

A DoS-limiting Network Architecture

by Xiaowei Yang, David Wetherall
and Thomas Anderson

appears in Proceedings of SIGCOMM 2005

presented by
Jon Turner

 Washington University in St. Louis

Introduction

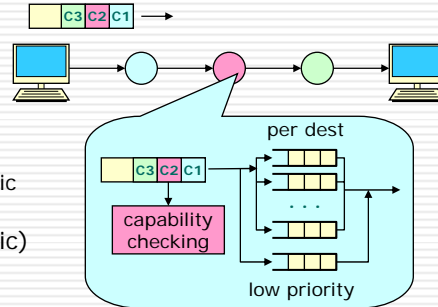
- Problem: protecting against Denial of Service attacks
 - » Internet is intrinsically vulnerable to DoS
 - » attacks have become common and impose significant costs
- Current (partial) solutions
 - » source address filtering
 - » traceback
 - » overlay filtering
 - » pushback
- Proposed Traffic Validation Architecture
 - » comprehensive approach based on *capabilities*
 - » senders must obtain permission from receiver before sending
 - permission represented by capabilities (hard to forge tokens)
 - » routers check capabilities and forward non-compliant traffic at lower priority
 - » other mechanisms protect request channel from DoS
 - » caching used to reduce cost of capability checking

4-2 - Jon Turner - 1/13/2006

 Washington University in St. Louis

Overview of TVA Data Forwarding

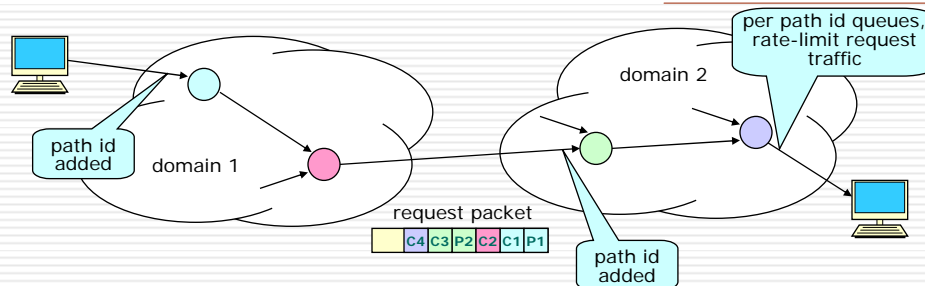
- Packets sent with capability for each router along path.
- Routers check capabilities.
 - » compliant packets placed in per destination queues
 - keep excessive traffic to subverted destination from blocking “good” traffic
 - » non-compliant packets placed in low priority queue (with legacy traffic)
- What’s in a capability?
 - » router i contributes pre-capability
 - local timestamp t_i
 - hash(t_i , src adr, dest adr, secret)
 - secret changes twice per timestamp rollover period
 - use hash function that is hard to invert
 - » destination converts pre-capabilities to capabilities
 - hash(N, T , pre-capability), where N is byte count, T is time limit
 - » packets also include plain-text versions of N and T
- Routers retain state to verify byte count.



4-3 - Jon Turner - 1/13/2006

Washington University in St. Louis

Obtaining Capabilities



- Senders use request packets to obtain capabilities.
 - » piggy-back on TCP SYN packets to avoid extra RTT
- To prevent DoS attack using requests,
 - » path ids (hash of entry id) added to requests at trust boundaries
 - » request packets queued based on path ids
 - » requests forwarded at fraction of link rate (e.g. 10%)
- Ensures that requests don't overwhelm receiver and that most legitimate requests get through.

4-4 - Jon Turner - 1/13/2006

Washington University in St. Louis

Bounding State

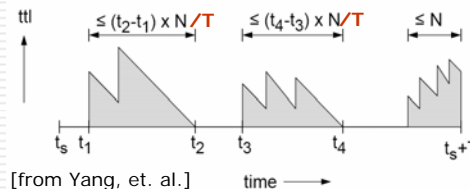
- Most functions require no per flow state in routers.
 - » to check capability, compute
 - hash($N, T, \text{hash}(t_i, \text{sadr}, \text{dadr}, \text{secret})$)
 - most required fields ($N, T, t_i, \text{sadr}, \text{dadr}$) present in packet and only two secrets needed per router
 - » check that capability has not expired by comparing $T + t_i$ to current time
- To check that sender has not sent too many bytes, need record of number sent previously.
 - » creates possibility of attack designed to exhaust router state
- To bound state required,
 - » when router i receives packet from flow j with length L ,
 - if no flow state, create pair (byte count, state exp. time = now + LT_j/N_j)
 - if state present, add L to byte count (and check against limits) and increase state expiration time by LT_j/N_j
 - » if more space needed for state, remove state pair for which expiration time has been reached

4-5 - Jon Turner - 1/13/2006

Washington University in St. Louis

Bounding State (continued)

- Enforcing flow limits.
 - » if flow state was created at t_1 and removed at t_2 , then flow sent at most $(t_2 - t_1)(N_j/T_j)$ bytes during interval
 - » so consider sequence of periods in which flow state is present for flow j , with the last period ending before capability time limit
 - number of bytes sent during these periods is at most N_j
 - » if state expires just before capability time limit, can send at most N_j more bytes
- Given minimum rate $(N/T)_{min}$ for capabilities, an input link with capacity C requires state for at most $C/(N/T)_{min}$ flows (for 10 Gb/s link, 100 kb/s min rate, need 100K flow state records)



4-6 - Jon Turner - 1/13/2006

Washington University in St. Louis

Reducing Packet Overhead

- Capability header adds at least 8 bytes to packet header for each router on path.
- Can reduce overhead by caching capabilities.
 - » sender includes random nonce with packets
 - » routers cache relevant capability information and nonce
 - » after first packet sent, sender includes only nonce with packet
 - » router checks nonce against value stored with per flow data; if nonce matches, it does resource checks and updates state
- If packet is received with nonce, when no flow state is present, it is forwarded at low priority
 - » senders can prevent this by sending full capability when cache state has expired
 - » senders maintain state expiration time using same algorithm as router – send full capability for packets with expired state
 - » flows sending below their nominal rate may need to include capabilities with each packet – expensive for short packets

4-7 - Jon Turner - 1/13/2006

Washington University in St. Louis

Limiting Impact of Route Changes

- Route changes invalidate capabilities.
 - » packet takes path different from one for which capability was constructed
 - » routers mark packets with invalid capabilities and forward at low priority
- When destination receives packet with invalid capability,
 - » it marks bit in next return packet sent, informing sender that it needs to request new capability
 - » sender then issues request packet, triggering construction of new capability by routers on new path
- Requires route changes to be relatively infrequent.

4-8 - Jon Turner - 1/13/2006

Washington University in St. Louis

Limiting Cost of Fair Queueing

- Fair queueing on destination address prevents receivers from getting unfair share of link bandwidth.
 - »but, this requires separate queue per receiver
- To limit cost of queueing
 - »maintain separate queues only for flows with per flow state
 - »remaining flows placed in separate shared queue
- Ambiguities in text
 - »appears to advocate use of per-flow queues, not per-receiver queues,
 - enables bandwidth hogging using single receiver, many spoofed source addresses
 - correct by using per receiver queue whenever some flow for that receiver is in cache – requires reference count per receiver queue
 - »does not address packet ordering issue raised by queue switching

Short, Slow or Asymmetric Flows

- TVA most efficient for long, higher rate flows.
- Short flows can be quite inefficient (e.g. DNS queries).
- Effect on aggregate efficiency is small.
 - »assuming that most traffic carried by longer, high rate flows
- Ratio of request traffic to data traffic may be different for different links.
 - »network operators must determine appropriate ratio for the traffic mix on a given link
 - »extreme example: link leading to root DNS server will mostly carry request traffic

Implementation

Common Header

version (4)	type (4)	upper protocol (8)
-------------	----------	--------------------

1xxx: demoted
 x1xx: return info
 xx00: request
 xx01: regular w/ capabilities
 xx10: regular w/ nonce only
 xx11: renewal

Request Header

common header (16)	
capability num (8)	capability ptr (8)
path-id 1 (16)	
blank capability 1 (64)	
• • •	
path-id n (16)	
blank capability n (64)	

Regular / Renewal Header

common header (16)	
flow nonce (48)	
capability num (8)	capability ptr (8)
N (10)	T (6)
capability 1 (64)	
• • •	
capability n (64)	

cached

Return info

return type (8)

00000001: demotion notification
 0000001x: a 8-bit capability num field, N, T, and a list of return capabilities follow this field.

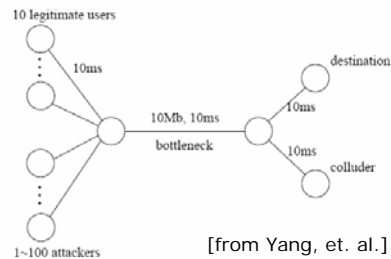
- Shim header that precedes IP header.
- Request packets
 - » path id per domain (16 bits)
 - » pre-capability per router (64 bits)
 - eight bit timestamp plus hash
- Data packets with cap.
 - » flow nonce (48 bits)
 - » N, T (16 bits) – units?
 - » capability per router (64 bits)
- Data packets with no cap.
 - » flow nonce (48 bits)

[from Yang, et. al.]

4-11 - Jon Turner - 1/13/2006
Washington University in St. Louis

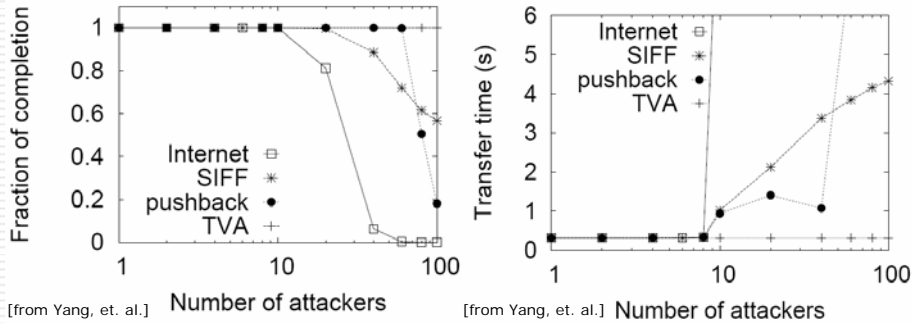
Simulation Setup

- Bottleneck link shared by
 - » 10 legitimate users
 - each repeatedly sends a 20 KB file
 - TCP limits each to 533 Kb/s, so use only 53% of bottleneck link
 - » 1-100 attackers
 - » one legitimate destination and one “colluder” at far end



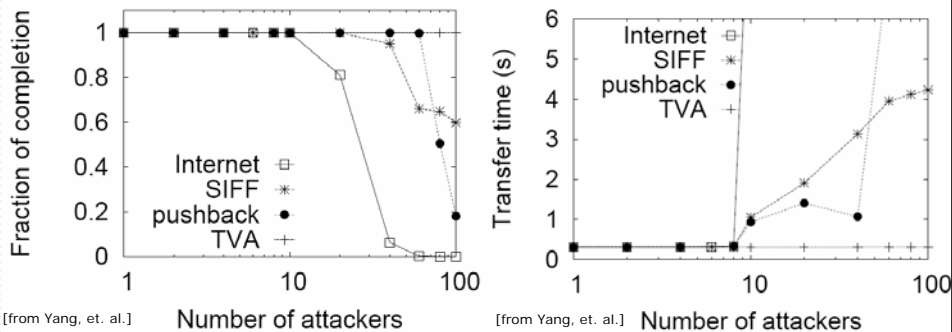
- Compared systems
 - » Stateless Internet Flow Filter (SIFF)
 - earlier capability-based scheme
 - no special handling of requests
 - » Pushback
 - during attack, recursively insert destination-based rate limiters on most loaded input links
 - » plain old Internet

Legacy Packet Floods



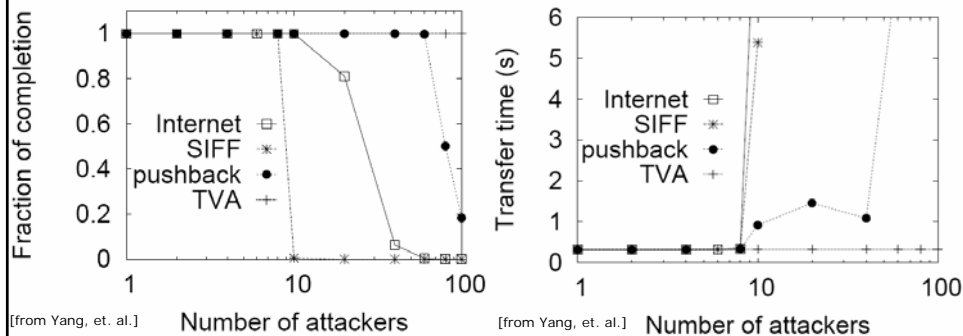
- Fraction of completions measures fraction of 20 KB files successfully transmitted before sender gives up.
 - » TCP connection modified to attempt 8 SYNs with 1 second timeout
 - » abandon connection if >64 second retransmission timeout, or some packet sent >10 times
- With few attackers, pushback identifies and filters attackers – with many, it cannot distinguish attackers from legitimate users.
- SIFF request packets compete with legacy traffic, causing failures.

Request Packet Floods



- Assumes destination can distinguish legitimate requests from attack packets.
- Essentially same behavior as with legacy packets.

Authorized Packet Floods



[from Yang, et. al.]

Number of attackers

[from Yang, et. al.]

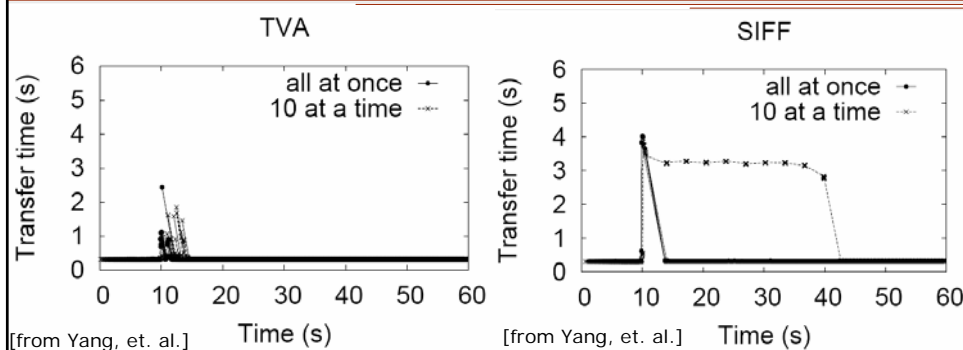
Number of attackers

- Attackers get valid capabilities from colluder and then flood bottleneck link.
- Per destination queueing means that colluder and legal receiver share link bandwidth equally, so TVA receiver unaffected.
 - » with three colluders, receiver's bandwidth share can be halved
- Pushback behavior same as before.
- SIFF fails to establish connections due to low priority requests.

4-15 - Jon Turner - 1/13/2006

Washington University in St. Louis

Coping with Authorization Errors



[from Yang, et. al.]

Time (s)

[from Yang, et. al.]

Time (s)

- Destination grants all requests, but limits initial capability to 32 KB in 10 seconds, and does not renew attacker capabilities.
- Two attack scenarios – all at once, or sequential groups of 10.
- With TVA, 100 attackers can send 3.2 MB in 10 seconds, so can send at > 5 Mb/s for 5 seconds – similar for groups of 10.
- SIFF revokes capability only on timeout expiration, so groups of 10 can sustain attack for longer period.

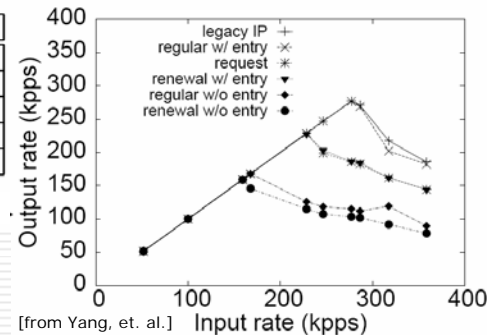
4-16 - Jon Turner - 1/13/2006

Washington University in St. Louis

Computation Cost

Packet type	Processing time
Request	460 ns
Regular with a cached entry	33 ns
Regular without a cached entry	1486 ns
Renewal with a cached entry	439 ns
Renewal without a cached entry	1821 ns

[from Yang, et. al.]



- Computation cost of capability processing on 3.2 GHz Xeon.
- Data packet with capability suffers penalty of 45x relative to packet with cached entry – prefer to keep state for >200 packets.
- Cycle budget for IXP 2800 NP on 10 Gb/s link is 18 cycles/byte.
 - » packet carrying 10 capabilities plus TCP/IP headers is 120B, giving us 2160 cycles – vs. 4755 cycles used by Xeon
- Peak throughput of Xeon is 1% of what's needed for 10 Gb/s link.

4-17 - Jon Turner - 1/13/2006

Washington University in St. Louis

Security Analysis

- Using another host's capabilities.
 - » to be effective, must share same path to destination
 - » such attackers are indistinguishable from legit. host, so host suffers
- Forging capabilities.
 - » strong cryptographic hash functions, key changes every 128 seconds
 - » effective attacker must break hash in <<128 seconds
- Discovering pre-capabilities by triggering their return in ICMP error messages.
 - » allows sender to substitute different N , T values
 - » block by using packet formats that do not place pre-capabilities in first eight bytes of the packet returned by ICMP error message
 - apparently assuming ICMP applied to TVA packets, as with IP packets
- Compromised router masquerading as receiver (or receivers).
 - » attacker sends requests to colluding router, which returns capabilities
 - » affects upstream traffic, but downstream traffic is best-effort
 - » can crowd out traffic to receiver on congested upstream links
- Attacks on resources at capability routers
 - » can be provisioned for worst-case, independent of attacker behavior

4-18 - Jon Turner - 1/13/2006

Washington University in St. Louis

Summary

- TVA provides greatly improved resistance to DoS attacks.
- Key elements
 - » packet transmissions authorized using capabilities
 - » special treatment of requests, to prevent DoS for requests
 - » resource limits on capabilities to mitigate effects of bad authorizations
- Limitations
 - » little protection if many colluders or compromised router
 - » high bandwidth overhead for short, low rate flows
 - » requires NPs that are at least twice as fast for given link bandwidth
- What about alternative of ATM-style VC setup plus datagrams?
 - » with per VC queues or entry policing, attackers cannot steal bandwidth
 - » with explicit signaling channels, attacker cannot prevent legitimate users from establishing new connections
 - cryptographic authorization required to obtain signaling channel
 - » some mechanism (e.g. source-based queueing) still needed to protect mechanism for setting up signaling channel
 - attacks here do not prevent new communication, just new “host connects”
 - only need low rates, so relatively easy to protect
 - » datapath cost and bandwidth cost far lower than TVA