CPS 590.5 Computer Security Lecture 12: Preventing Internet Denial of Service

Xiaowei Yang
xwy@cs.duke.edu

Roadmap

- Previous lecture
  - Prevent IP address Spoofing
- Today
  - Capability-based DDoS defense
  - Filter-based DDoS defense

TVA: a DoS-limiting Network Architecture

Xiaowei Yang

Tom Anderson

David Wetherall

Desired Properties of Information System Security

- Confidentiality and integrity
  - Protection against unauthorized access to or modification of information
- Resource Availability
  - Protection against the *denial of service* to legitimate users
  - TVA protects against network DoS attacks.

Attacks and Defenses on Confidentiality and Integrity

- Attackers may eavesdrop or modify conversations or actively break into a system.
- Defenses can usually be implemented end-to-end
  - **Cryptographic techniques**
    - **TLS, SSH, IPSec**
  - **Host security**

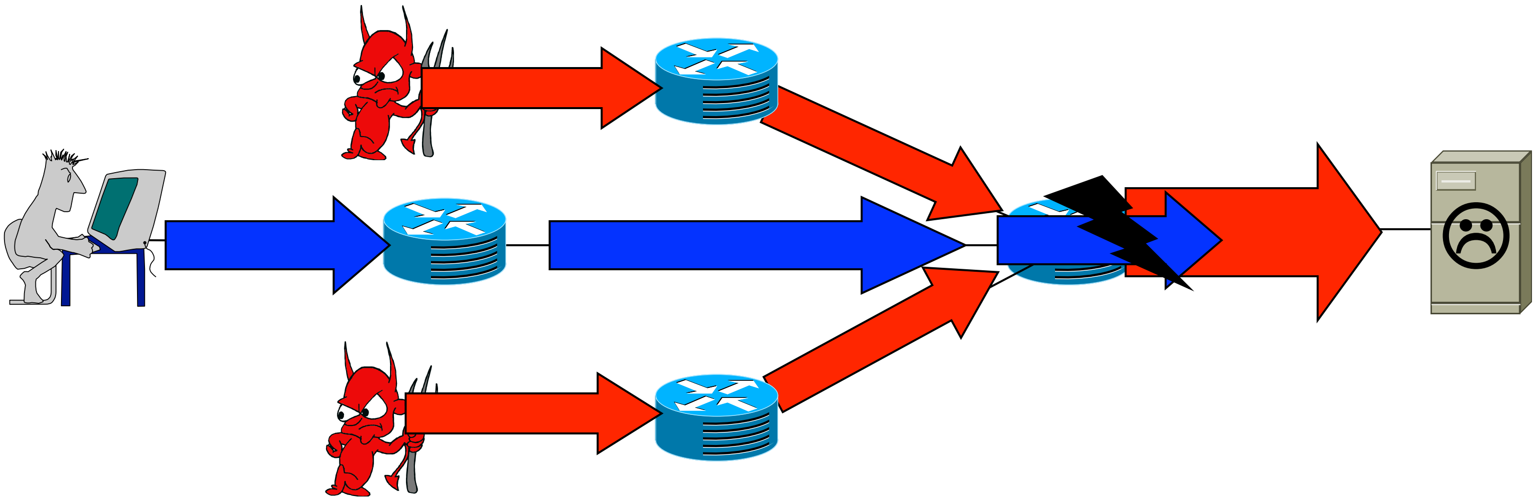
Denial of Service (DoS) Attacks

- Exhausting shared resources
  - Bandwidth, memory, or CPU time
- Disrupting configuration information
  - Routing
- Disrupting physical network components
  - Cutting off a fiber

Extortion via DDoS on the rise

- Egotism or vandalism
- Cyber crime
  - In March, 2004, a sustained campaign of DoS attacks was launched against Britain's top 20 betting sites. (BBC News)
  - "A Massachusetts businessman allegedly paid members of the computer underground to launch organized, crippling distributed denial of service (DDoS) attacks against three of his competitors, in what federal officials are calling the first criminal case to arise from a DDoS-for-hire scheme ." (http://securityfocus.com/news/9411/, August, 2004)
  - "Criminals are increasingly targeting corporations with distributed denial-of-service attacks designed not to disrupt business networks but to extort thousands of dollars from the companies." (http://www.networkworld.com/news/2005/051605-ddos-extortion.html, May, 2005)

How to Attack : Exhausting shared resources

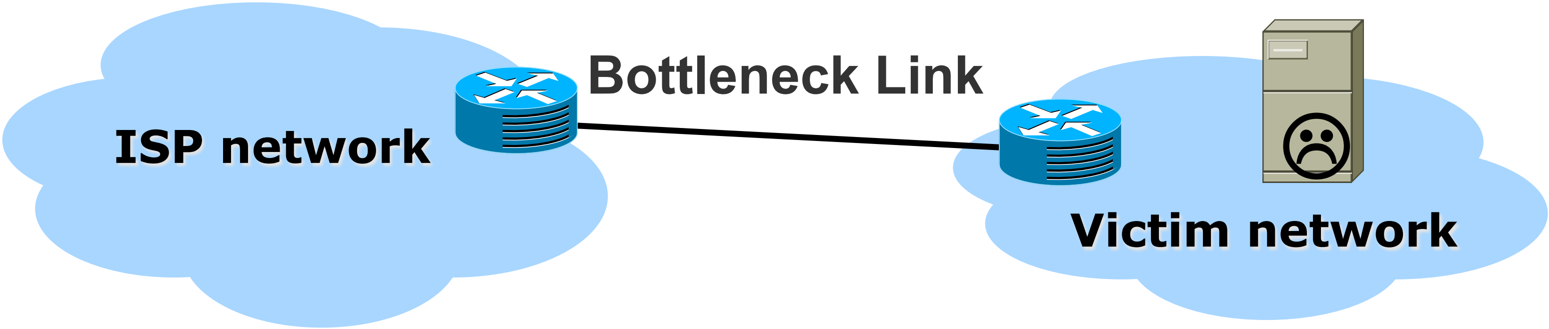


- Flooding traffic to exhaust the bandwidth, memory, or CPU of a victim
  - Spoof addresses to hide
  - Distributed DoS (DDoS) to hide and to maximize damage
    - Multiple (weak) machines against (strong) victim

More flooding attacks

- Memory attack
  - TCP SYN flooding
- CPU attack
  - Unnecessary computation, e.g., TLS attack
- Easier to mitigate
  - End-to-end mechanisms
    - cookies, puzzles

Defending against bandwidth attacks is hard



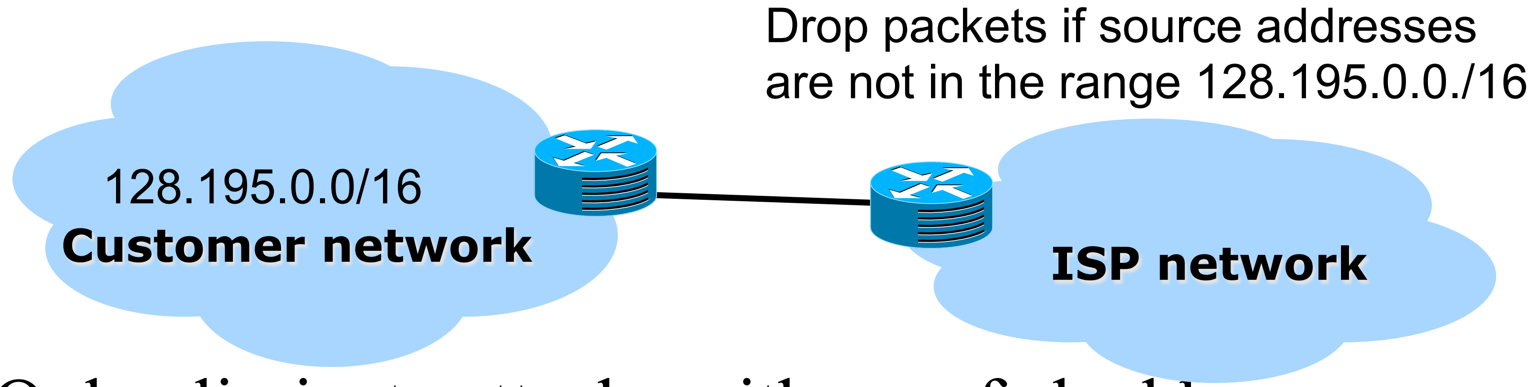
- Effective defense requires packets drop before the bottleneck
  - ISPs must drop flood packets before they reach the victim networks
  - But only destinations know what packets are desired
  - Anti-distributed DoS services cost around $12,000 per month from carriers such as AT&T and MCI

Existing Defense Mechanisms

- Preventive
  - Goal is to make attacks impossible by design
  - Address validation, overlay filtering, overlay distribution
- Incomplete attack coverage

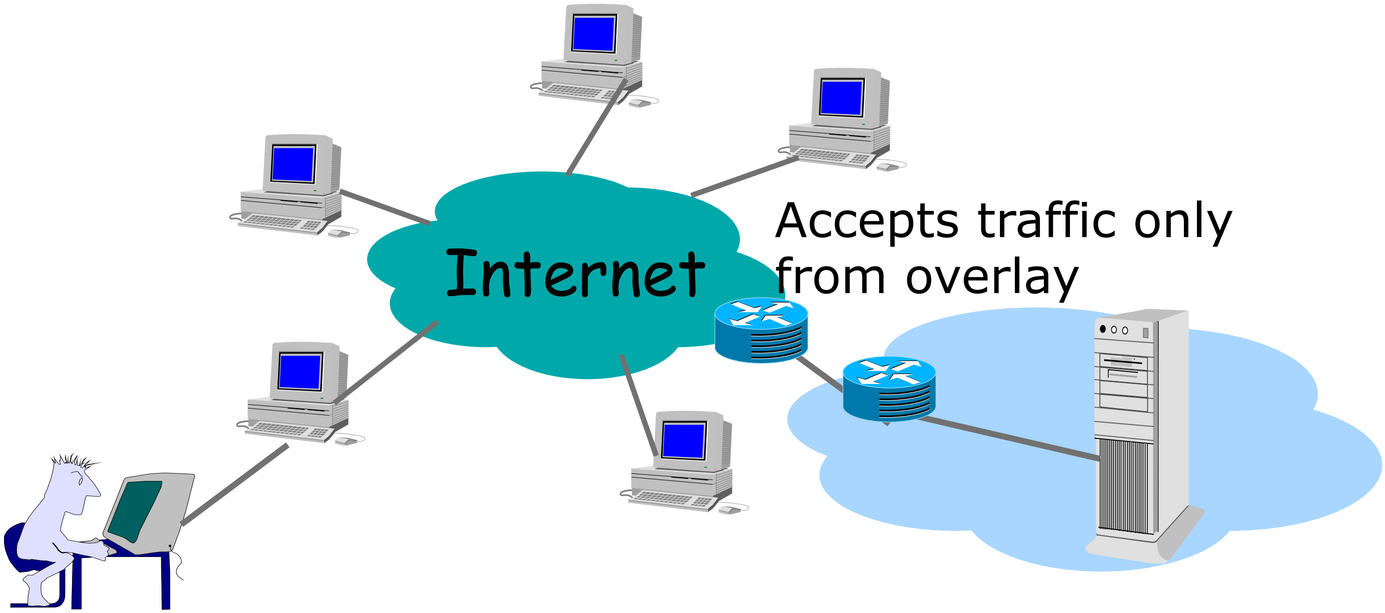
- Reactive
  - Goal is to contain the damage of attacks after they are detected.
  - Traceback, pushback, active filtering, intrusion detection
- Postmortem and imprecise

Preventive: Address validation is insufficient


Drop packets if source addresses are not in the range 128.195.0.0./16
128.195.0.0/16
Customer network
ISP network

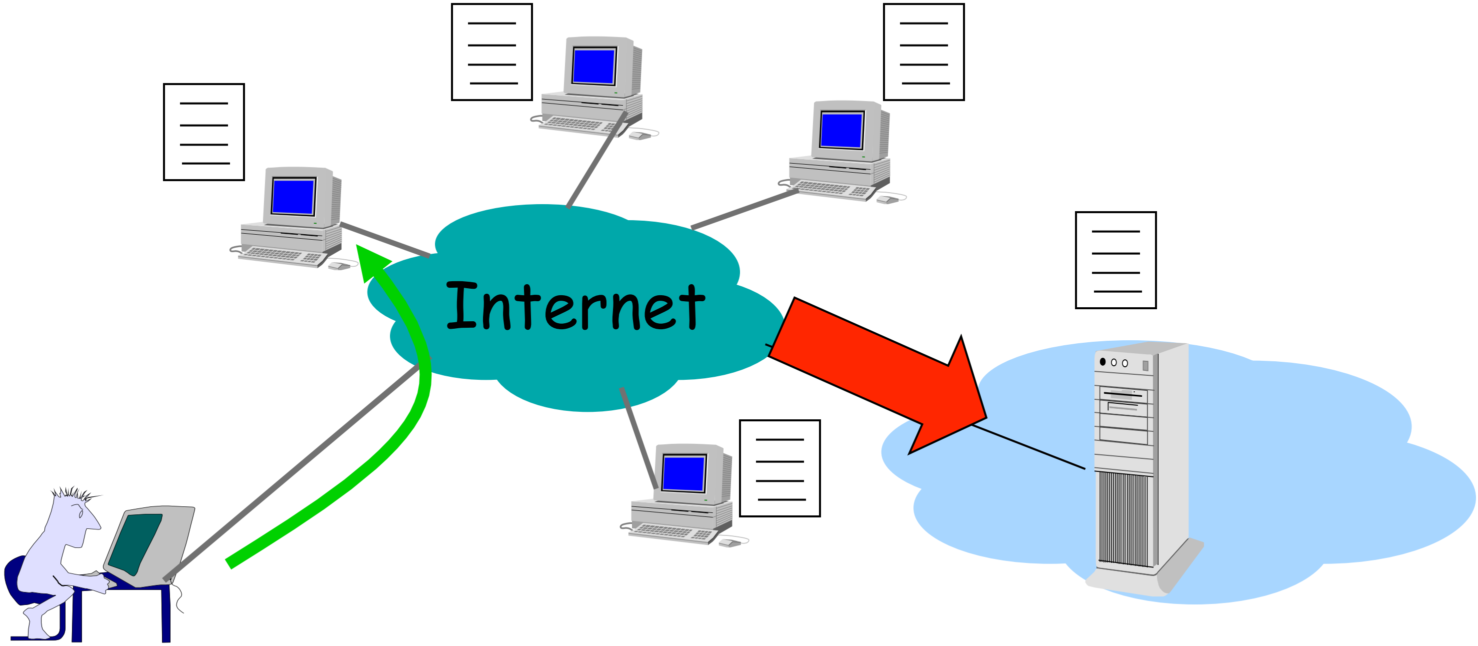
- Only eliminate attacks with spoofed addresses
- One spoofable network allows attacks to happen
  - Nearly 25% spoofable networks [Beverly05]
- RFC 2827, RFC 3704

Preventive: Overlay filtering does not protect open communication



- Overlay nodes apply filters and authenticate clients.
- They need to know destination policies.
- [SOS], [Mayday]

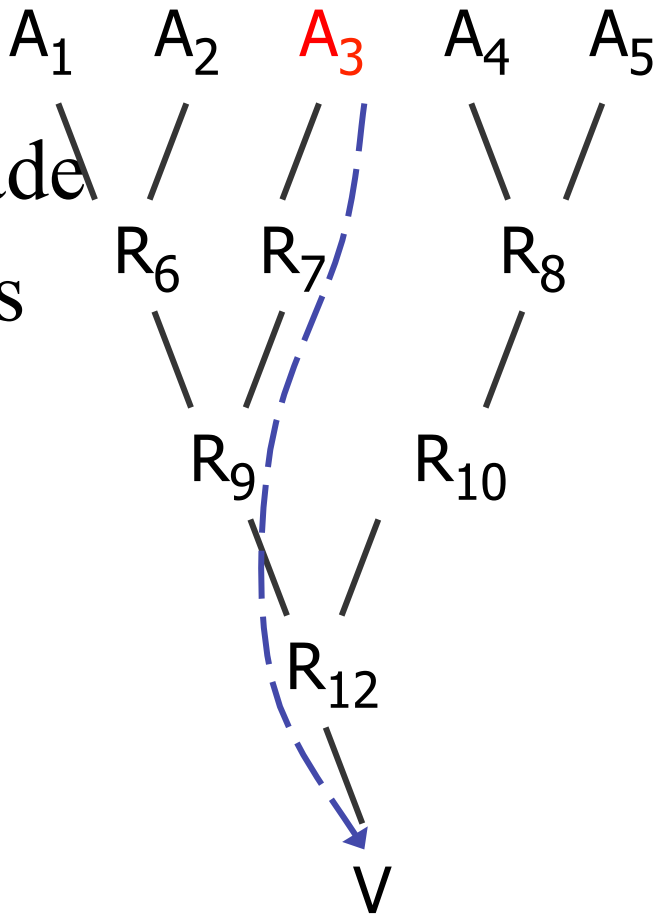
Preventive: Overlay distribution does not help all applications.



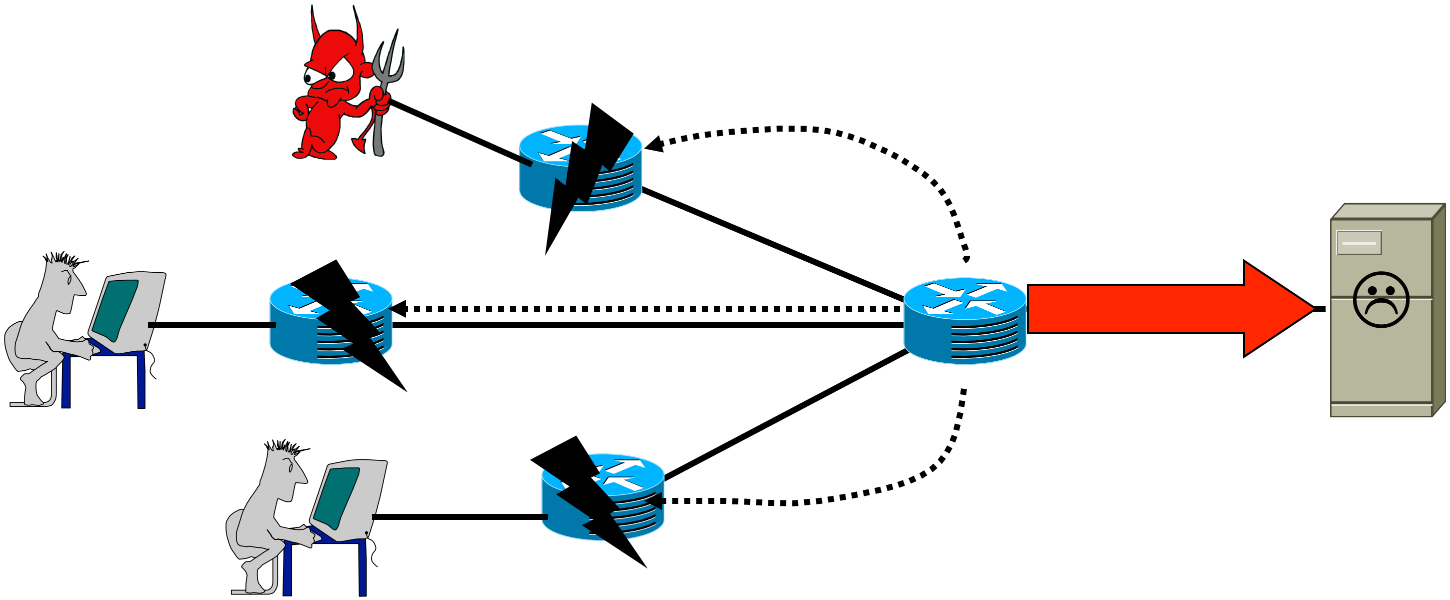
- Dynamic contents:
  - real time communication, database queries, transaction
- Akamai, Coral, etc.

Reactive: IP traceback is too little too late

- Damage has already been made
- Needs additional mechanisms to stop the attacks
- [Bellovin00], [Savage00], [Snoeren01], [Yarr03] …

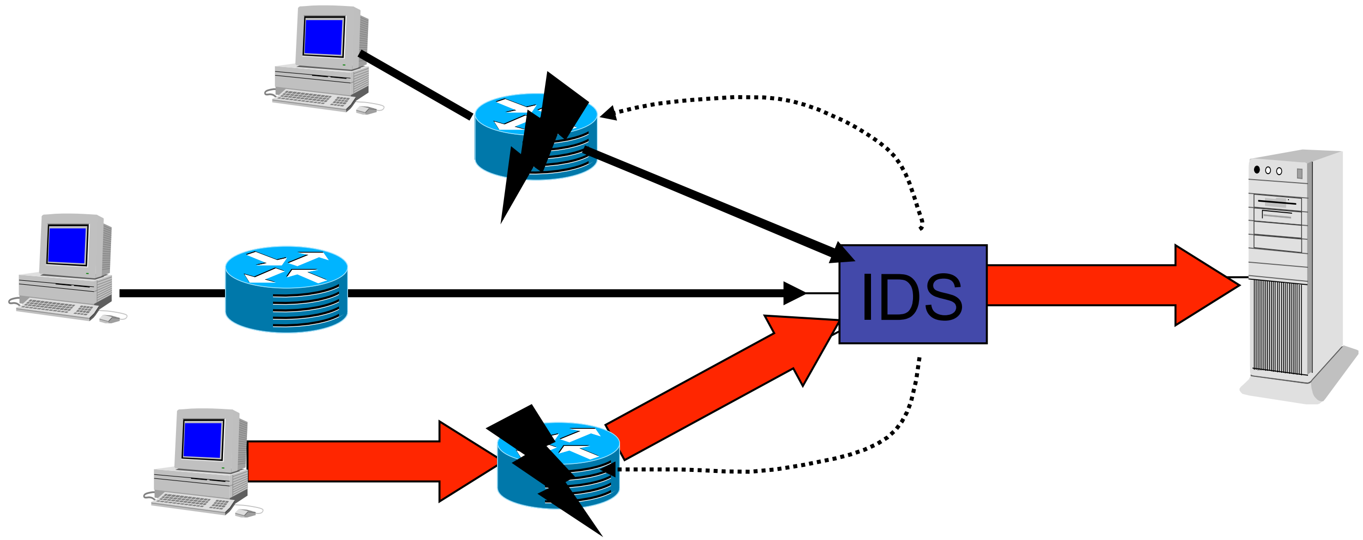


Reactive: pushback lacks discrimination



- Attacks may forge arbitrary packets.
- Both legitimate sources and attackers suffer.
- Network controlled pushback
  - Volume based filtering is ineffective in large-scale DDoS attacks.
  - [Mahajan01], [Ioannidis02]
- Host controlled pushback
  - Path-based filtering affects hosts that share the same domain-level path.
  - [Argyraki05]

Reactive: Intrusion detection system (IDS) threatens openness



- Detects anomaly from known traffic patterns and packet signatures
- Automated IDS alarms on unusual traffic
  - Clamps down on attacks and new applications alike
  - In the limit this leads to a closed system without innovation.

Our approach

Free our mind for the moment: a clean architectural solution

- Can we design a network that is free of DDoS attacks, yet supports open communication?
- The answer is NO.
  - Ex: a million hosts send one packet to a host behind a DSL link.
- Open communication and DoS-free are conflicting goals.
- Goal: design an open and resilient network
  - Open
    - Allow any two hosts to communicate
  - Resilient
    - Performance degrades gracefully as the number of attacking hosts increases
    - Minimizes the damage caused by the attacking hosts

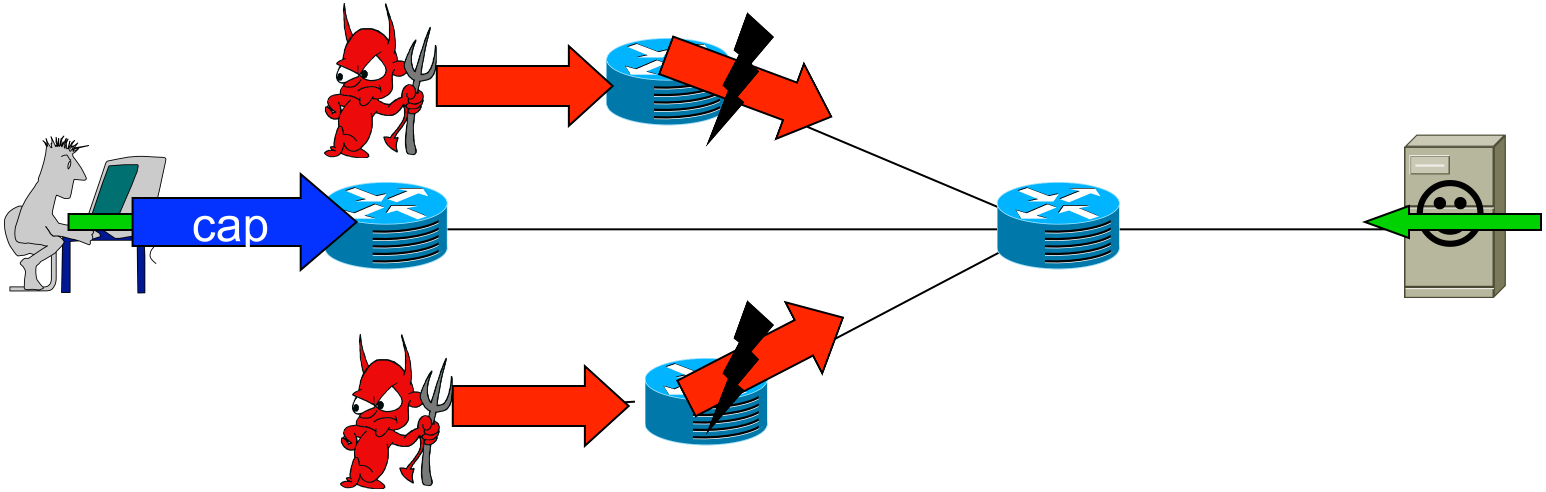
The Need for Capabilities

- Goal: design an open and resilient network
  - Open
    - Allow any two hosts to communicate
  - Resilient
    - Performance degrades gracefully as the number of attacking hosts increases
    - Minimizes the damage caused by the attacking hosts

- Observe that:
  - Only the network can shed load before it is excessive
  - Only destinations know which packets are desired

- End result:
  - Network filtering must be based on destination control
    - Destination tells the network what packets are desired
  - Authorization needs to be explicit so it can be checked throughout the network, i.e., packets carry capabilities
    - A preventive mechanism minimizes damage.

Sketch of the capability approach



1. Source requests permission to send.
2. Destination authorizes source for a limited transfer, e.g, 32KB in 10 secs
   - A capability is the proof of a destination's authorization.
3. Source places capabilities on packets and sends them.
4. Network filters packets based on capabilities.

- Anderson et al. [Anderson03], Yarr et al. [Yarr04]

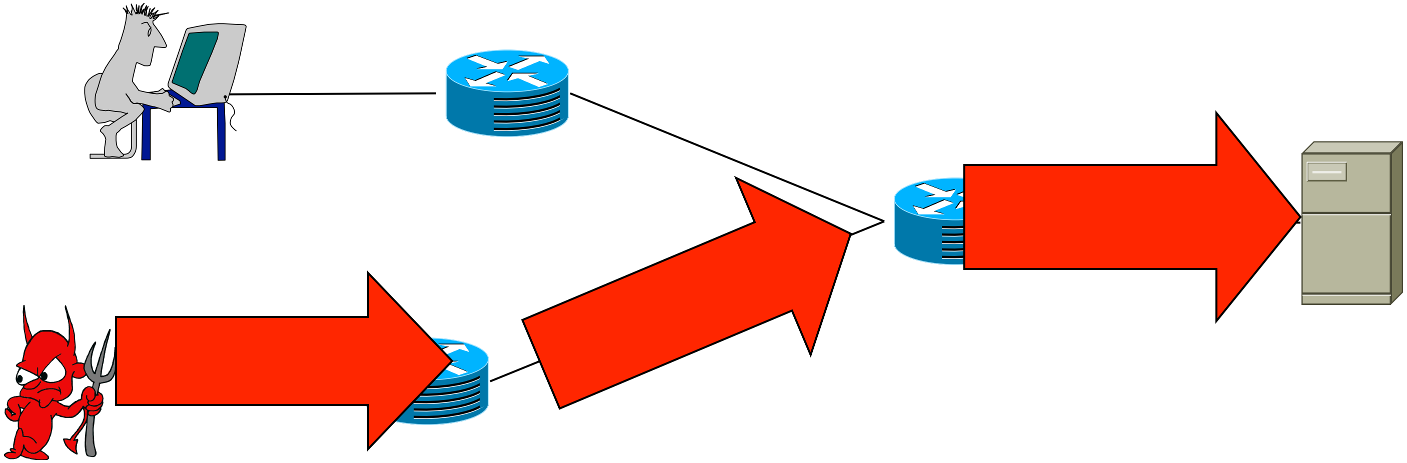
The capability approach can be attacked.

- Problems
  1. Request packet floods
  2. Authorized packet floods
  3. Added functionality in a router’s
  4. Authorization policies
- TVA addresses all of the above.

Challenges

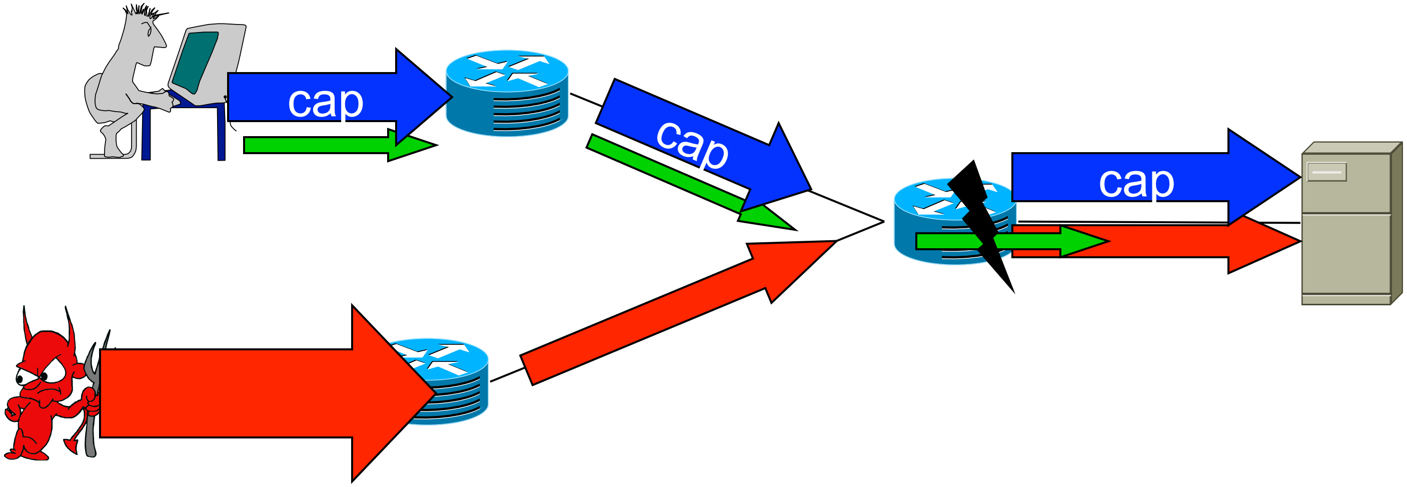
1. Counter a broad range of attacks, including request and authorized packet floods
2. Router processing with bounded state and computation
3. Effective authorization policies

Request packet floods



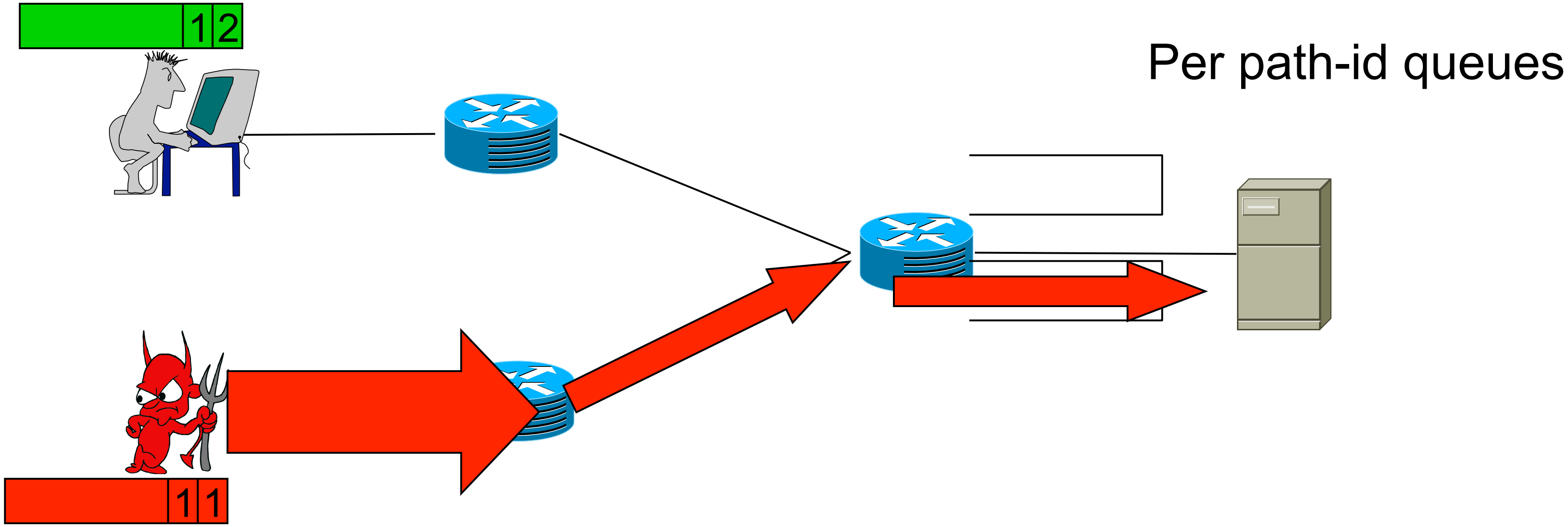
- Request packets bootstrap communication and do not carry capabilities.

Counter request packet floods (I)



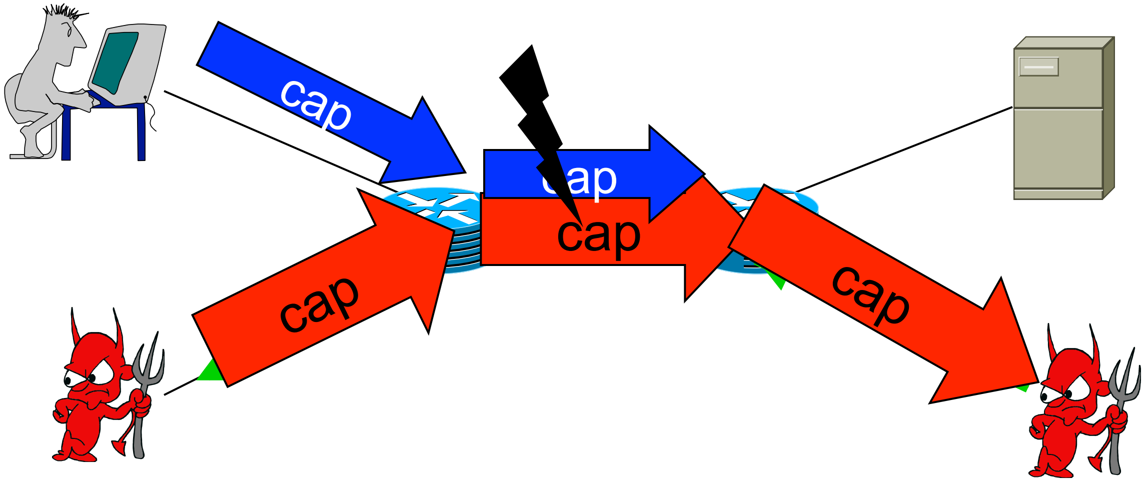
- Rate-limit request packets

Counter request packet floods (II)

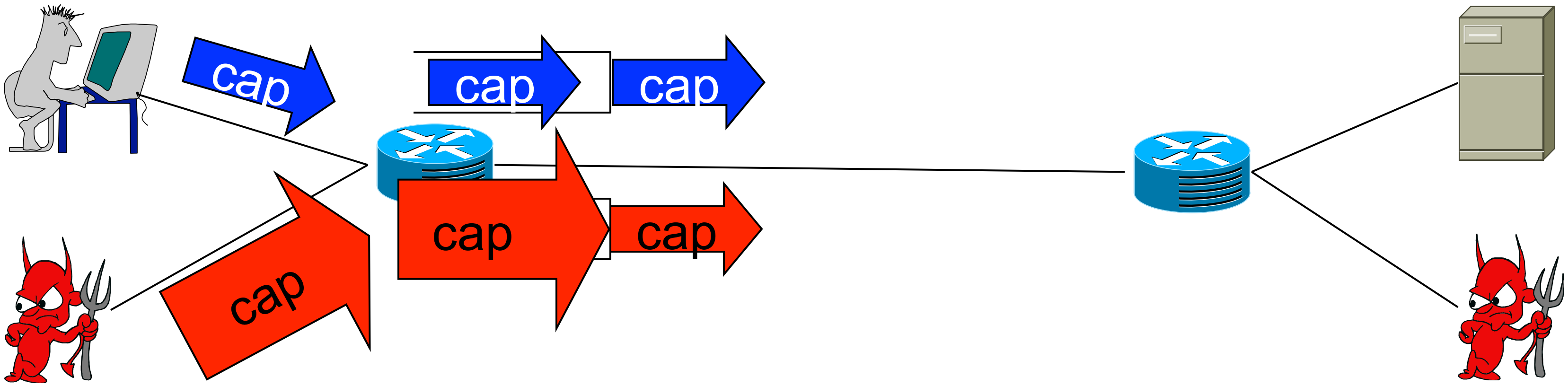


- Rate-limit request packets
- Routers insert path identifier tags [Yarr03].
- Fair queue requests using the most recent tags
  - Localize the damage of attacks
  - Damage can be further reduced with volume-based pushback

Authorized packet floods



Counter authorized packet floods

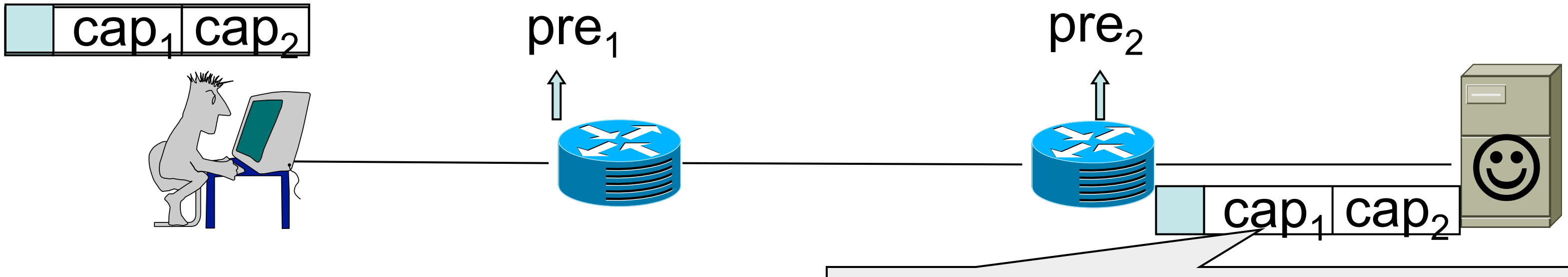


- Per-destination queues
- TVA bounds the number of queues.

Challenges

1. Counter a broad range of attacks, including request packet floods and authorized packet floods
2. Router processing with bounded state and computation
3. Effective authorization policies

TVA's implementation of capabilities

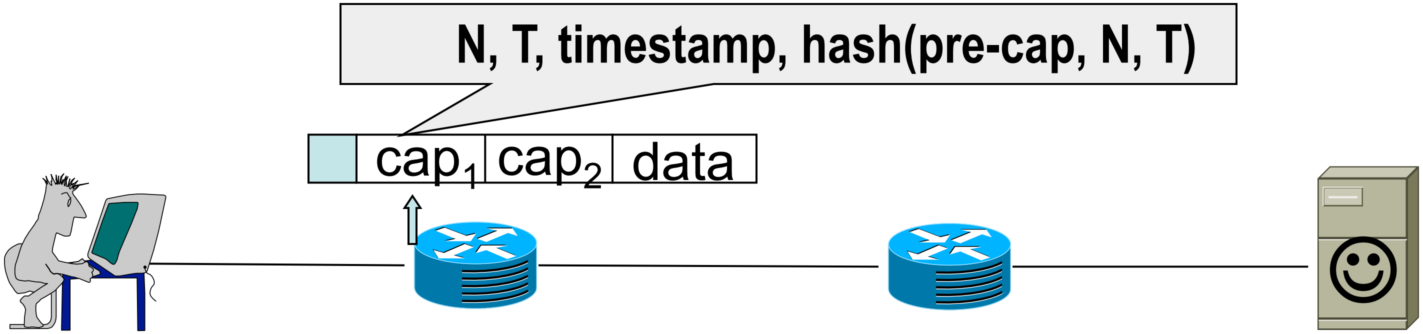


N, T, timestamp, hash(pre_1, N, T)

- Requirements
  - Unforgeable, must expire, fine-grained, and efficient

1. Routers stamp pre-capabilities on request packets
   - **(timestamp, hash(src, dst, key, timestamp)**
2. Destinations return fine-grained capabilities.
   - **(N, T, timestamp, hash(pre-cap, N, T))**
   - send N bytes in the next T seconds, e.g. 32KB in 10 seconds
   - Return paths need not be symmetric.
   - Capabilities can be renewed inline

Validating fine-grained capabilities

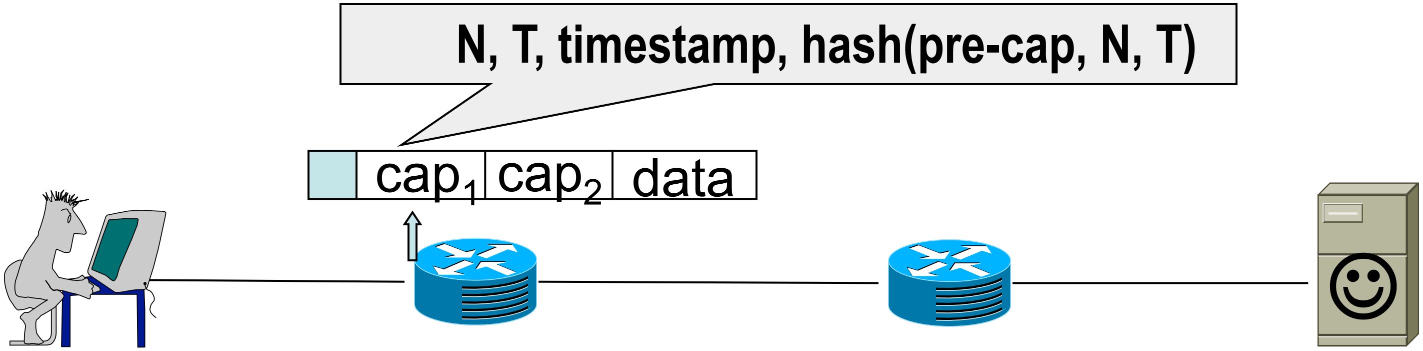


1. A router verifies that the hash value is correct.
2. Checks for expiration: $\mathit{timestamp} + T \cdot \mathit{now}$
3. Checks for byte bound: $\mathit{sent} + \mathit{pkt\_len} \cdot N$
   - Requires state

Bounded computation

- The main computation overhead is hash validation.
- On a Pentium Xeon 3.2GHz PC
  - Stamping pre-capabilities takes 460ns
  - Validating capabilities takes 1486ns

Bounded state



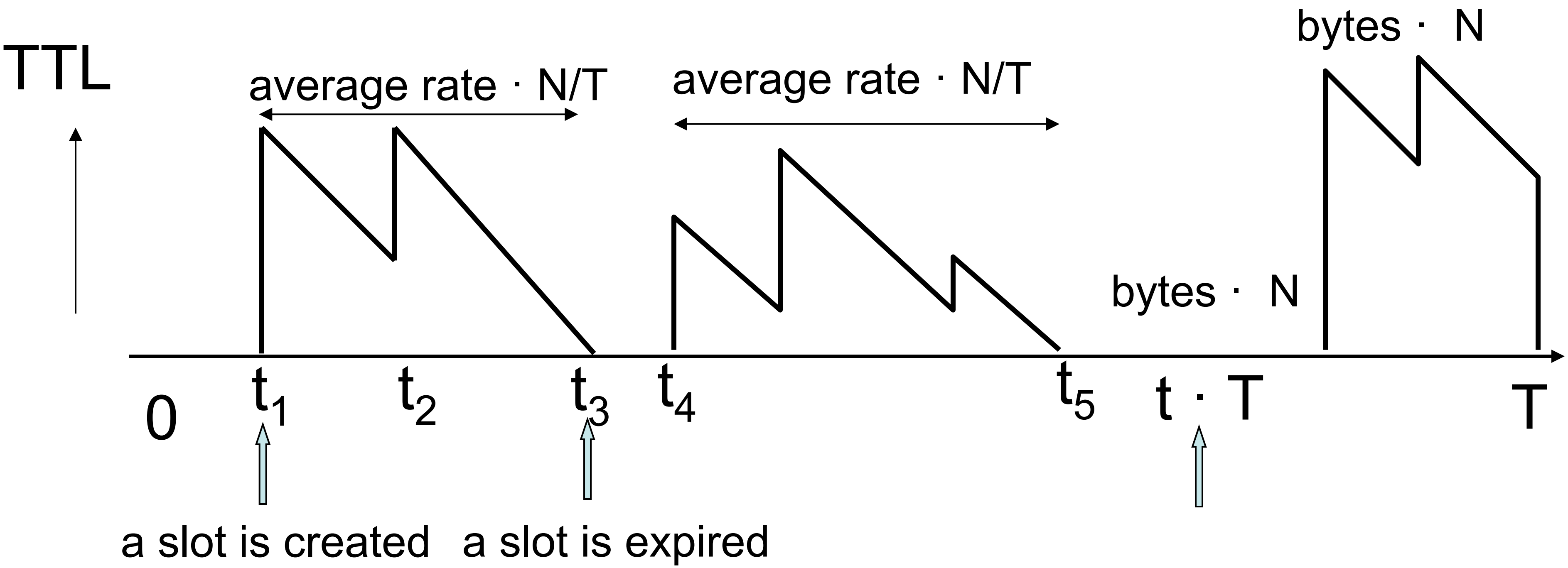
N, T, timestamp, hash(pre-cap, N, T)

cap_1 cap_2 data

sent + pkt_len · N

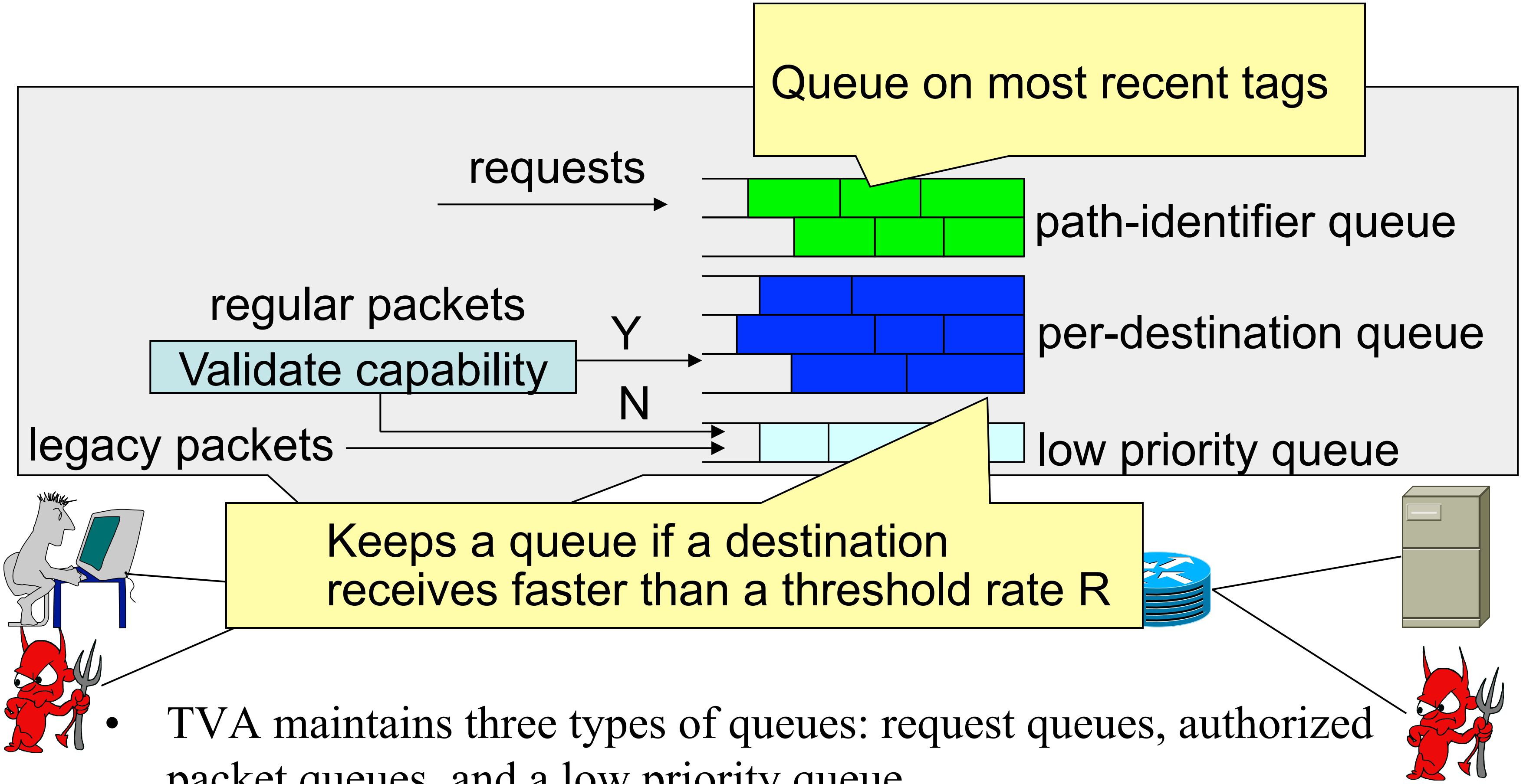
- Create a slot if a capability sends faster than N/T.
- For a link with a fixed capacity C, there are at most C/(N/T) flows
- → Number of slots is bounded by C / (N/T)

Worst case byte bound is 2N in T seconds



- If a slot expires, it indicates that a capability sends slower than N/T.

Bounded number of queues



- TVA maintains three types of queues: request queues, authorized packet queues, and a low priority queue.
- Tag space bounds the number of request queues.
- Number of destination queues is bounded by C/R

Challenges

1. Counter a broad range of attacks, including request packet floods and authorized packet floods
2. Router processing with bounded state and computation
3. Effective authorization policies

Simple policies can be effective

- Client policy
  - Authorize requests that match outgoing ones
- Public server policy
  - Fine-grained capabilities tolerate authorization mistakes.
    - Authorize all initial requests
    - Stop misbehaving senders
  - A server has control over its incoming traffic when overload occurs.

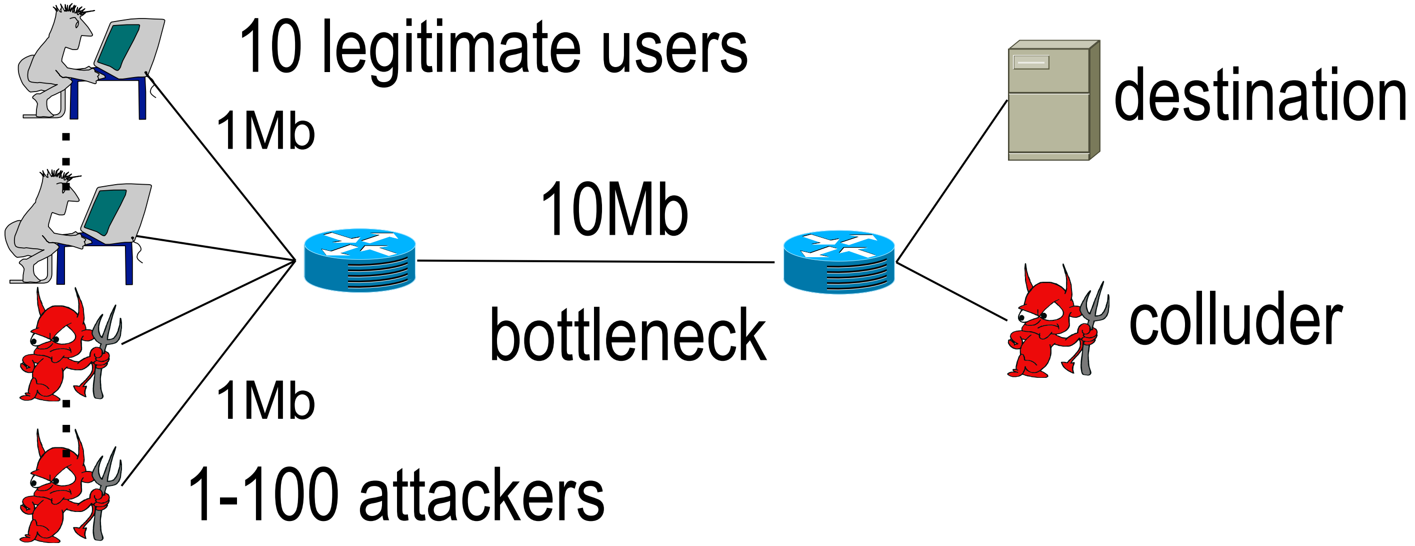
Evaluation

Overview of different schemes

- SIFF [Yarr04]
  - request and legacy traffic have the same priority
  - authorized traffic has a higher priority
  - time-limited capabilities
- Pushback [Mahajan01, Ioannidis02]
  - Network controlled filtering
- Legacy Internet
  - best-effort

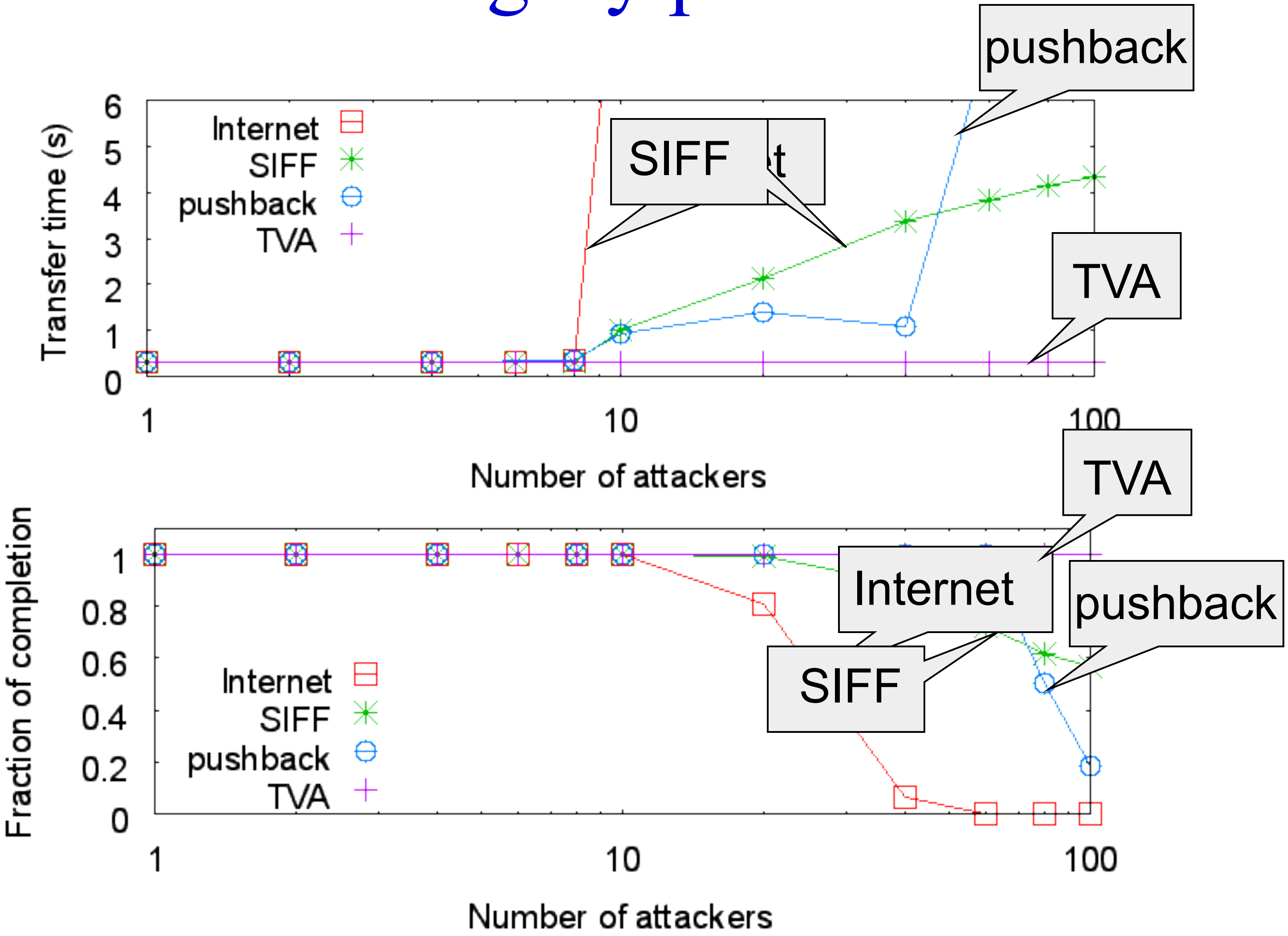
The purpose is to illustrate the cost and benefit of TVA’ s design choices.

Ns-2 Simulation Setup



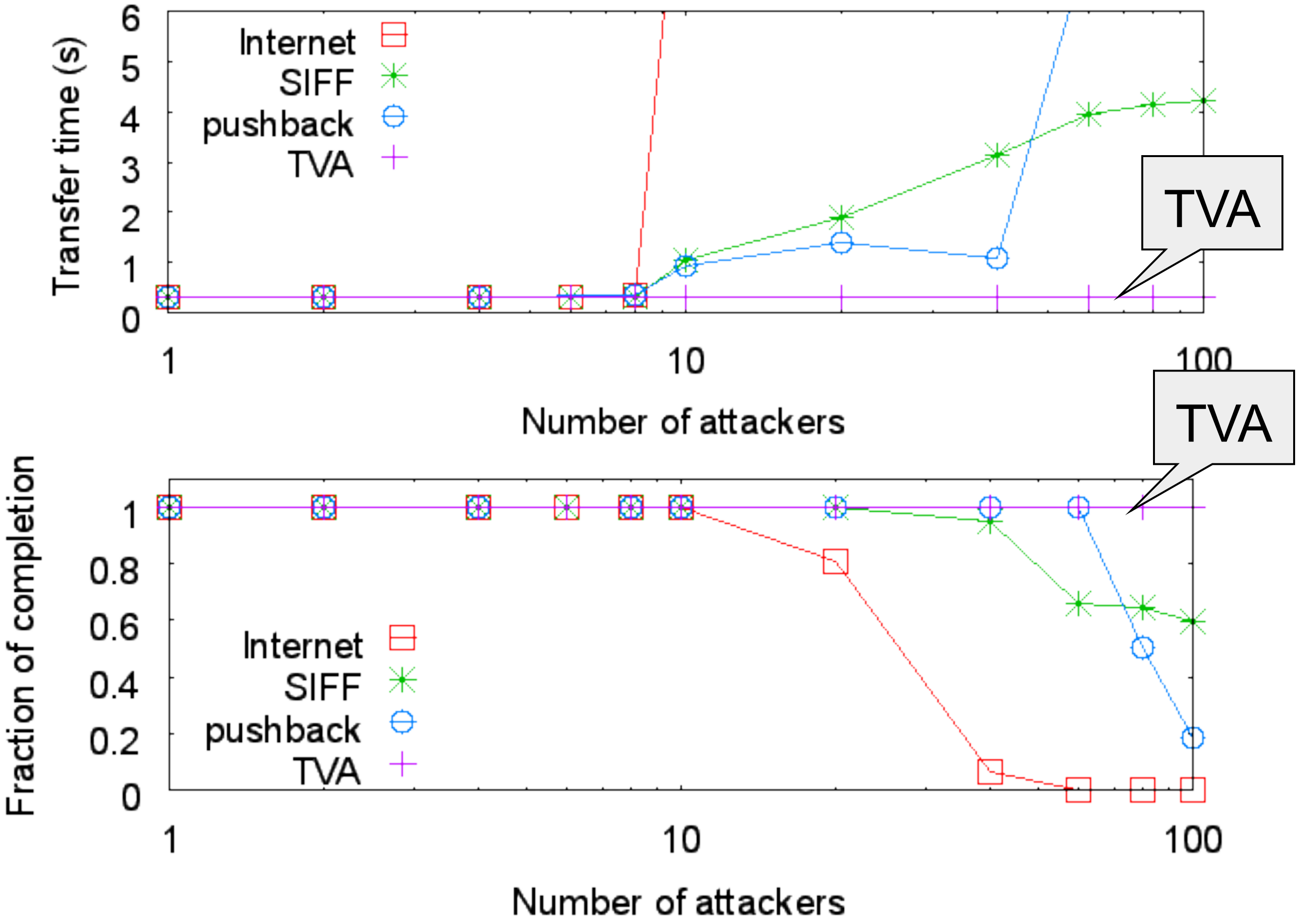
- Scale down topology to speed up simulations
- Two metrics:
  - The transfer time of a fixed-length file (20KB)
  - Fraction of completed transfers
    - A transfer aborts if TCP retransmits the same packets more than 10 times.

Limits legacy packet floods

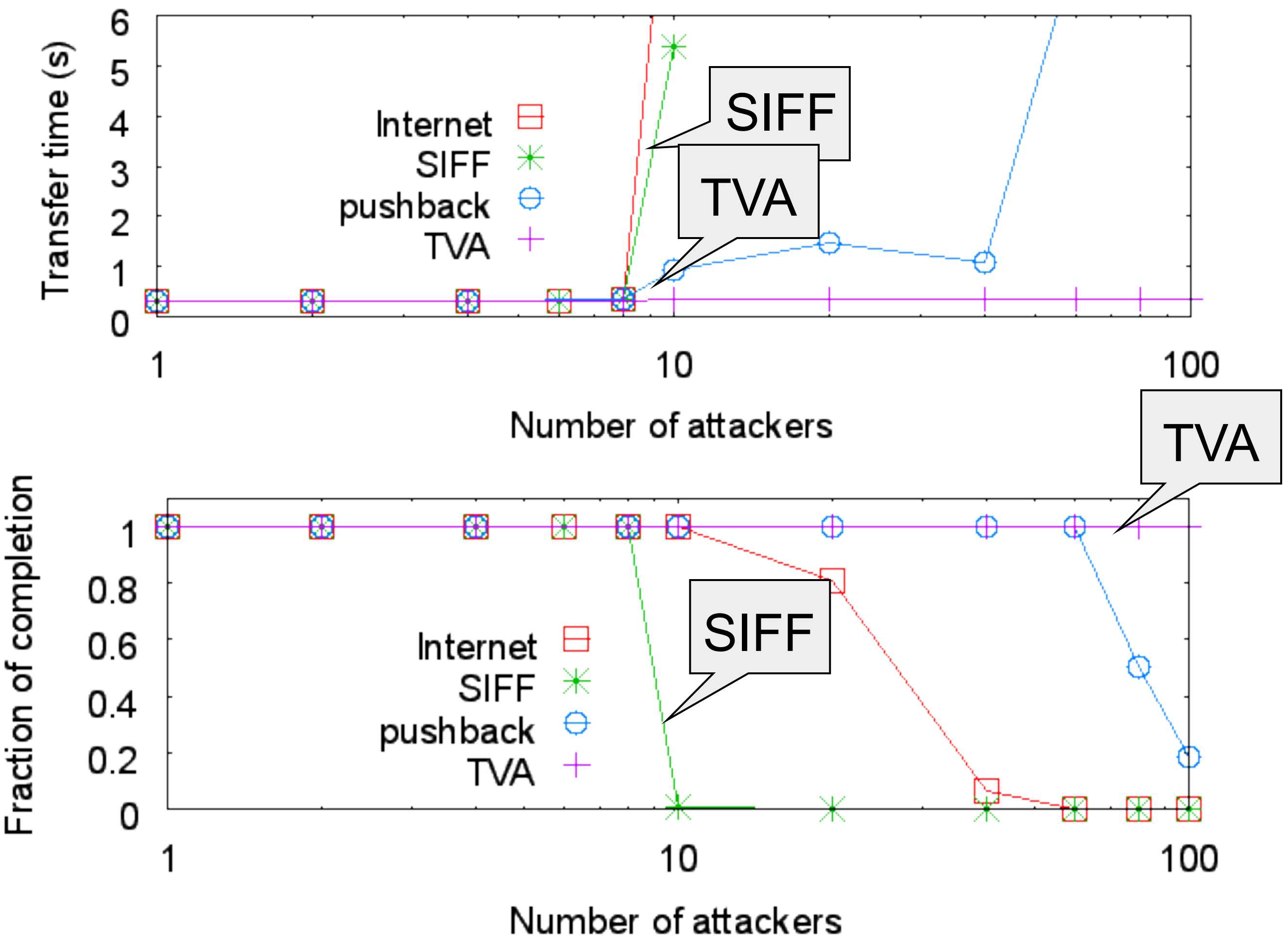


- At number 10, the bottleneck is congested.

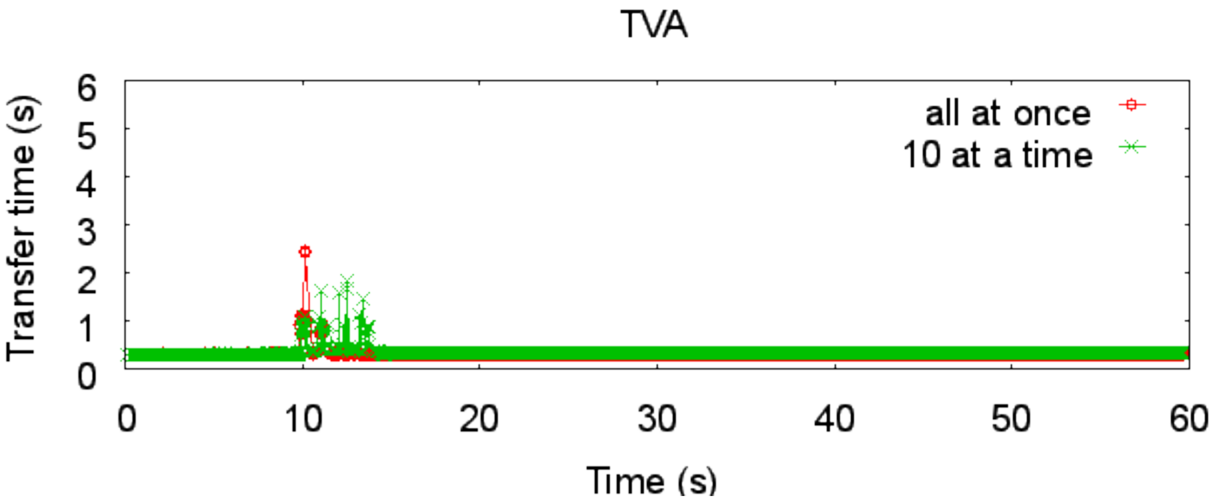
Limits request packet floods

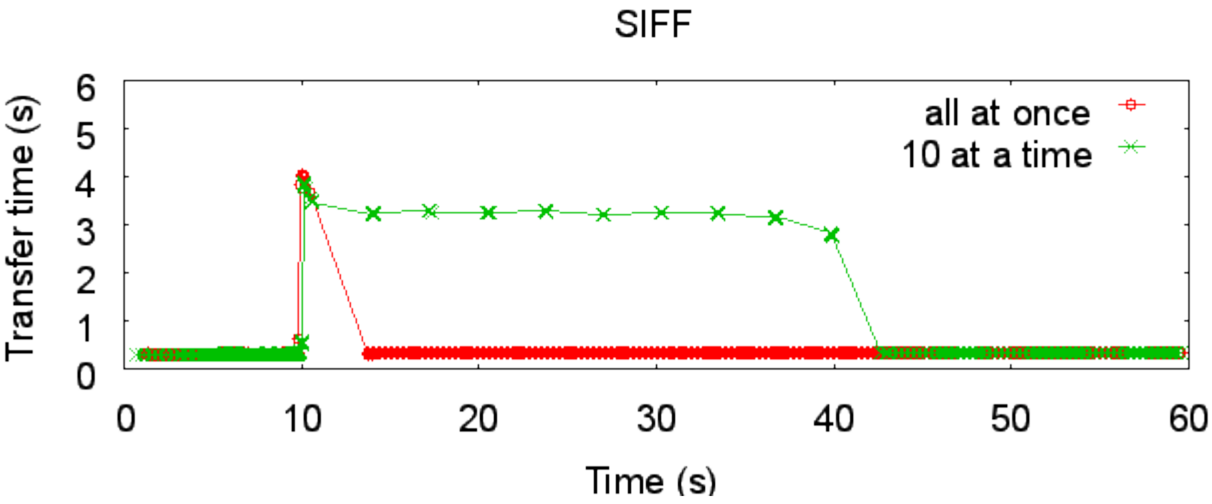


Limits authorized packet floods



Simple policies can be effective





3-second key turn-over time

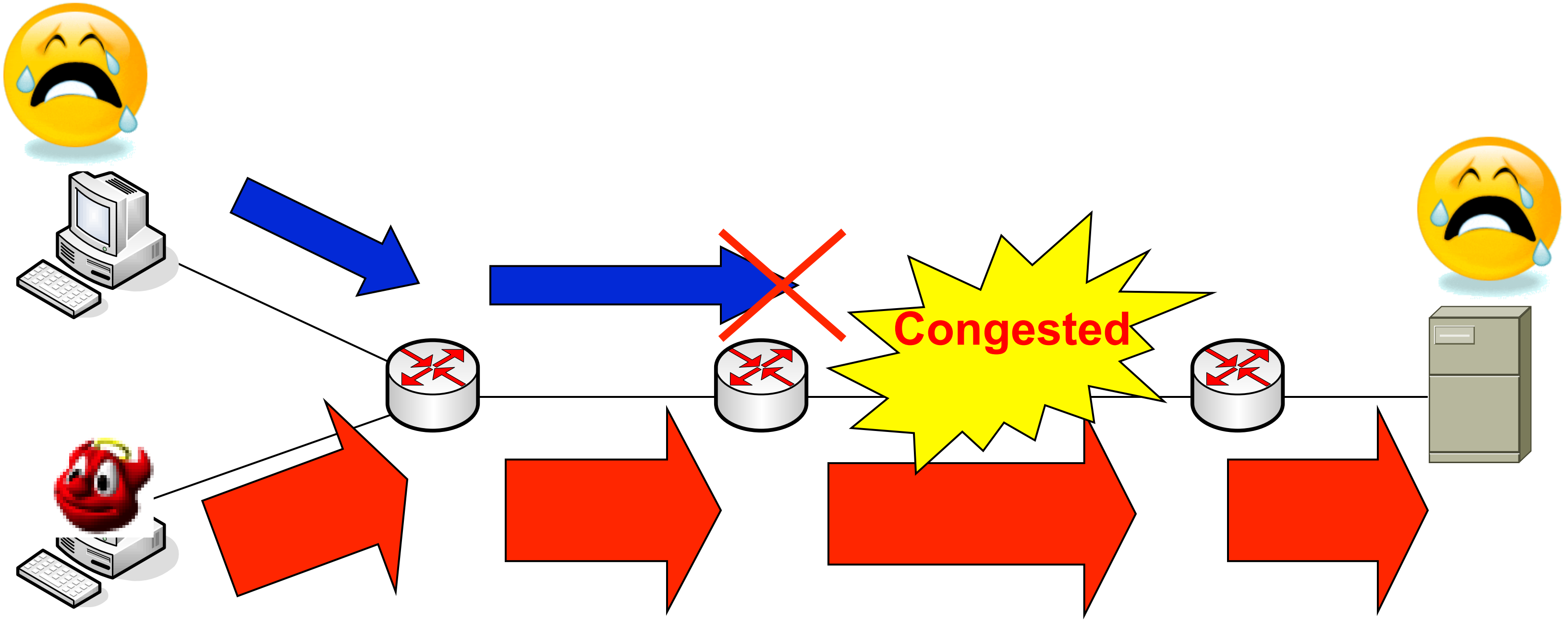
Conclusion

- Key contribution
  - A comprehensive and practical capability system for the first time.
  - Attackers are only as powerful as they are numerous.
- We made TVA practical in three aspects
  - Counter a broad range of attacks
  - Bounded state and computation
  - Simple and effective authorization policies
- Lessons we learned
  - It is feasible to design an open and resilient network, but it is costly.

To Filter or to Authorize: Network-Layer DoS Defense against Multimillion-node Botnets

Xin Liu   Xiaowei Yang   Yanbin Lu

Denial-of-Service Flooding Attack



DoS Flooding Attacks are Serious Threat

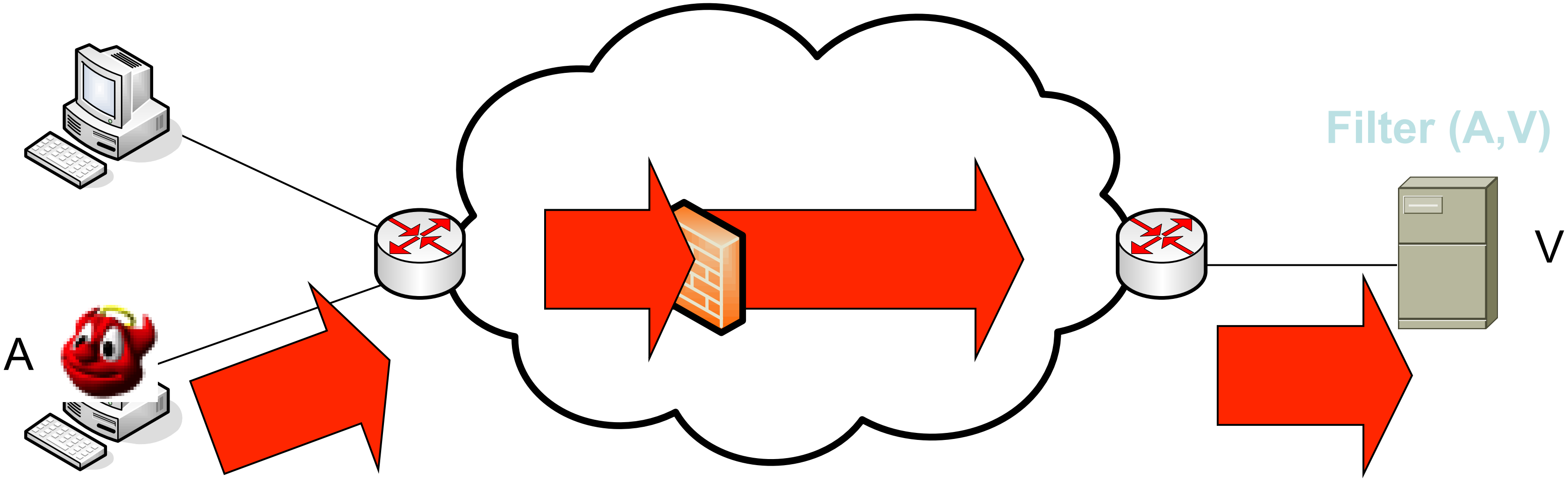
- The attacks appear frequently in news and forums like NANOG
- Attack traffic volume can be enormous
  - Some attacks can reach as high as 10Gbps
    - From Vaughn et al., 2006
- The number of attack sources can be huge
  - 48M bots in 6 months
    - From Rick Wesson, Support Intelligence, LLC, 2007

No Consensus on How to Combat DoS

- Many proposals to mitigate DoS flooding attacks
  - Mayday, AITF, Flow-Cookies, Phalanx, SOS, Pushback, dFence, Portcullis, OverDoSe, CenterTrack, Defense-by-Offense, FastPass, SIFF, TVA, …
- Two intriguing schools of thought
  - Filters
  - Capabilities

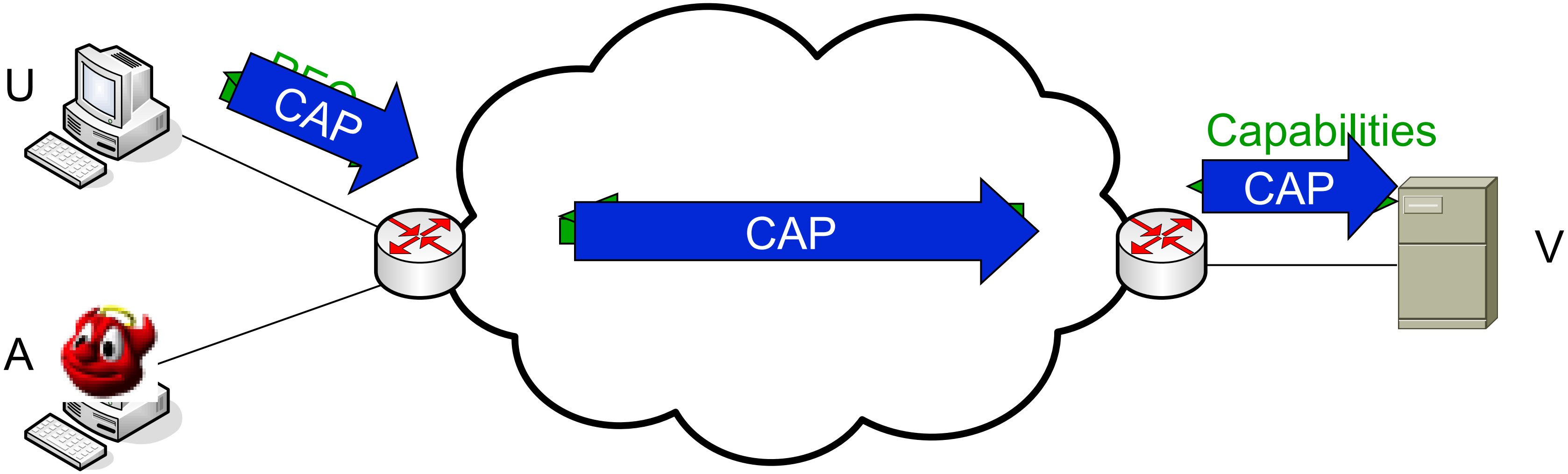
Filter-based Approach

1. Anyone can send to anyone by default
2. A receiver requests the network to install filters



Capability-based Approach

1. Source requests permission to send
2. Destination authorizes source for limited transfer
3. Source places capabilities on packets and sends them
4. Network filters packets based on capabilities



Goal of This Work

To design a DoS-resistant network architecture, should we use filters, capabilities, neither, or both?

"…capabilities are neither sufficient nor necessary to combat DoS."

by K. Argyraki, et al.

"We strongly disagree: … a simple and highly efficient network-based defense … can prevent DoC attacks."

by A. Perrig, et al.

Our Approach

> **“We believe in: rough consensus and running code.”**
> **-- David Clark**

1. Design an effective filter-based system
   - Existing filter systems have several limitations
     - Loss of control messages
     - Filter exhaustion attacks
     - Damage when filters fail to install

2. Compare the effectiveness of filter-based and capability-based systems under various attacks

Design Goals of StopIt

- Effective with little collateral damage
  - Do not block legitimate communications
- Resilient to a wide range of strategic attacks
  - E.g.: impersonation attacks, filter exhaustion attacks
- Fail-safe
  - Limit the damage when filters fail to install
- Incentivizing deployment
  - Early adopters should benefit immediately

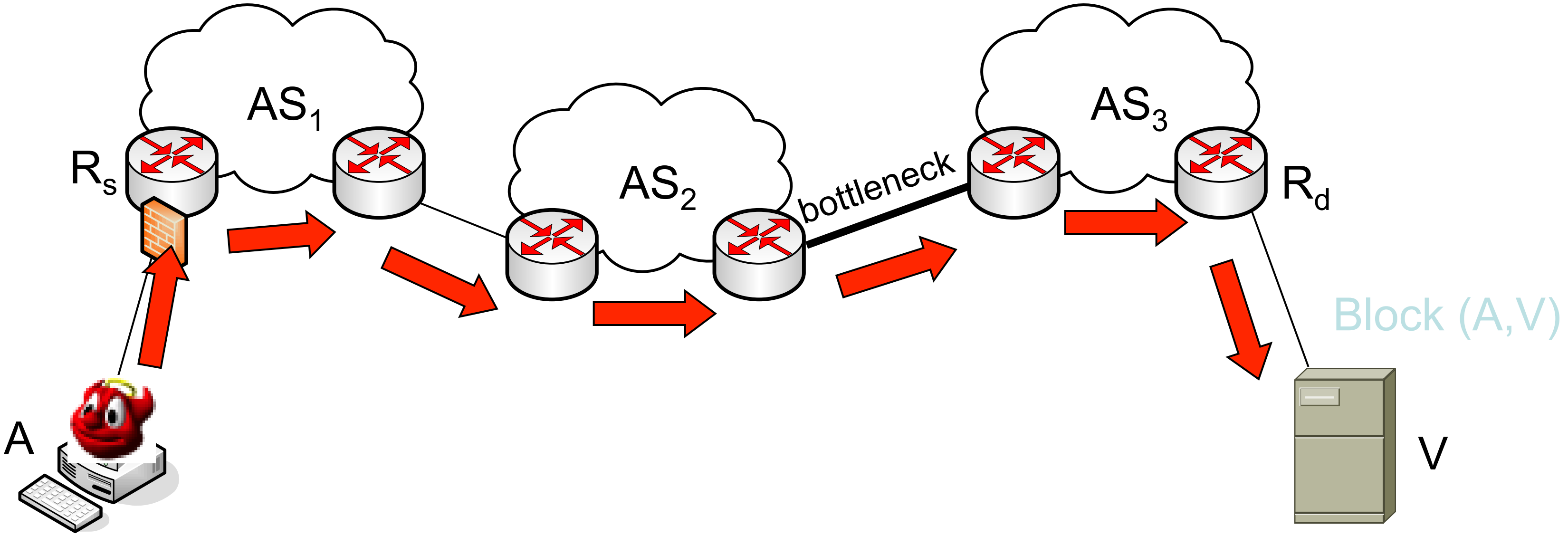
Design Premises

- Similar to capability-based systems

- Simplifying assumptions
  - End systems can distinguish attack traffic
  - Both routers and hosts can be upgraded
  - Securable intra-AS communications

- Practical constraints
  - No special hardware
    - E.g.: no tamper-proof hardware, no line-speed per-packet public key operations
  - Both hosts and routers may be compromised

Overview of an Ideal Filter System

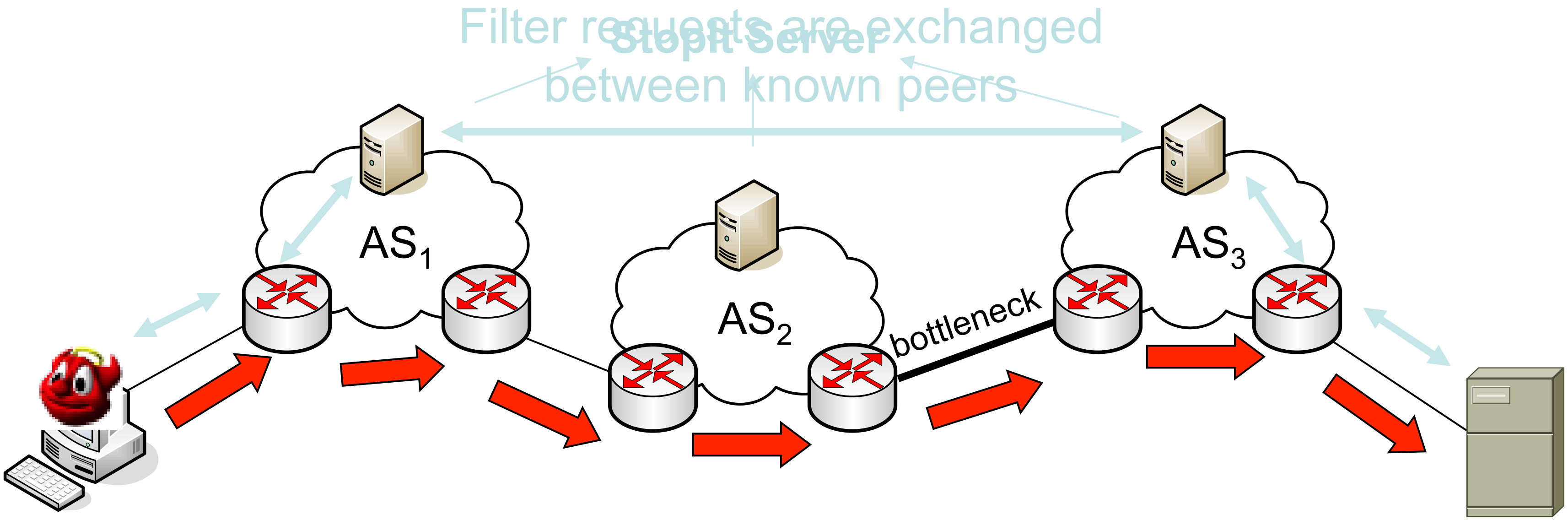


Scalable: no per-flow state in the network core

Secure the Basic Design

Problems	Solutions	
Source address spoofing attacks	Authenticate source addresses with Passport [NSDI’ 08]	
Impersonation attacks	Authenticate filter requests with standard authentication techniques	Closed control channel
Filter exhaustion attacks	Confirm attacks before accepting filter requests; avoid filters against compliant sources; catch and punish misbehaving sources	
Control channel DoS attacks	Source-based fair queuing	
Filters fail to install		
Incentives to deploy		

Closed Control Channel


Filter requests are exchanged between known peers

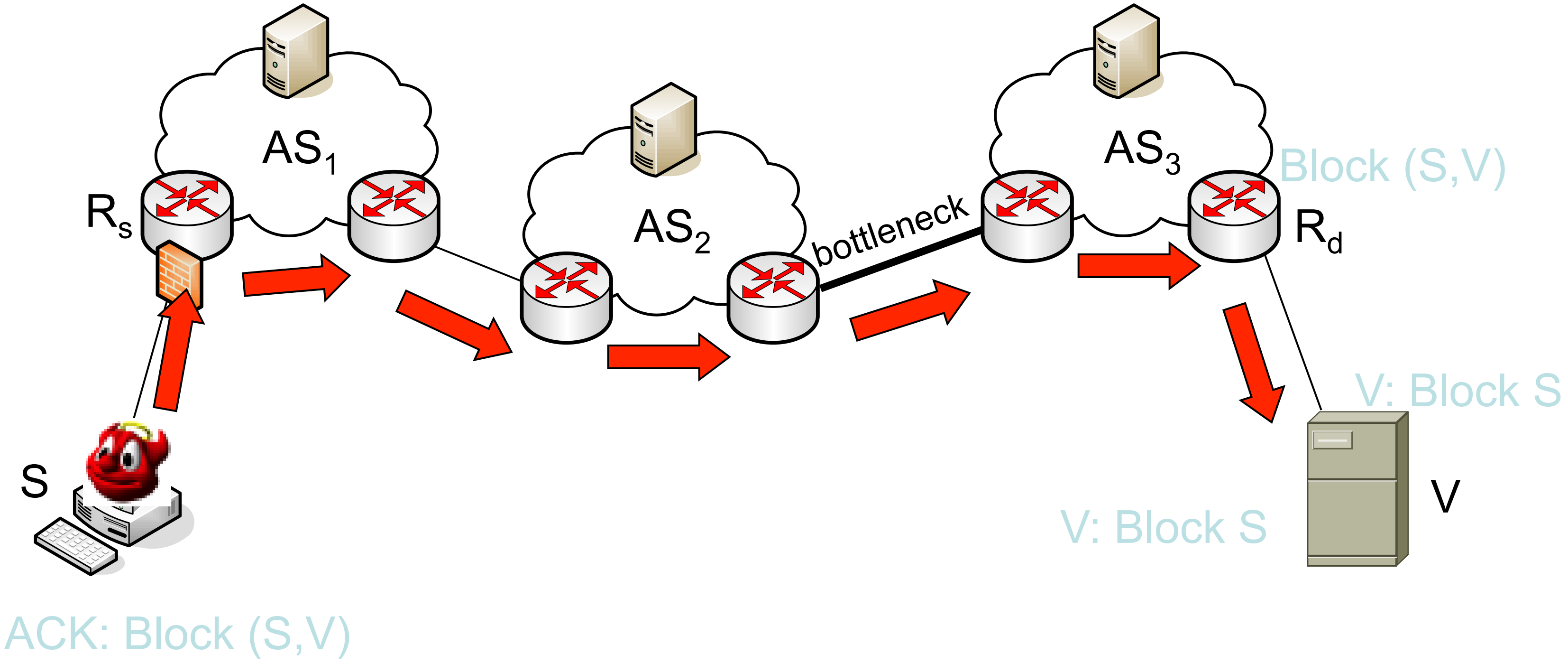
StopIt Server

StopIt Server addresses are published in BGP

BGP Prefix Announcement

10.1.0.0/16		StopIt Server Address	

Steps to Block Attack Traffic



End-to-end requests before submitting filter requests

Attack confirmation on R_d to mitigate filter exhaustion attacks

Use source address and IP-ASN mapping to locate source AS

Request-ACK between S and R_s to mitigate filter exhaustion attacks

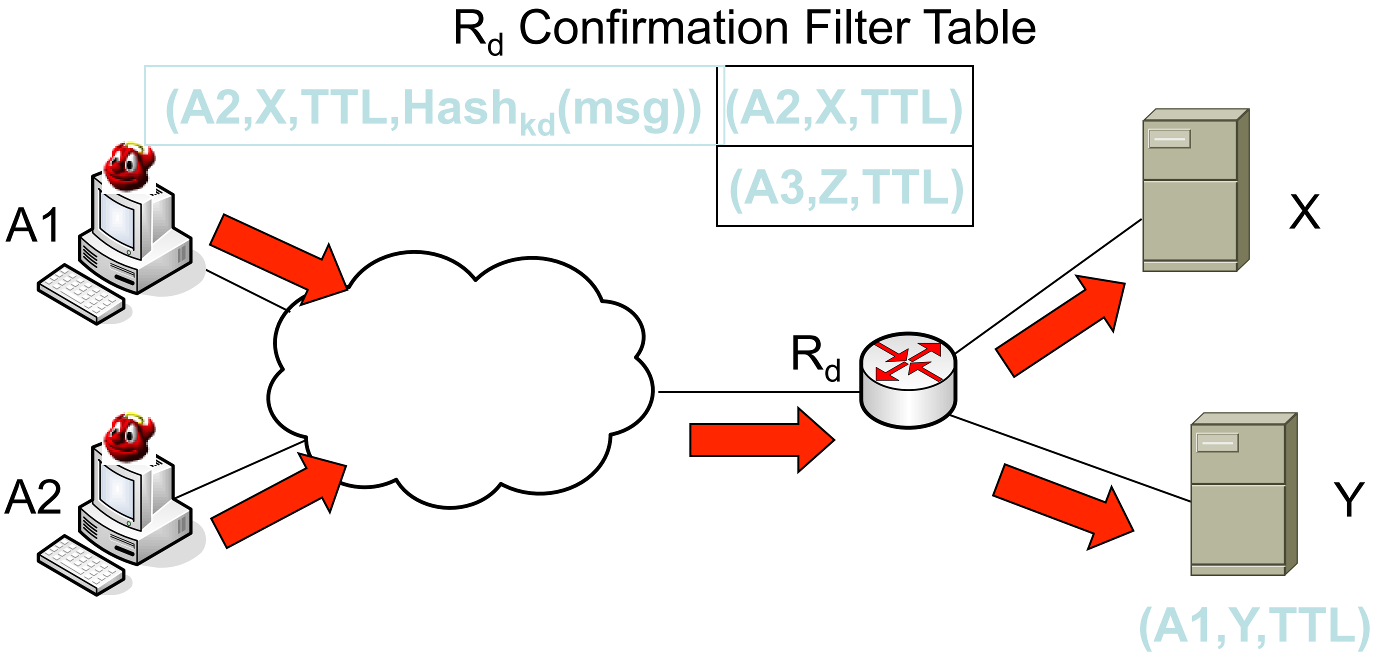
Confirm that Attack Traffic Exists

- Goal: prevent attackers installing filters against non-existent traffic

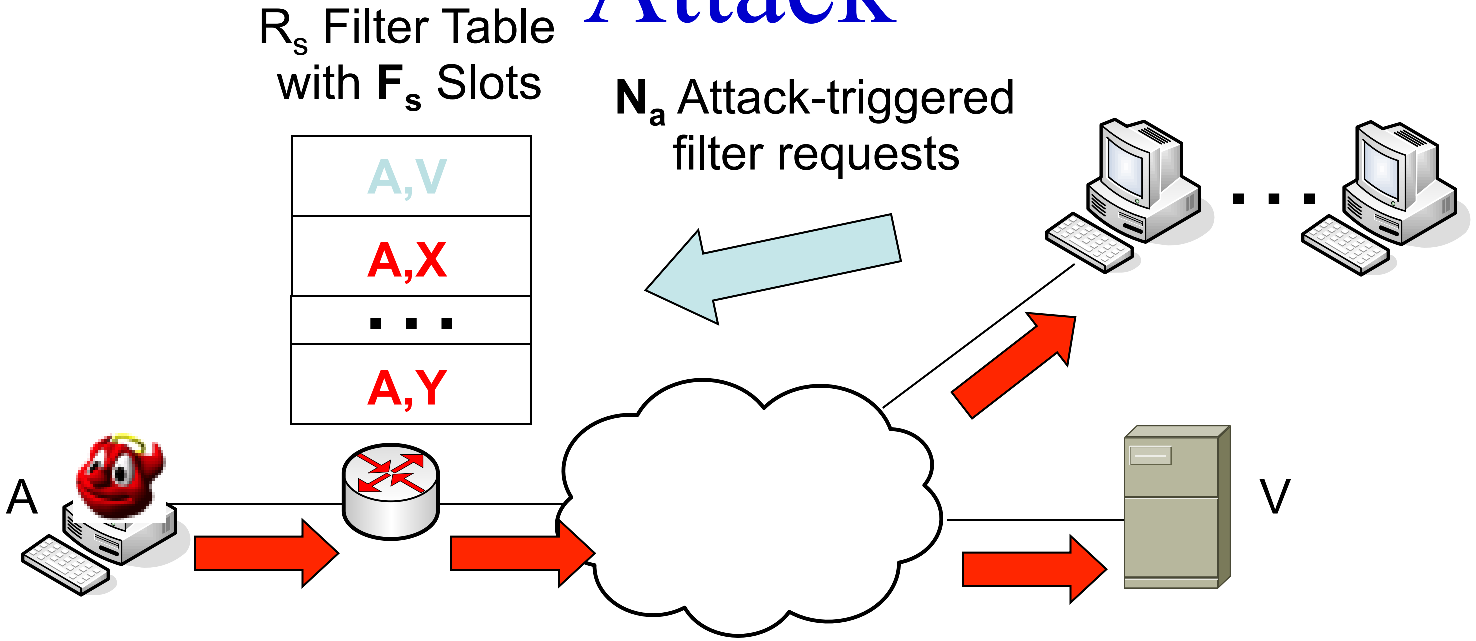
- Confirm attack traffic with flow cache
  - Access routers use flow cache to record recent src-dst pairs
  - Filter requests against traffic not in the flow cache are discarded

Confirm Source is Non-compliant

- Goal: prevent malicious destinations installing filters against compliant sources on source access routers
- Mitigate filter exhaustion: secure filter swapping



Source-side Filter Exhaustion Attack



R_s Filter Table with $\mathbf{F_s}$ Slots

$\mathbf{N_a}$ Attack-triggered filter requests

- Random filter replacement: $P_{caught}=(1-1/F_s)^{Na}$
  - E.g.: if Fs=1k and Na=1k, Pcaught=36.8%
- Aggregate misbehaving sources’ filters
- Quota on filter requests to limit attacker capacity

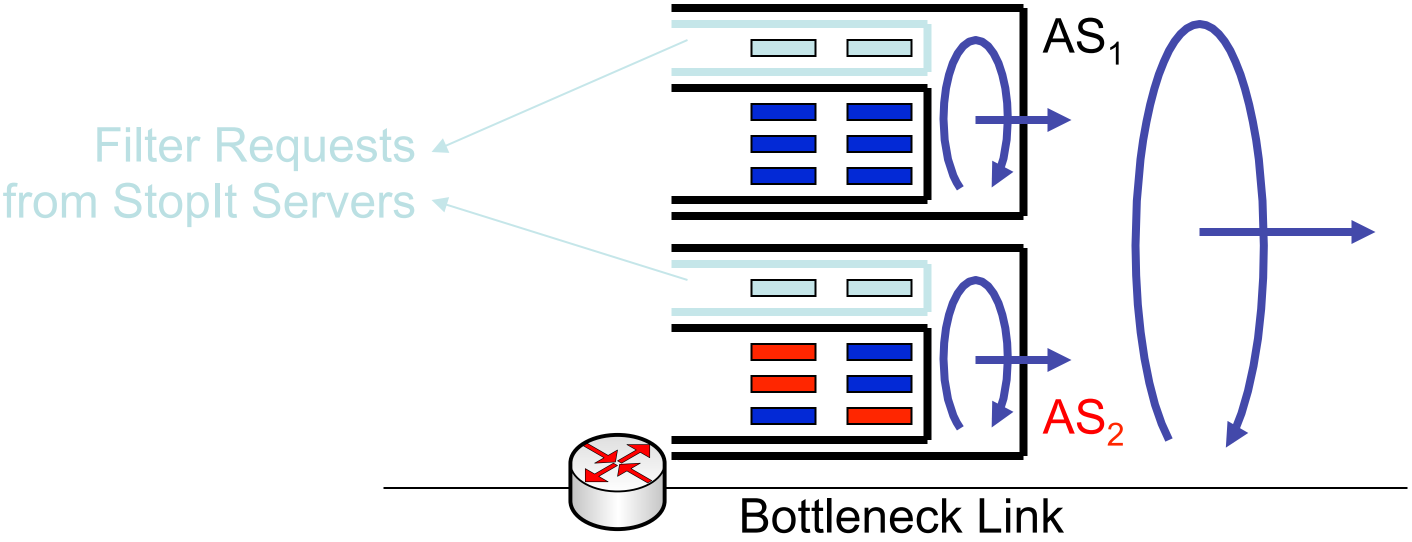
Secure the Basic Design

Problems	Solutions	
Source address spoofing attacks	Authenticate source addresses with Passport [NSDI' 08]	
Impersonation attacks	Authenticate filter requests with standard authentication techniques	Close the control channel
Filter exhaustion attacks	Confirm attacks before accepting filter requests; avoid filters against compliant sources; catch and punish misbehaving sources	
Control channel DoS attacks	Source-based fair queuing	
Filters fail to install		
Incentives to deploy		

Two-level Hierarchical Fair Queuing

- First-level fair queuing: source AS
  - Limit damage of attack traffic when filters fail to install
  - Incentivize deployment

- Second-level fair queuing: source address
  - Give inter-domain filter requests guaranteed bandwidth



Evaluate StopIt

- Prototype implemented on Linux using Click

- Evaluated on Deterlab
  - Block various number of attackers with destination-side filter exhaustion
  - Source-side filter exhaustion attack

- Main Results
  - Block 10M attackers in 1658 seconds
  - With 10M filter slots and 10M daily quota on filter requests, on average an attacker can at most attack a victim 2.4 times per day

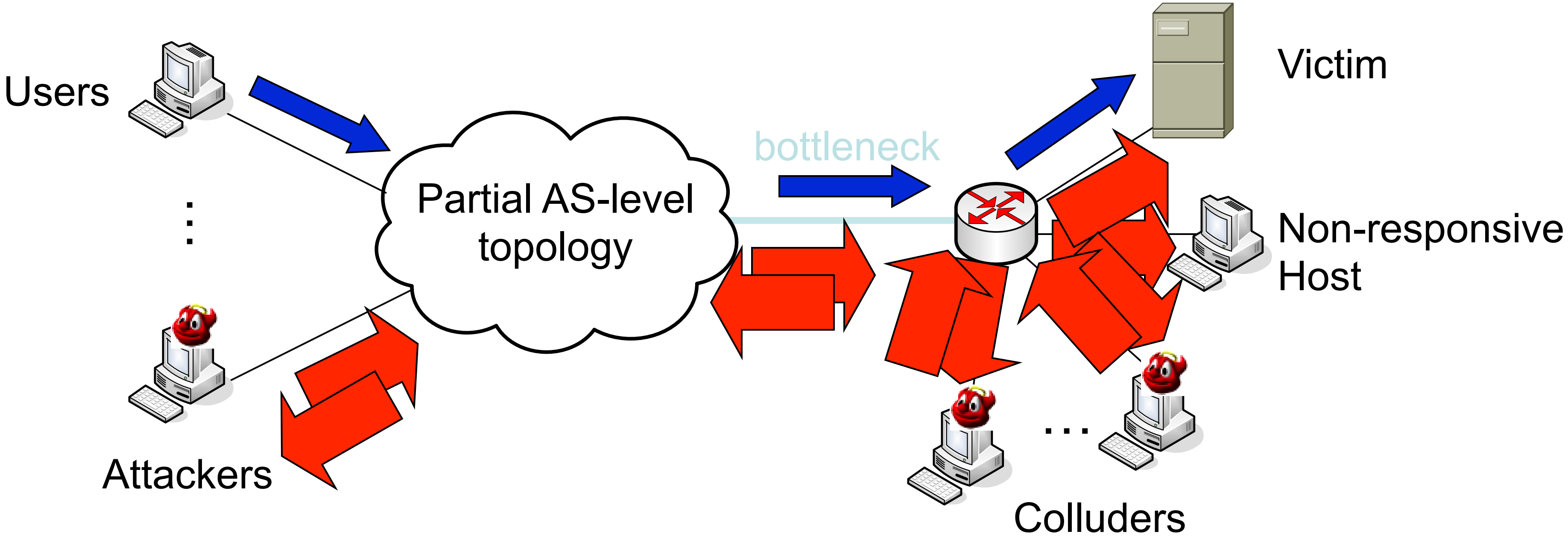
Compare Filters & Capabilities: Settings

- DoS Mitigation Systems
  - Filter-based: StopIt, AITF, Pushback
  - Capability-based: TVA, TVA+(Passport), Portcullis
- Topology
  - a branch of AS-level topology from RouteViews
- Scale-down factor: 1/20
  - E.g., bottleneck bandwidth: 1Gbps(simulated) = 50Mbps(real)
- Metrics of effectiveness
  - Ratio of successful file transfers
  - Average file transfer time

Default file size: 20KB

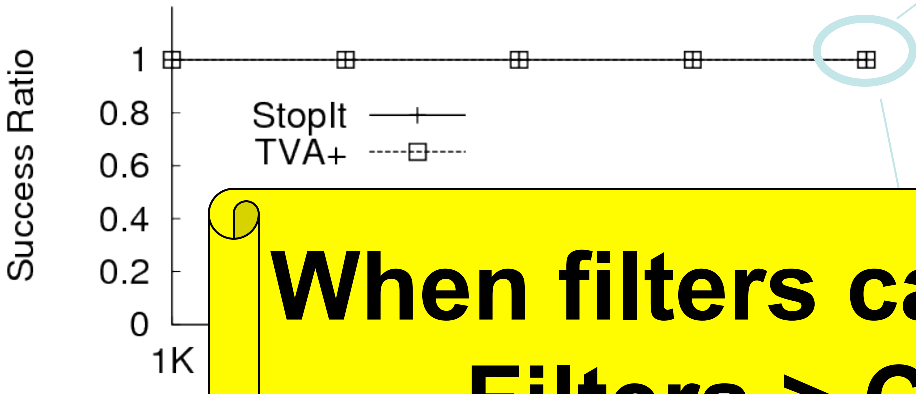
- Default simulated bottleneck bandwidth: 1Gbps

Compare Filters & Capabilities: Attacks



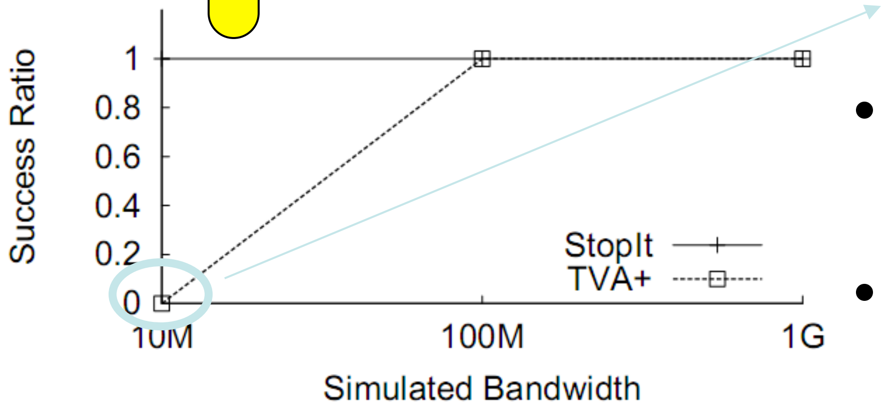
- Destination flooding attacks
- One-way link flooding attacks
- Two-way link flooding attacks

Destination Flooding Attacks



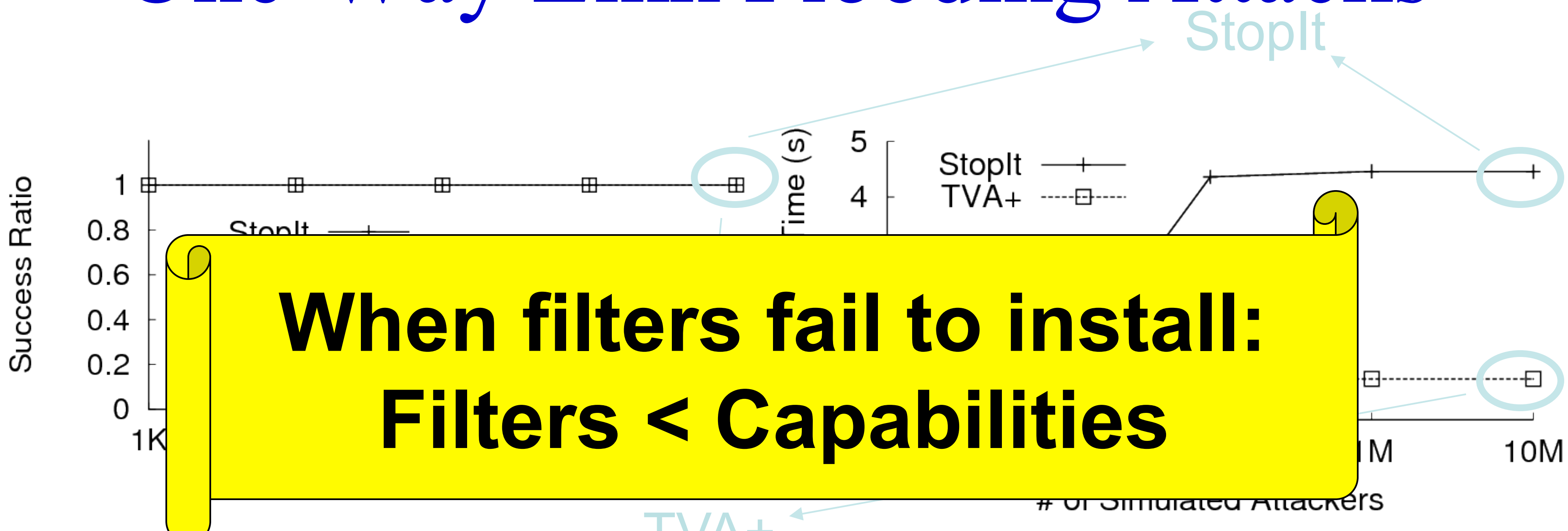
StopIt
fer Time (s)
5
4
3
StopIt
TVA+
10M

When filters can be installed: Filters > Capabilities

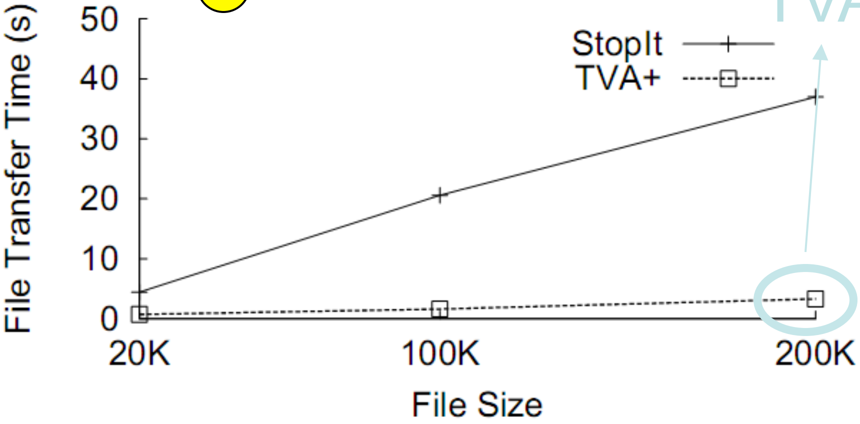


- StopIt
  - Block attack traffic completely
- TVA+
  - Not good with small bottlenecks

One-Way Link Flooding Attacks

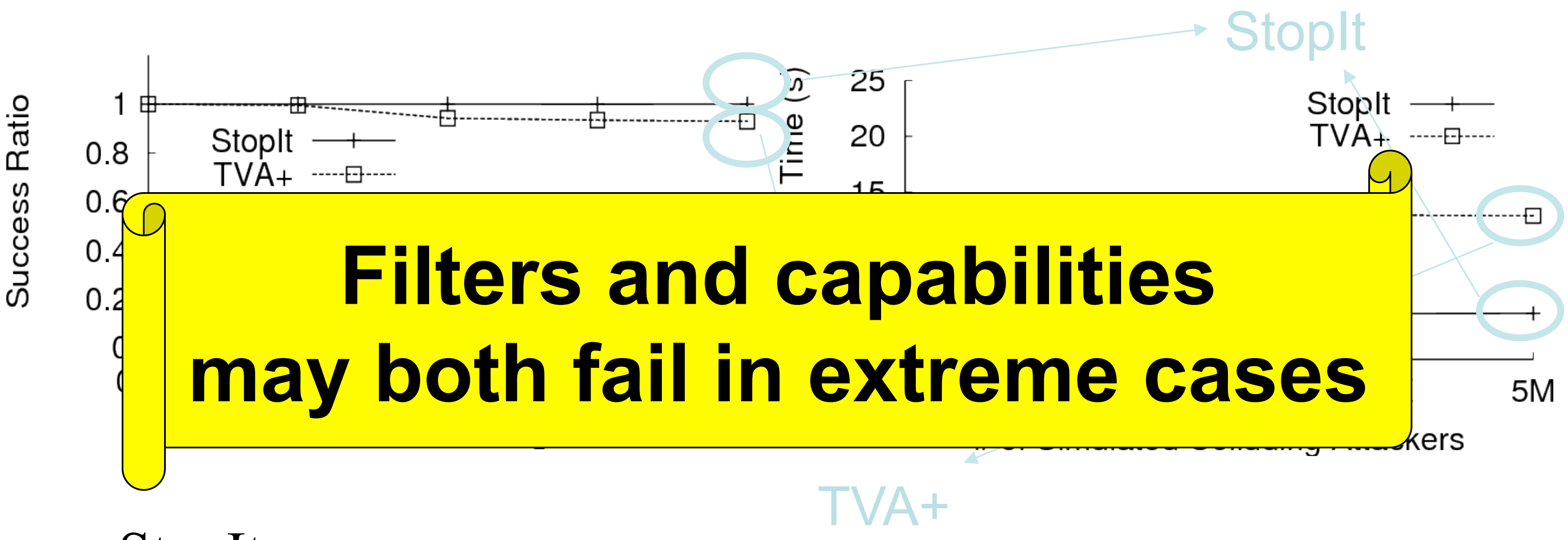


When filters fail to install:
Filters < Capabilities



- StopIt
  - No filters installed; fail-safe
- TVA+
  - More effective when file size increases

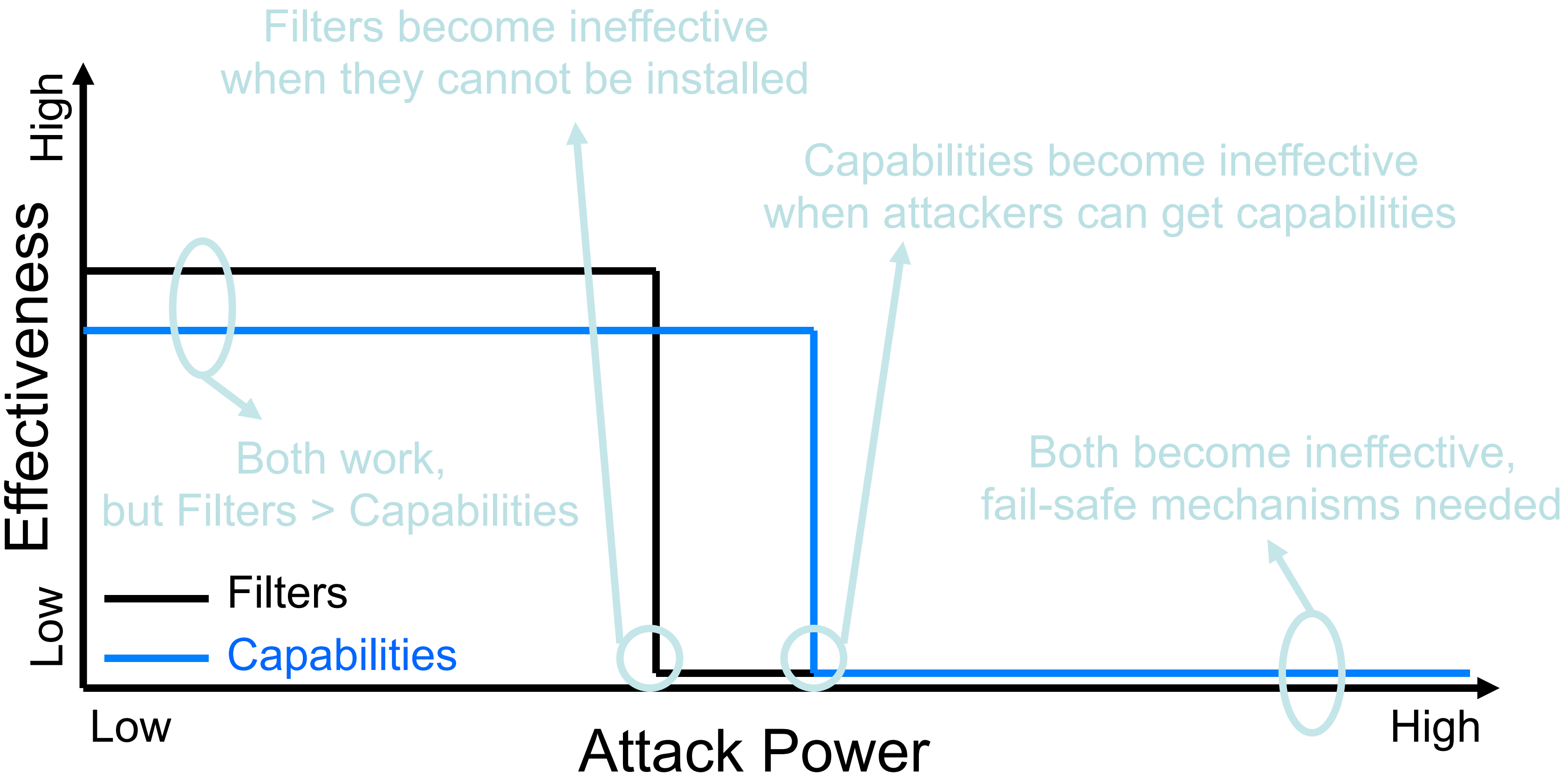
Two-Way Link Flooding Attacks



Filters and capabilities may both fail in extreme cases

- StopIt
  - No filters installed; degraded to per-source FQ
- TVA+
  - Attackers get capabilities; degraded to per-destination FQ
- Under the specific settings, per-src FQ > per-dst FQ

Compare Filters & Capabilities: Summary



Filters become ineffective when they cannot be installed

Capabilities become ineffective when attackers can get capabilities

Both work, but Filters > Capabilities

Both become ineffective, fail-safe mechanisms needed

Conclusion

- It’ s feasible to design an effective filter system
  - Resilient to various attacks
  - Fail-safe
- Filters v.s. Capabilities
  - Filters are more effective if they can be installed
  - Capabilities are more robust against attacks
  - Capability systems tend to be simpler
- Capabilities + Per-AS fairness: might be the most cost-effective solution