FILTER-BASED APPROACH

StopIt

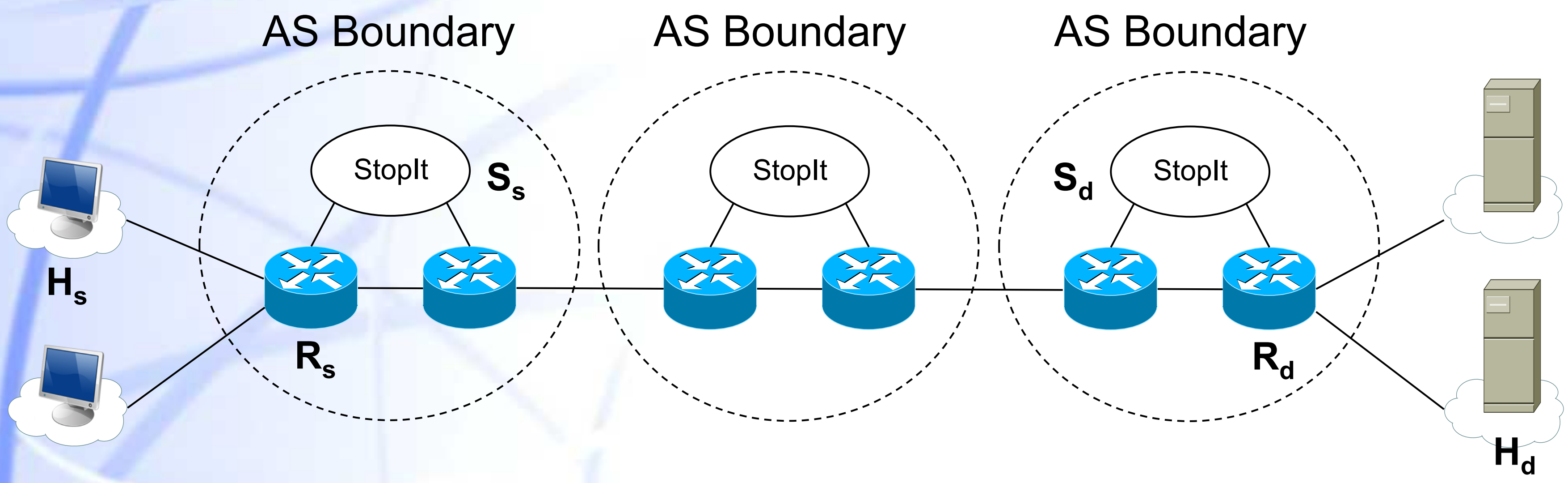
Introduction

- **Described in:**
  - **Liu, X., Yang, X., and Lu, Y. 2008. To filter or to authorize: network-layer DoS defense against multimillion-node botnets. *SIGCOMM Comput. Commun. Rev. 38, 4 (Oct. 2008), 195-206.***
- **Presents:**
  - **The design and implementation of a filter-based DoS defense system**
  - **A comparison study on the effectiveness of filters and capabilities**

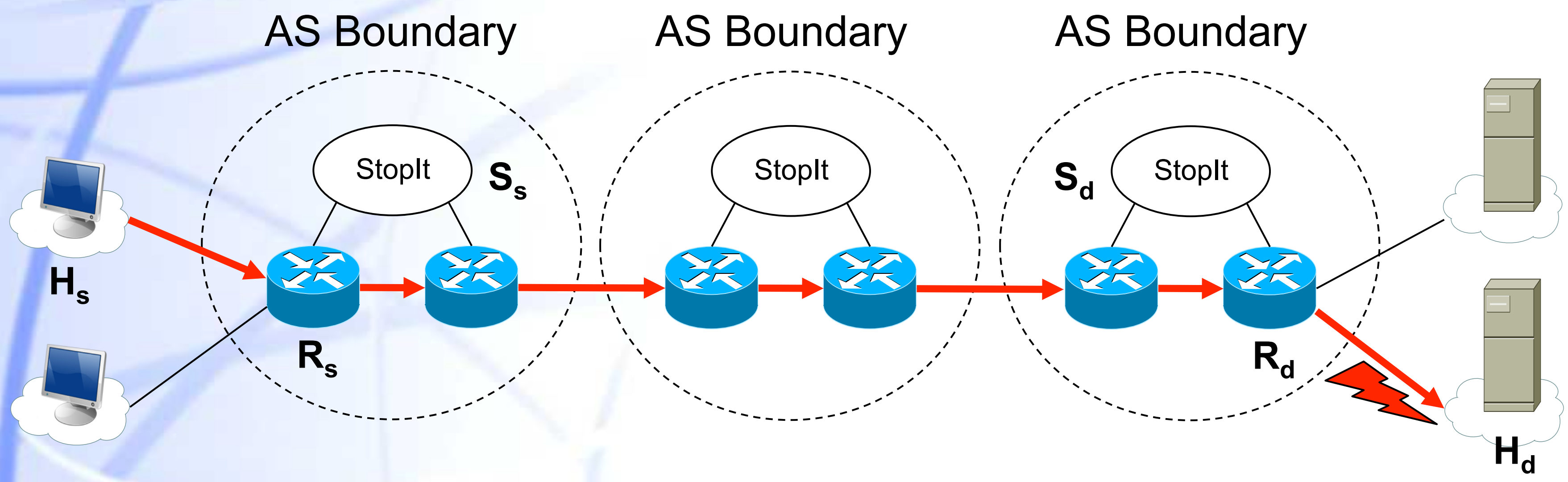
Motivation

- There is no consensus on how to build a DoS resistant network architecture
  - Capability-based approach
  - Filter-based approach
- Question: which one is a more effective DoS defense mechanism?
  - Procedure to answer: systematically compare filter-based and capability-based designs
    - ✓ Problem: not viable
    - ✓ StopIt enables a systematic comparison

StopIt overview: components

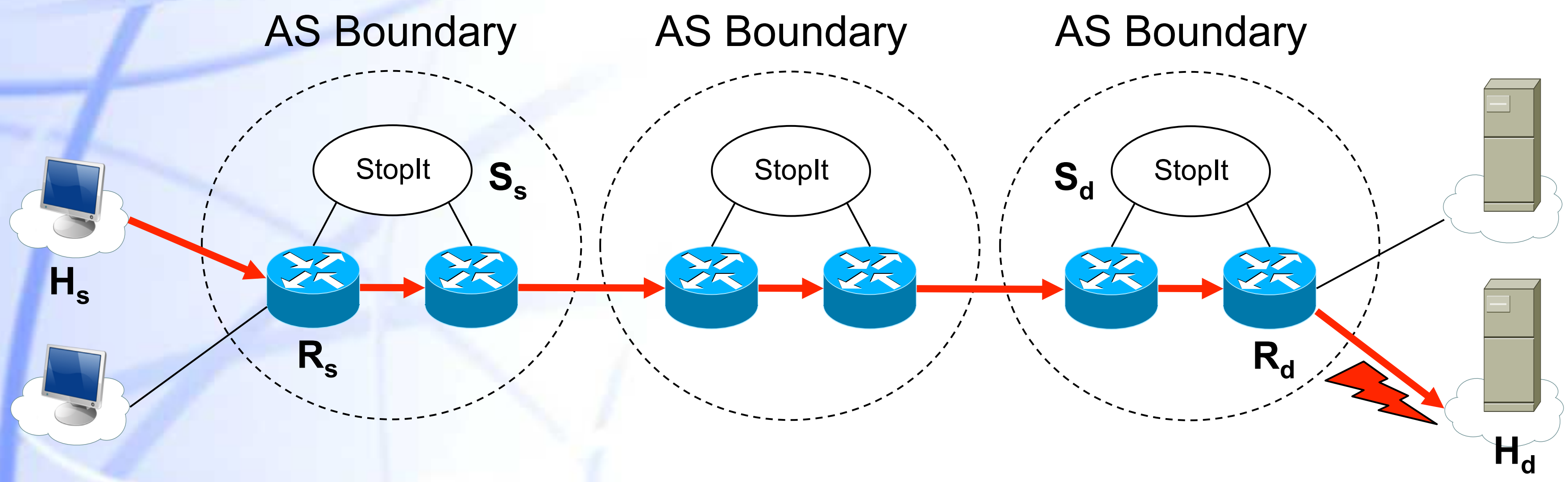


StopIt overview: components



- **When Hd detects attack traffic from Hs:**
  - **It invokes StopIt to block the attack flow during a period of time Tb**
  - **Attack flow is defined as (Hs, Hd)**

StopIt overview: components (II)

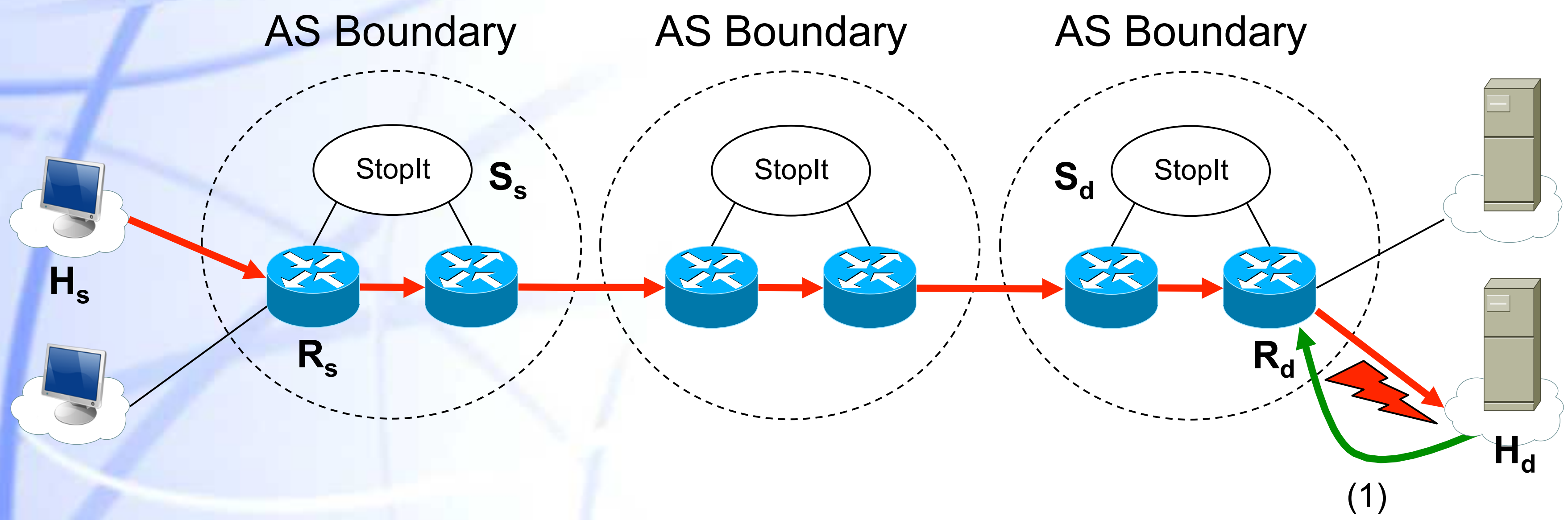


- **Each AS has a StopIt server:**
  - **Interdomain filter requests can only be sent between StopIt servers**
  - **Routers are configured with the address of its own StopIt server**

StopIt overview: components (III)

- **StopIt design uses BGP to publish StopIt server addresses**
  - **StopIt server address is encapsulated in optional and transitive BGP attribute**
- **A StopIt server gets BGP and IGP feeds from the routing system**
  - **BGP feeds → StopIt server addresses of other ASs**
  - **IGP feeds → addresses of routers in its own AS and the prefixes they originate**

StopIt overview: interactions (I)

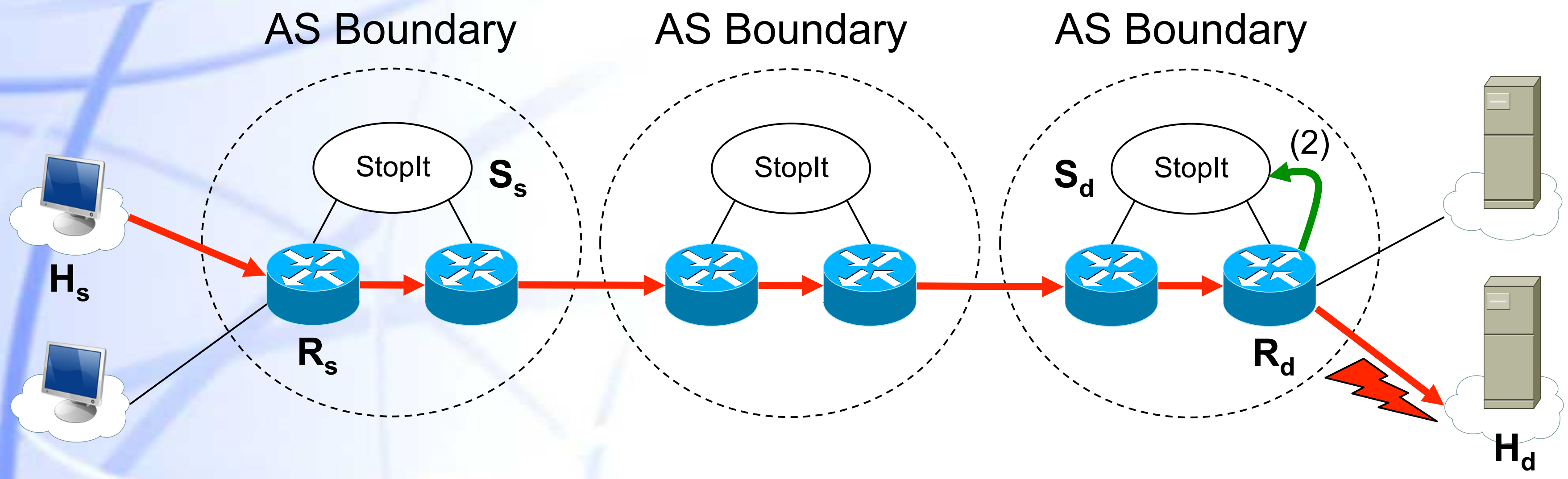


① **Hd sends a host-router StopIt request to Rd**

The request includes

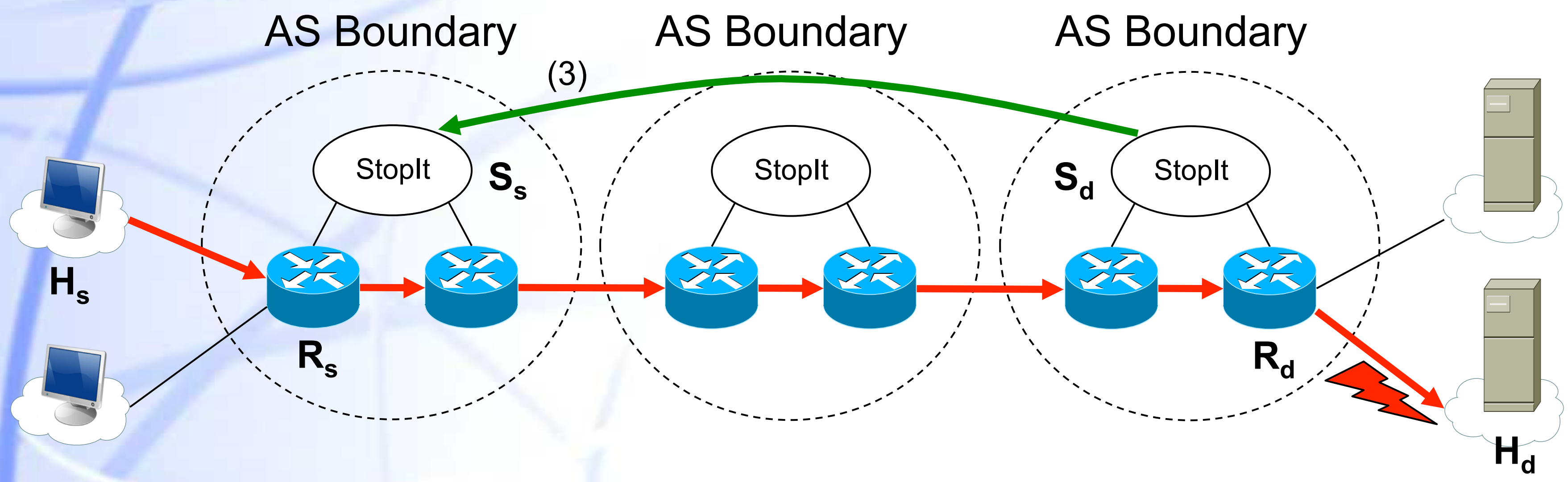
- Description of the attack flow (Hs, Hd), and
- a block period Tb

StopIt overview: interactions (II)



② **Rd verifies the request and sends a router-server StopIt request to Sd**

StopIt overview: interactions (III)

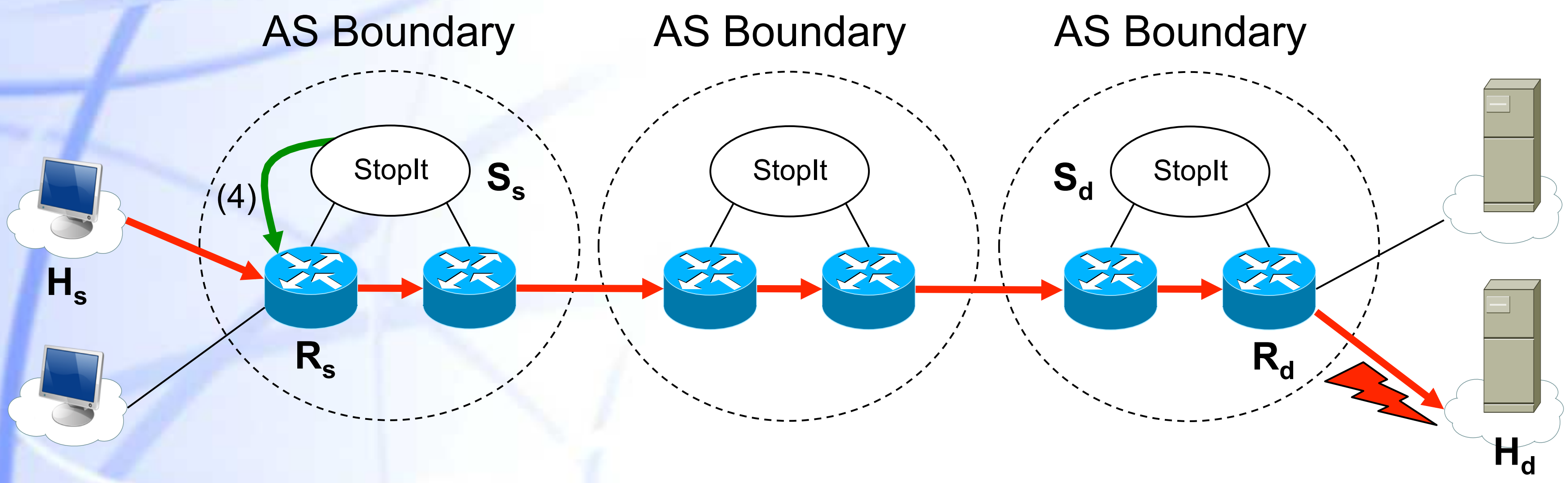


③ **Sd forwards an inter-domain StopIt request to Ss**

It includes:

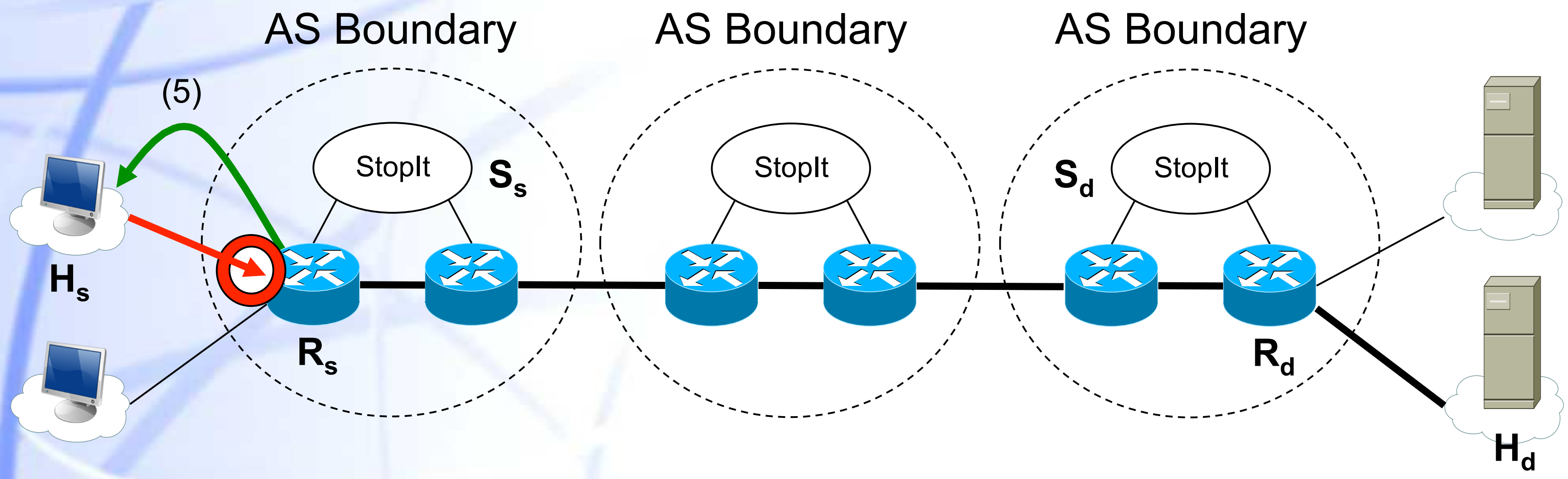
- (Hs, Hd)
- Tb

StopIt overview: interactions (IV)



④ **Ss locates Rs and sends a server-router request to the access router**

StopIt overview: interactions (V)



⑤ **Rs verifies the StopIt request, installs a filter and sends a router-host StopIt request to Hs**

Hs installs a local filter to stop sending to Hd

Secure StopIt: strategic attacks

- Source address spoofing attacks
- Resource exhaustion attacks
  - Flood filter requests to overload routers or StopIt servers
  - Send packet floods to cause filter requests to be discarded
  - Exhaust router filters
- Blocking legitimate traffic attacks

Systematic comparison

- **StopIt was compared, using NS-2, with:**
  - **Capability-based solutions: TVA, Portcullis**
  - **Filter-based systems: AITF, Pushback**
- **Simulation results:**
  - **StopIt outperforms AITF and Pushback**
  - **StoptIt does not always outperform a capability-based system**

Conclusion

- Filter and capabilities are viable choices to build a DoS-resistant network architecture
- Neither is more effective that the other in all types of attacks
- A DoS-resistant network architecture is likely to incorporate multiple mechanisms

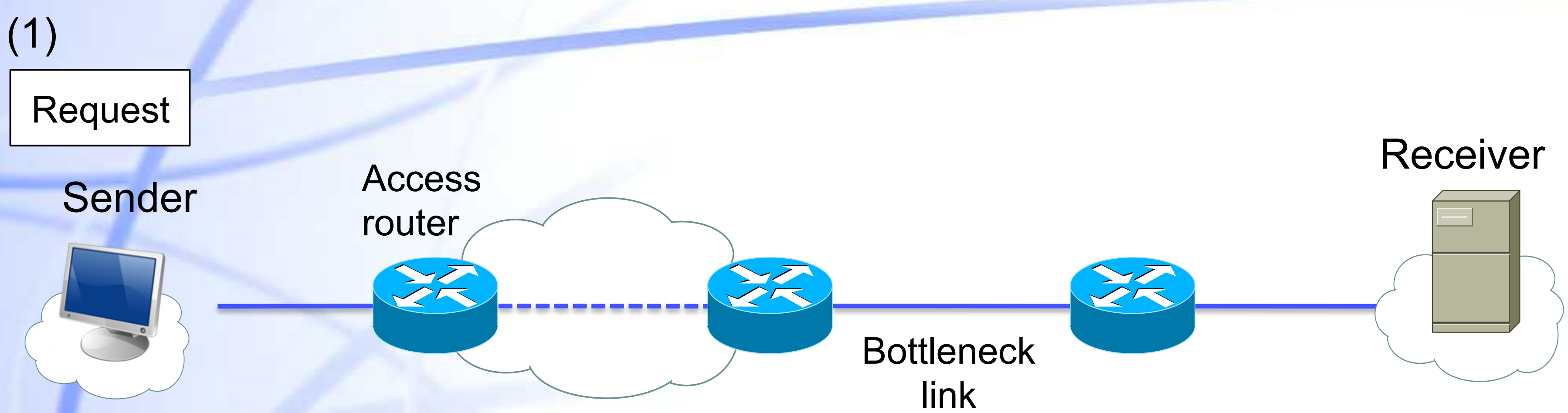
COLLUDING ATTACKERS

NetFence

Introduction

- **Described in:**
  - Xin Liu, Xiaowei Yang, and Yong Xia. NetFence: preventing internet denial of service from inside out. In *Proceedings of the ACM SIGCOMM 2010). ACM, New York, NY, USA, 255-266.*
- **Motivation:**
  - Colluding attackers introduces scalability problems in capability and filter solutions
- **NetFence:**
  - Probably guarantees each sender a fair share of a bottleneck capacity
  - Does not keep per-host state at bottleneck routers
  - Places the network at the first line of DoS defense
  - Enables DoS victims to suppress unwanted traffic following a capability-based approach

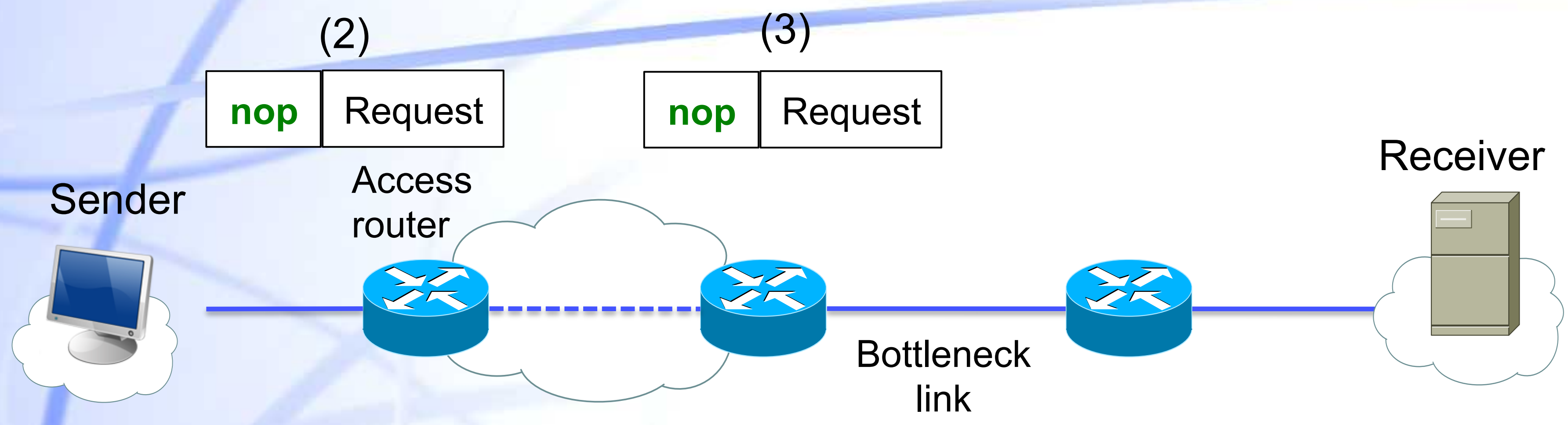
System overview



- NetFence is based on <u>unforgeable feedback</u> and <u>policing functions</u> included at bottlenecks and access routers

① A NetFence sender starts an end-to-end communication by sending request packets to the NetFence receiver

System overview



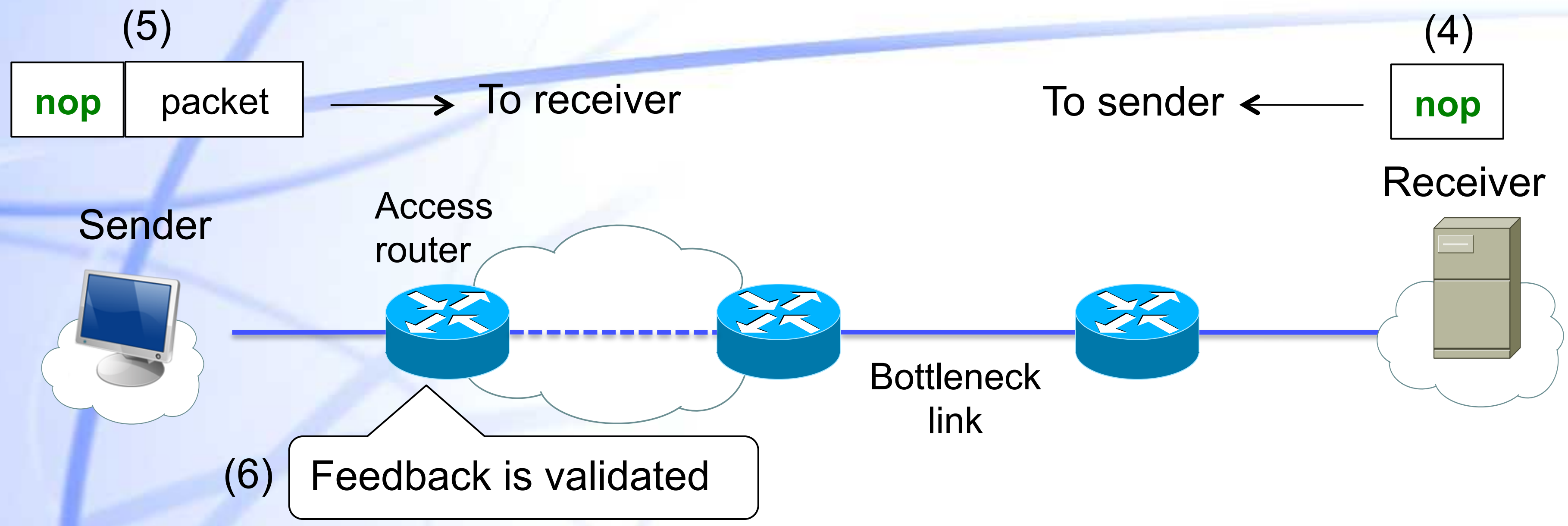
② **The access router inserts a "*nop*" feedback in the NetFence header of the packet**

- ✓ **"*nop*" indicates that no policing action is needed**

③ **A bottleneck router on the path might modify the feedback**

- ✓ **Similarly to TCP ECN**

System overview



④ **The receiver returns the feedback to the sender**

  ✓ **E.g. TCP can piggyback the feedback in data packets**

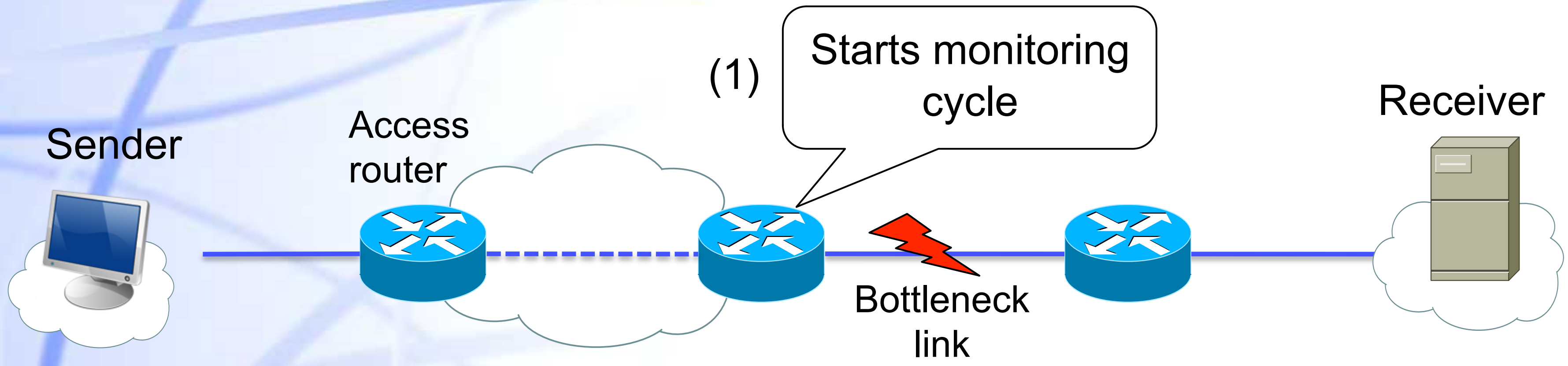
⑤ **The sender can send regular packets containing the feedback**

⑥ **The feedback is some kind of "capability" that is validated by the access router**

Protecting the request channel

- The request channel is limited to 5% of any link capacity
  - Similarly to TVA
- NetFence combines packet prioritization and priority-based rate limiting
  - A sender can assign different priority levels to request packets
  - Routers send level-k packets with higher priority than lower-level packets
  - But sender is limited to send level-k packets at half of the rate of level-(k-1) packets
    - ✓ Enforced at access routers

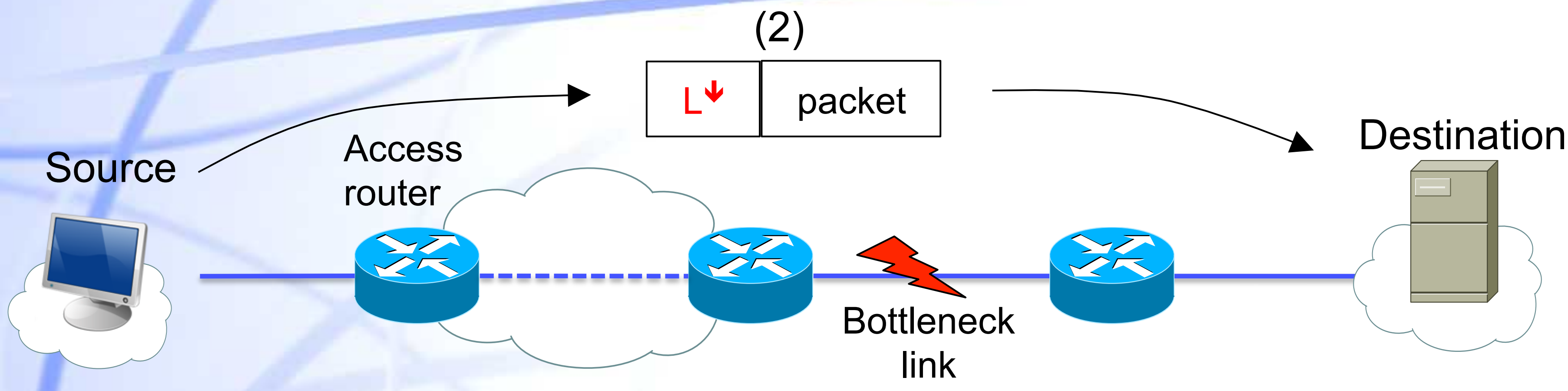
Protecting the regular channel



- A NetFence router periodically verifies if each output link is under attack
  - ✓ Based on a combination of utilization and loss rate of regular packets

① If an attack is detected, the router starts a <u>monitoring cycle</u>

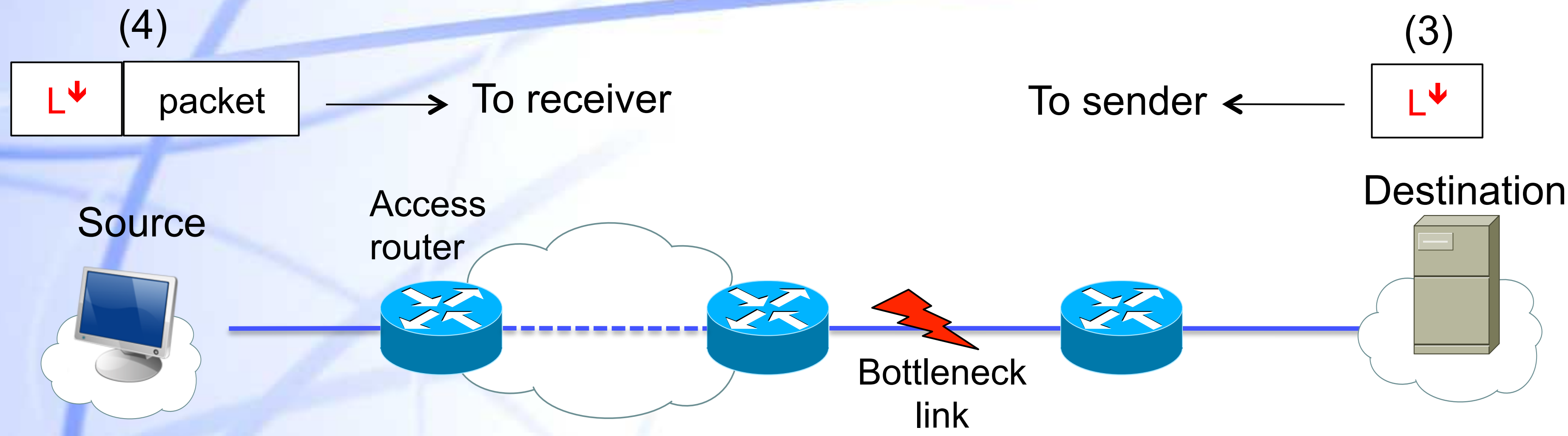
Protecting the regular channel



② **During a monitoring cycle:**

- ✓ **<u>While bottleneck link L is overloaded</u>, any request/ regular packet traversing L is stamped the L↓ feedback**
- ✓ **L↓ indicates that link L is overloaded and the access router should reduce the traffic traversing L**

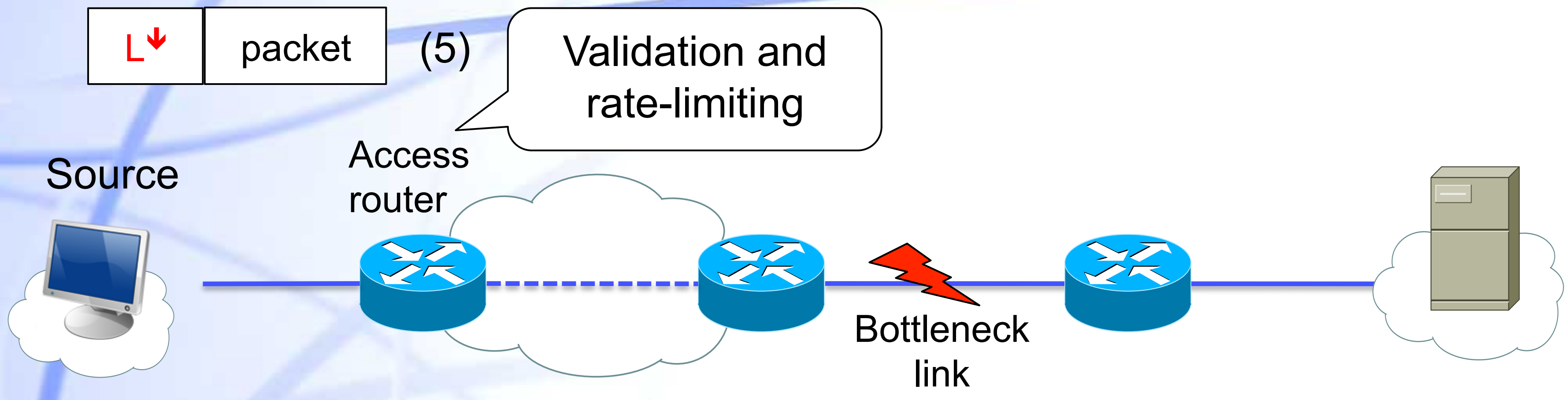
Protecting the regular channel



③ **Receiver returns L↓ feedback to sender**

④ **Sender includes L↓ feedback in regular packets sent towards the receiver**

Protecting the regular channel

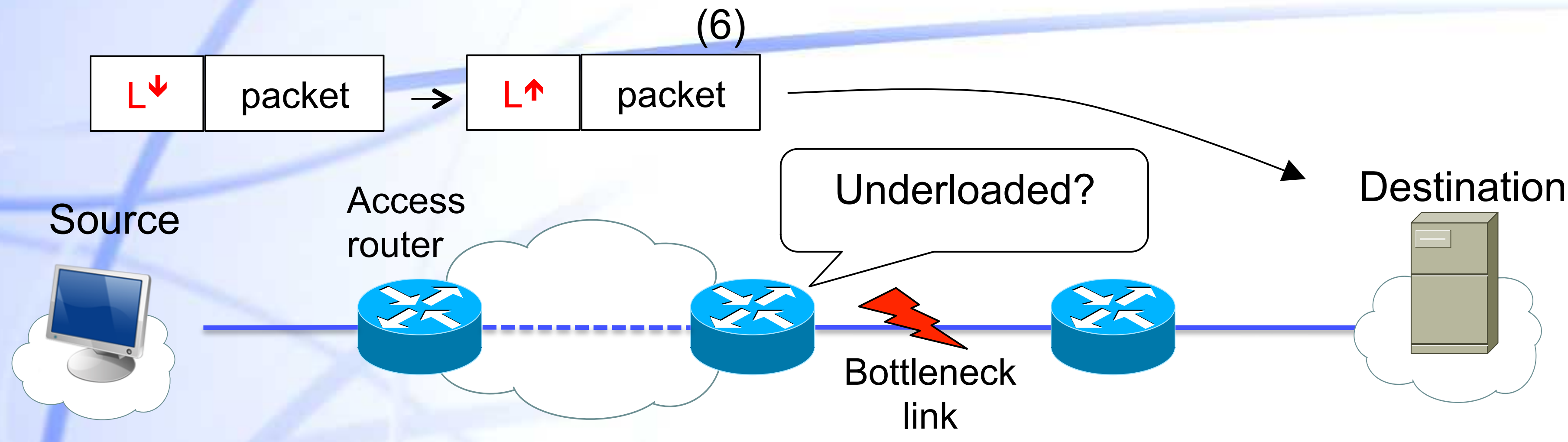


⑤ **The access router validates L↓ feedback**

It maintains one rate limiter for every pair sender-bottleneck

- ✓ **A packet from sender *src* carrying L↓ feedback must pass the rate limiter {*src*, L↓}**

Protecting the regular channel



⑥ **When the access router forwards the packet it resets the feedback to L↑**

- ✓ **L↑ indicates that link L is underloaded and access router can allow more traffic traversing L**

⑦ **The bottleneck router stamps L↓ feedback until the bottleneck gets underloaded**

Protecting the regular channel

- **The access router dynamically adjusts rate-limit of limiter {src, L}:**
  - **Additive Increase and Multiplicative Decrease (AIMD) algorithm is used**
    - ✓ **L↓ decreases the rate limit multiplicatively**
    - ✓ **L↑ increases the rate-limit additively**
  - **AIMD converges onto efficiency and fairness**
    - ✓ **Each legitimate client obtains its fair share of the bottleneck capacity:**

$$\frac{V_g \cdot p \cdot C}{G + B}$$

C: bottleneck link capacity
G: number of legitimate senders
B: number of malicious senders