs can protect most customers
- How will different customers benefit from SENSS adoption?
  - All direct single homed customers of SENSS ISPs are protected from direct floods and reflector attacks
  - 90% of direct multi homed or remote customers are protected from floods without signature and reflector attacks with just 1—3.8% of SENSS adoption
- SENSS comparison with existing cloud solutions
  - SENSS outperforms all after 0.4% of top transit deployment

Evaluation

- Conducted emulation and simulation over AS-level topology
- Used two strategy for SENSS server deployment
  - Top: SENSS is deployed in top *N* ASes ordered in decreasing customer size
  - Random: SENSS is randomly deployed in *N* Ases
- Two types of traffic
  - Uniform: Attack traffic are equally distributed among random ASes
  - Realistic: Attack traffic from only from residential network hosting Mirai botnet

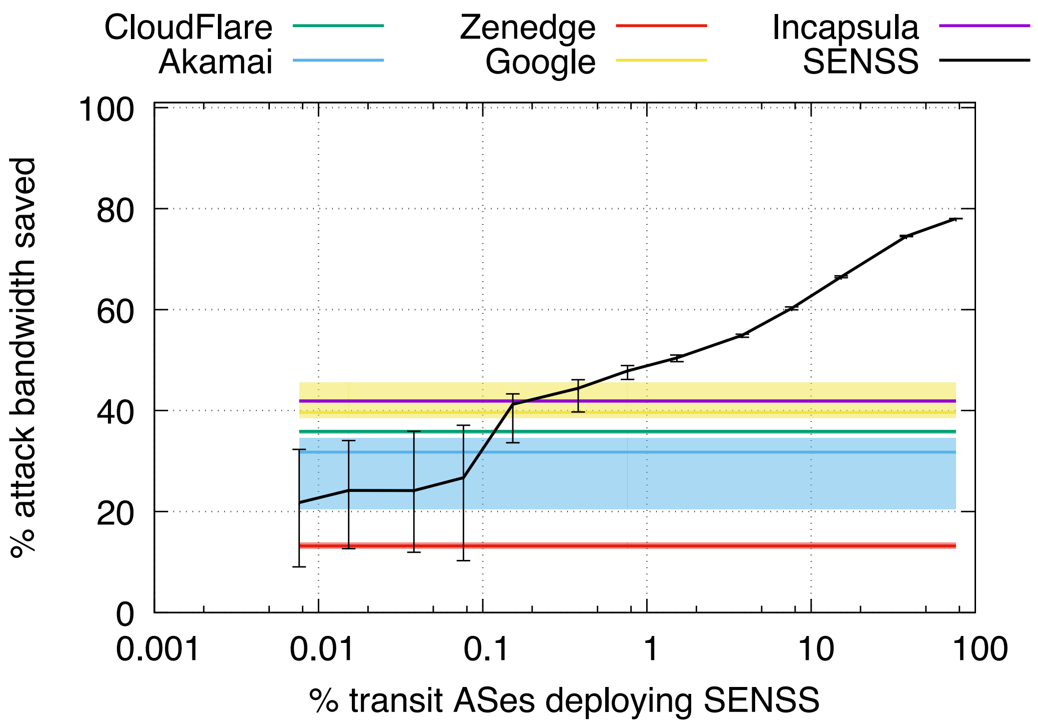
DDoS without signature

- SENSS is very effective in sparse deployment
- Deployment of 1.5% of top ASes achieves 90% for direct/single homed customer
- Deployment of 3.8% of top ASes achieves 90% of multi homed customers and remote customers



Comparison of SENSS with cloud deployments

- Estimate saved bandwidth by SENSS and cloud deployment strategies
- Saved bandwidth is the difference between bandwidth consumed with and without defense strategy
  - Ideal solution would have 100% saved bandwidth
- Existing solution save 13—46%
- For 10% deployment, SENSS saves 60% of bandwidth, 1.5—8 times more bandwidth than others



Overview

- Introduction
- SENSS
  - Architecture
  - SENSS API
- SENSS client programs
- Security and robustness
- Evaluation
- Conclusion

Conclusion

- SENSS is a collaborative defense where victims under volumetric DDoS attacks can request help from upstream ISPs
- SENSS API provides building blocks for clients to build custom defense to mitigate attacks
- SENSS servers are simple to deploy with monitory incentives to ISPs
- SENSS is effective in sparse deployment
- SENSS is more effective in saving bandwidth than other existing cloud based defense