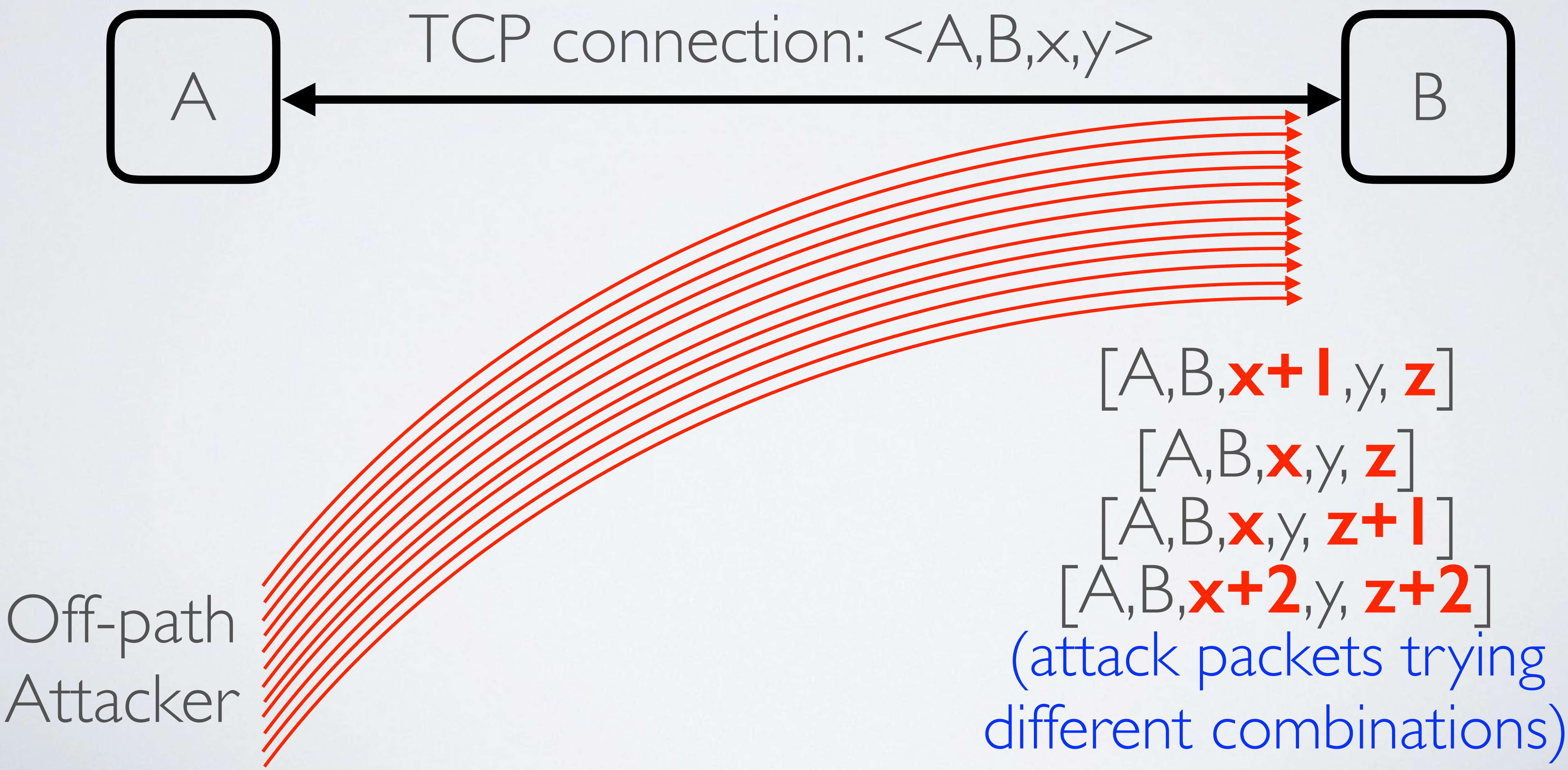
Resilience of Deployed TCP to Blind Attacks

Matthew Luckie, Robert Beverly,
Tiange Wu, Mark Allman, kc claffy

IMC 2015, October 28th 2015

What is a Blind Attack on TCP?

- A brute-force attempt by an **off-path attacker** to disrupt an in-progress TCP connection



What is a Blind Attack on TCP?

- A brute-force attempt by an **off-path attacker** to disrupt an in-progress TCP connection
- Attack methods (RFCs 4953 and 5961):
  - **RST attack**: cause an existing TCP connection to be reset
  - **SYN attack**: cause an existing TCP connection to be reset
  - **Data attack**: cause an existing TCP connection to accept the attacker's data, or enter an ACK war.
- Problematic with **long-lived connections** (e.g. BGP, SSH) and **large windows** (e.g. rsync)

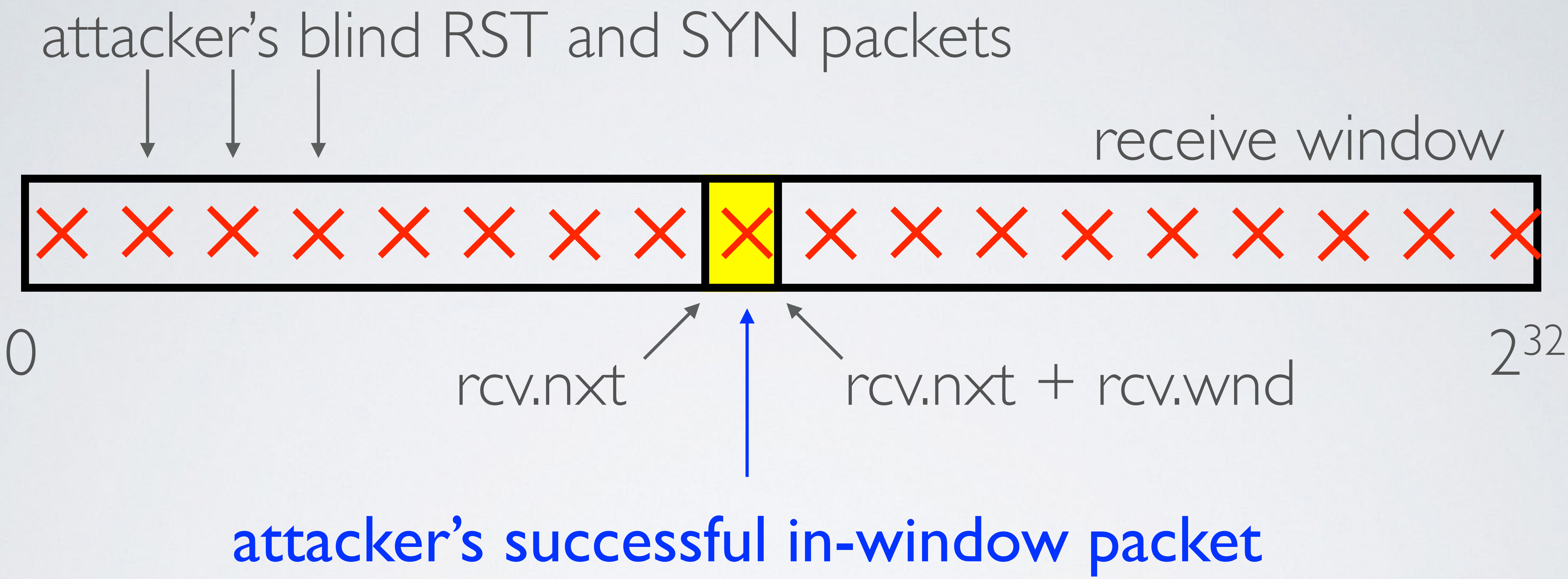
History

- Paul Watson: CanSecWest 2004 “Slipping in the Window”
  - Showed feasibility of a blind reset attack. RFC 793 “**a reset is valid if its sequence number is in the window**.”
    - Larger receive windows reduce an attacker’s work.
  - Attacker must guess source and destination IP addresses, and source and destination ports of victim’s connections.
    - Operating systems in 2004 chose ephemeral ports **sequentially from a small range**.

Slipping in the Window: RST or SYN

"a reset is valid if its sequence number is in the window"

- RFC 793

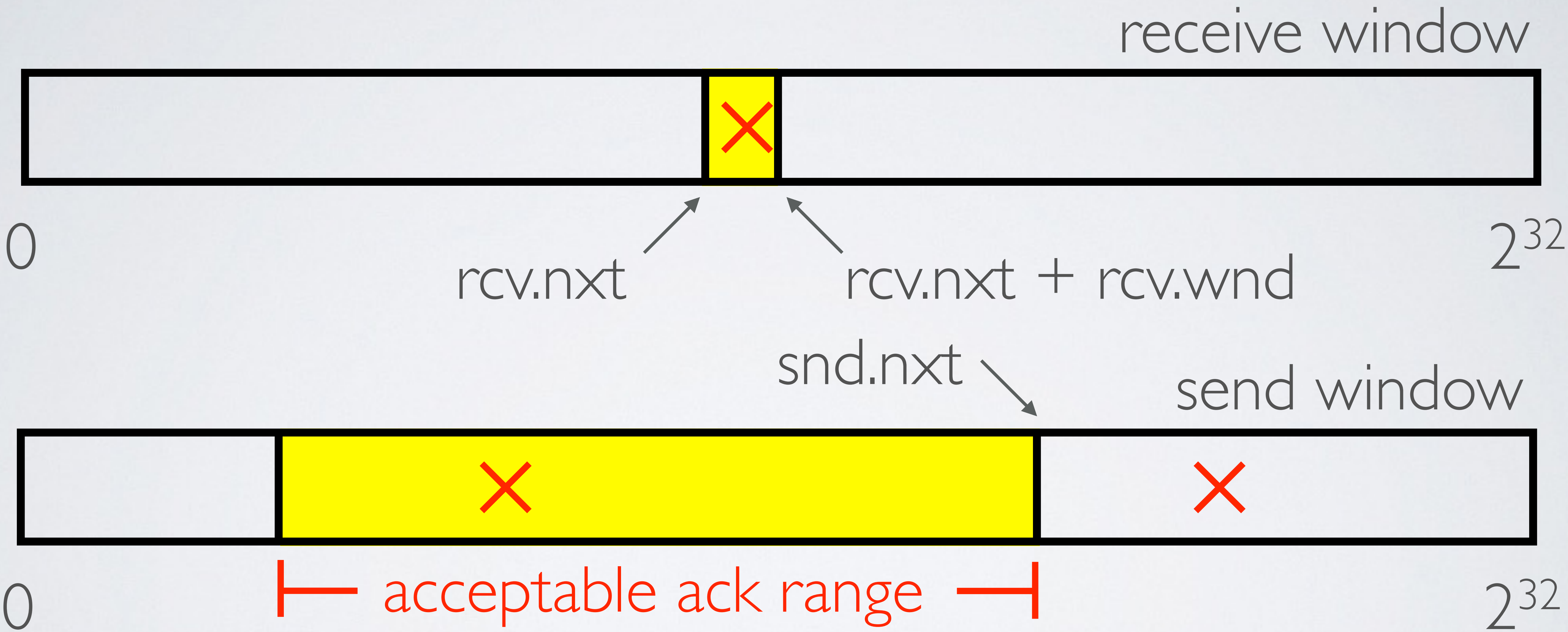


Theoretical receive window of 32k: up to 2^{17} packets.
Attacker constrained by network capacity.
Can complete in <1 second on 100Mbps Ethernet.

Slipping in the Window: Data

"an acknowledgement value is acceptable as long as it is not acknowledging data that has not yet been sent"

- RFC 793



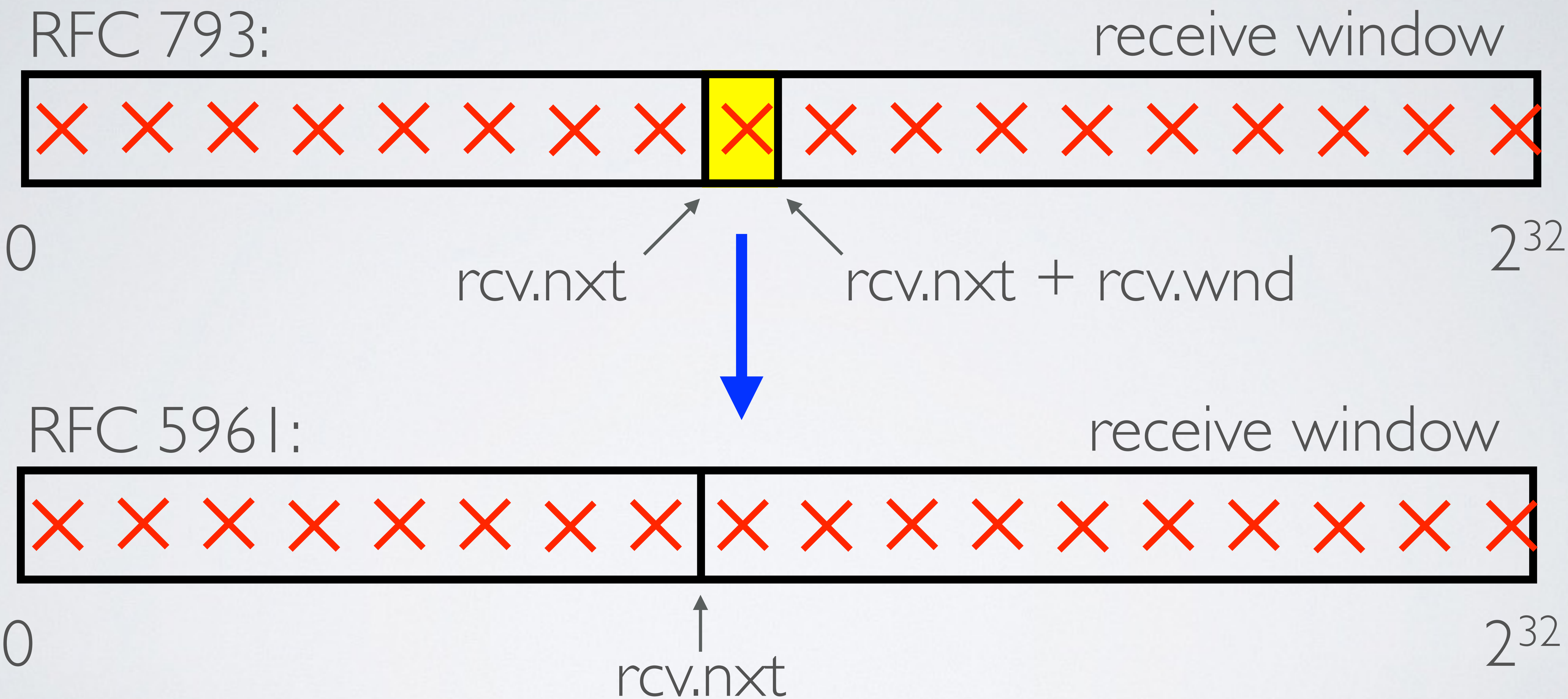
acceptable acknowledgement values have a range of 2^{31} values, so only twice as hard as RST/SYN attacks

Defenses

- **Choose ephemeral ports randomly!** IETF BCP 156 (2011)
- Generalized TTL Security Mechanism (GTSM) } **BGP**
- TCP MD5 and Authentication Options } **BGP**
- Discard packets with spoofed source IP addresses at origin
- RFC 5961, August 2010:
  - strictly validate (challenge) the sequence number in RST and SYN packets
  - reduce range of valid acknowledgement numbers in Data packets

RFC 5961 defenses: RST

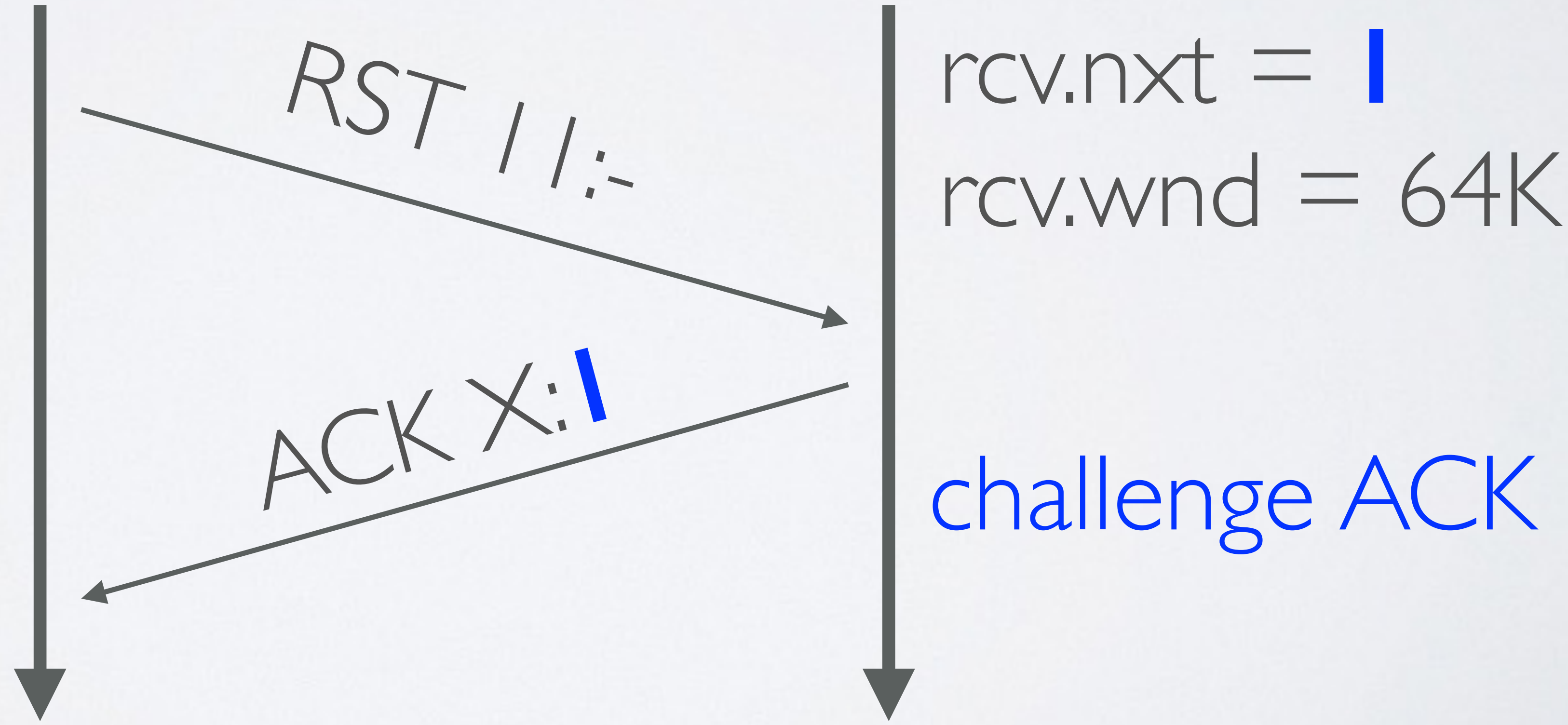
a reset is valid if the sequence number is exactly the next expected sequence number



Difficulty increased to 2^{31} attempts (on average)

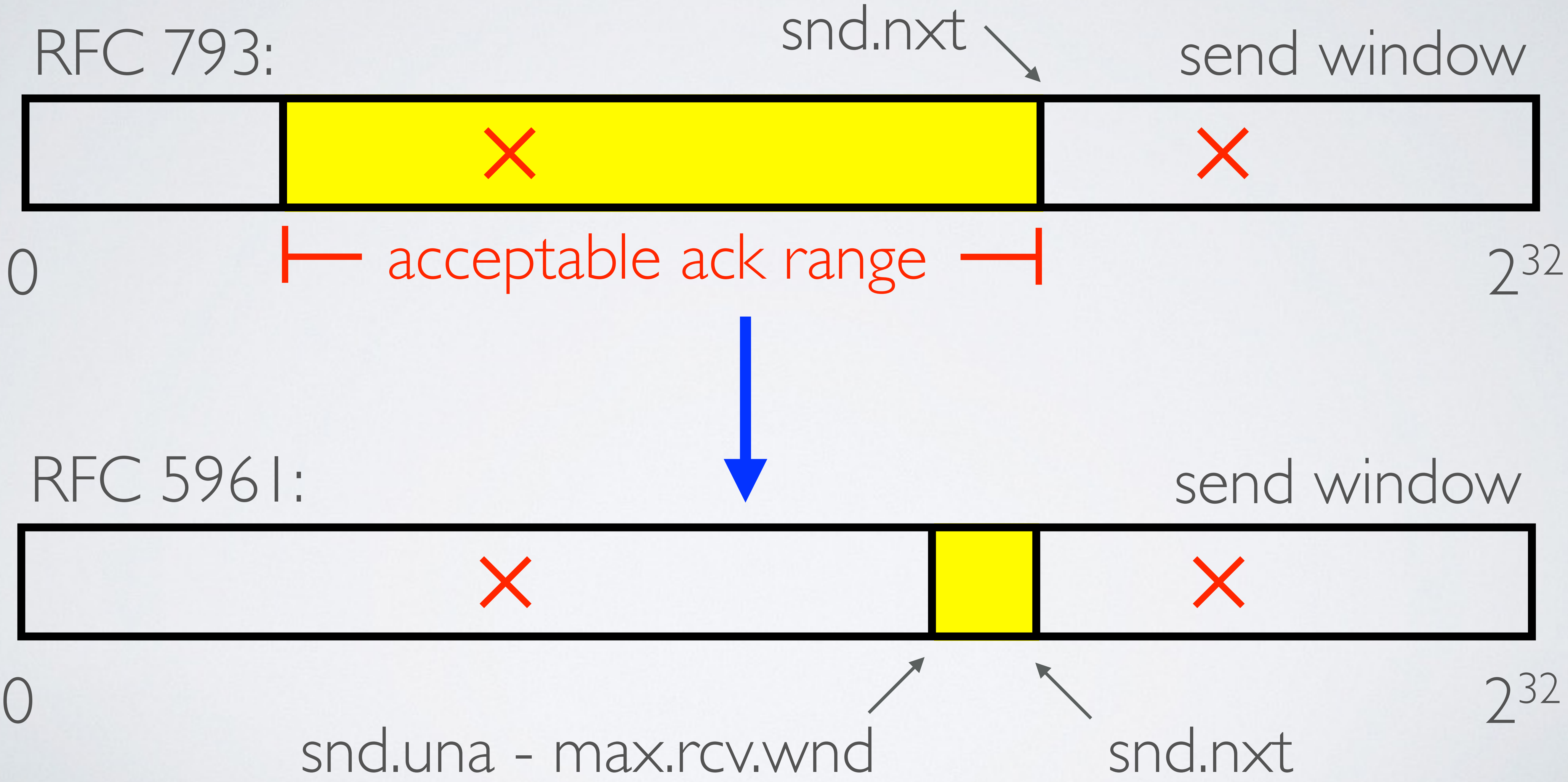
RFC 5961 defenses: RST or SYN

- **RST**: If the sequence number in a RST is in the window, receiver MUST send a **challenge ACK**
- **SYN:** Regardless of sequence number, send a **challenge ACK**
- **Challenge ACK purpose:** to elicit a reset with exact sequence number and confirm loss of connection



RFC 5961 defenses: Data

an acknowledgement number must fall in a smaller range



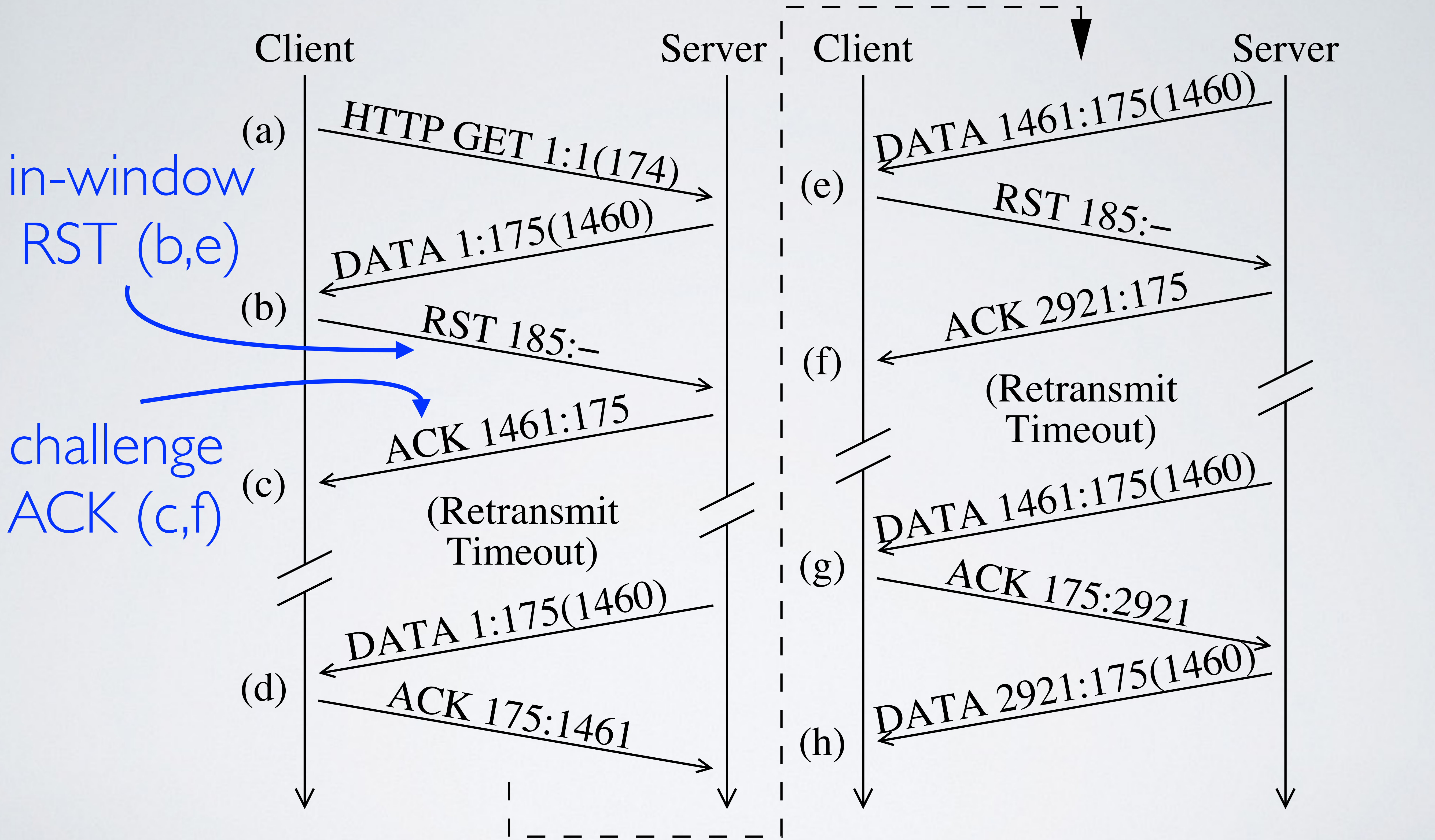
What did we do?

- We implemented and used an oracle-based approach to test **RFC 5961** support
  - Popular web-servers as a proxy for deployed TCP behavior of **general purpose operating systems and middleboxes**
  - Laboratory test of **BGP routers and SDN switches**
  - We tested sequence numbers **in (+10)** and **out (-70,000)** of receive window (Reset + SYN attacks)
  - We tested acknowledgement numbers **behind (-70,000)** and **ahead (+70,000)** of send window (Data attack)
- Evaluated **range and strategy** of OS ephemeral port selection:
  - Bro logs of communications to **ICSI hosts 2005-2015**
  - **March 2015 Tier-1 backbone** link packet trace

What did we find?

- September 2015, tested webservers:
  - 22% were vulnerable to blind reset and SYN packets
  - 30% were vulnerable to blind data packets
  - 38.4% were vulnerable to at least one attack vector
- Laboratory testing of 14 routers and switches
  - 12 were vulnerable to at least one attack vector (mostly blind data attack) that could impact BGP / SDN
- March 2015, 1 hour packet trace: most ephemeral ports were selected in a small range, 50% of predictable in a 2K range.
- 2005-2015: observed some evidence of an increase in ephemeral port range deployment

Testing resilience to blind reset attacks



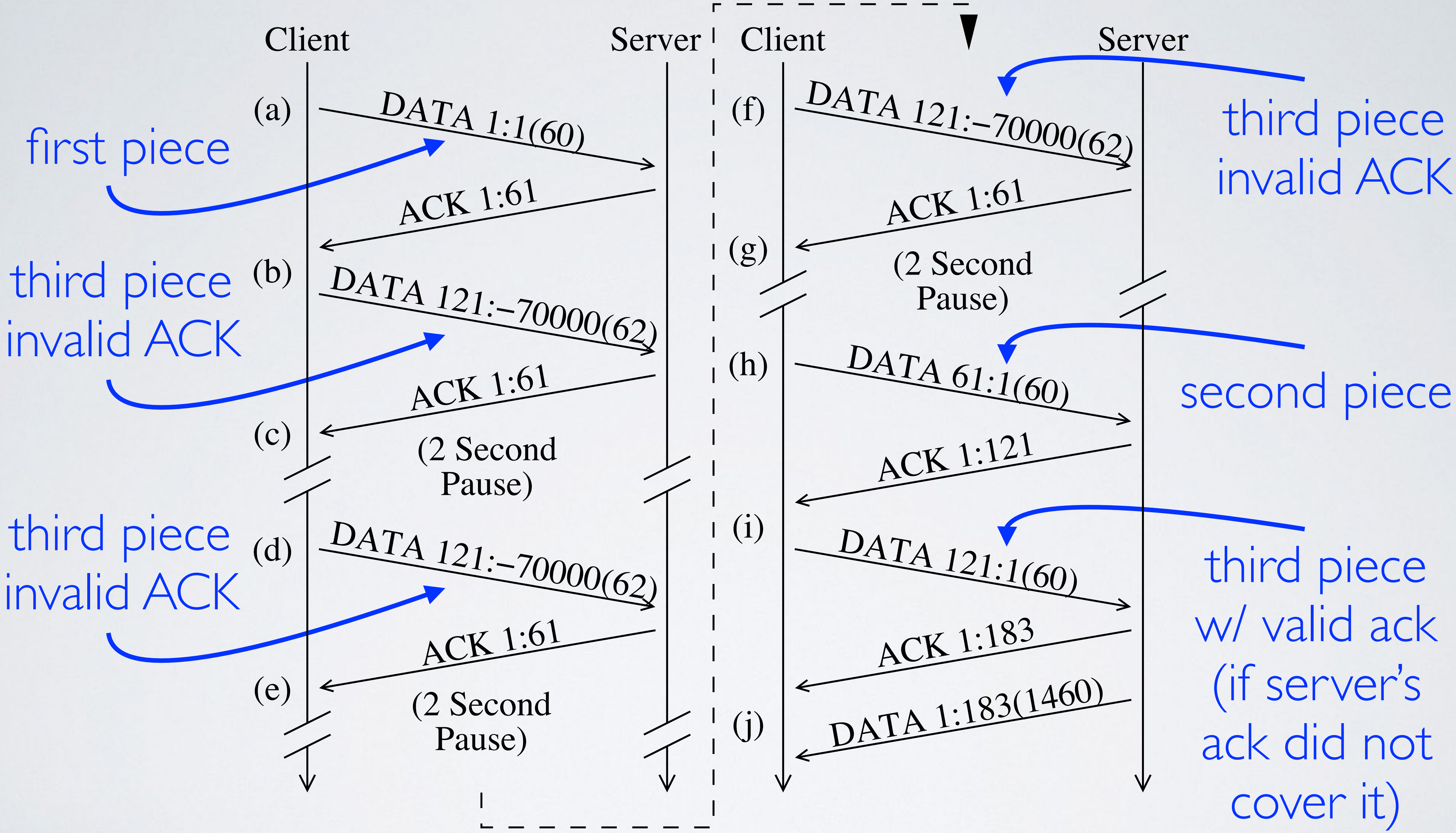
This example shows RFC 5961 compliance

Blind reset and SYN results summary

Testing ~41K webservers, randomly selected from Alexa 1M

Result	Blind Reset		Blind SYN	
	in	out	in	out
Accepted	3.4%	0.4%	—	—
Reset (ack)	—	—	17.1%	0.0%
Reset (dup-ack)	18.8%	0.6%	5.3%	1.2%
Vulnerable	**22.2%**	**1.0%**	**22.4%**	**1.2%**
Challenge ACK	71.4%	1.1%	37.7%	57.0%
Ignored	5.1%	91.8%	35.9%	38.3%
Not Vulnerable	**76.5%**	**93.0%**	**73.6%**	**95.3%**
Parallel connection	—	—	1.1%	1.1%
Early FIN	0.3%	3.3%	1.5%	1.6%
No Result	1.0%	2.7%	1.3%	0.9%
Other	**1.3%**	**6.0%**	**4.0%**	**3.6%**

Testing resilience to blind data attacks



Broke initial request into three pieces; sent third piece second with invalid acknowledgment

Blind Data results summary

Testing ~41K webservers, randomly selected from Alexa 1M

Result	Blind Data	
	behind	ahead
Accepted	29.6%	5.4%
Reset (ack)	0.6%	0.6%
Reset (dup-ack)	0.1%	0.2%
Vulnerable	**30.3%**	**6.2%**
ACK	37.1%	8.1%
Ignored	29.3%	81.3%
Not Vulnerable	**66.4%**	**89.4%**
Parallel connection	—	—
Early FIN	3.2%	3.7%
No Result	0.1%	0.7%
Other	**3.3%**	**4.4%**

5.4% accepted data with an ack value invalid in both RFC 793 and 5961

Evidence of Middlebox protection

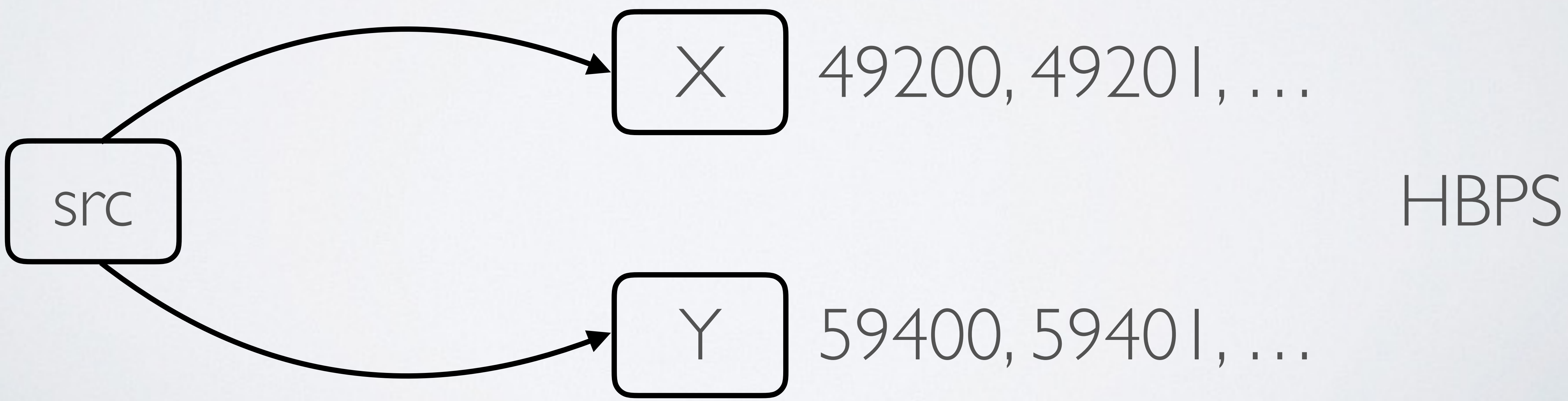
see paper for full details

- TCP connections with an observed MSS of 1380
  - were almost never vulnerable to blind reset and SYN packets, but were vulnerable to blind data packets
  - sent challenge ACKs that arrived with a different TTL than other TCP packets in the flow
  - suggestive of middle-box protection

Ephemeral Port Selection

see paper for full details

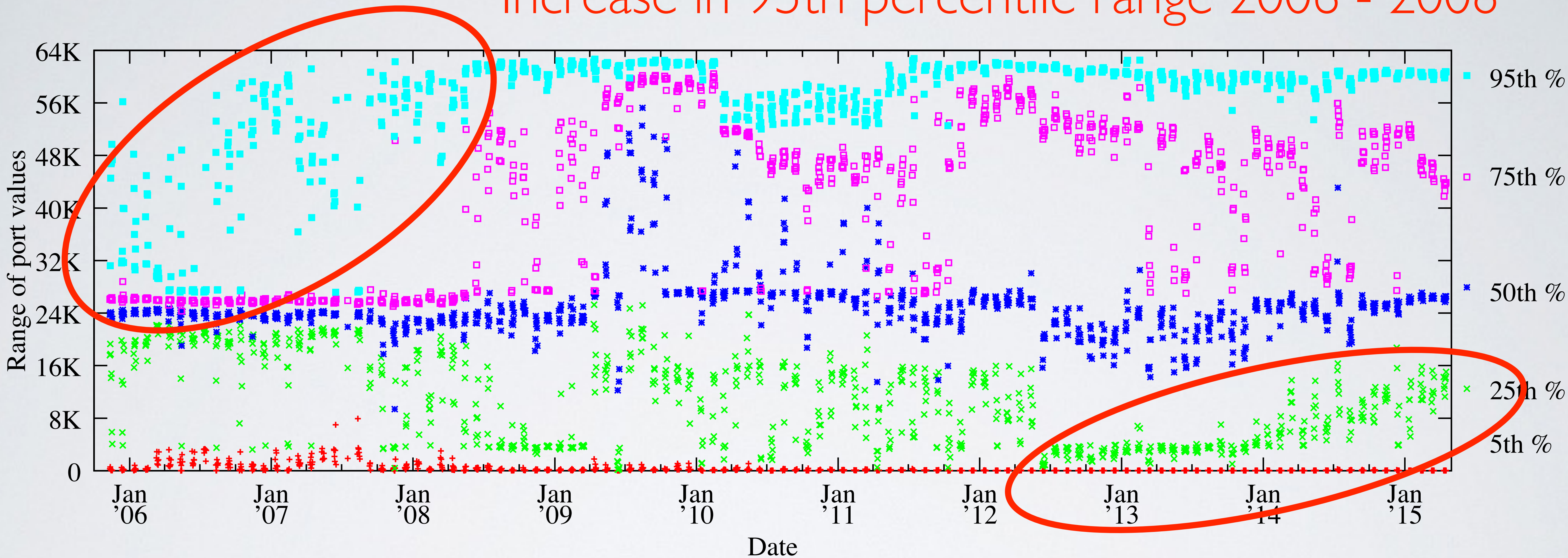
- Goal was to evaluate port selection and range strategies
- Messy problem, no ideal set of data to examine trends with:
  - Packet captures observe subset of traffic from outside hosts
  - Hash-based port-selection (HBPS) could be confused with systems that select ports sequentially.



Ephemeral Port Selection

ICSI Bro Logs

Increase in 95th percentile range 2006 - 2008



Increase in 25th percentile range Oct 2013 - May 2015

Examined ranges of ports chosen over time
(not selection strategy, due to sparseness)

Infrastructure testing results

see paper for full details

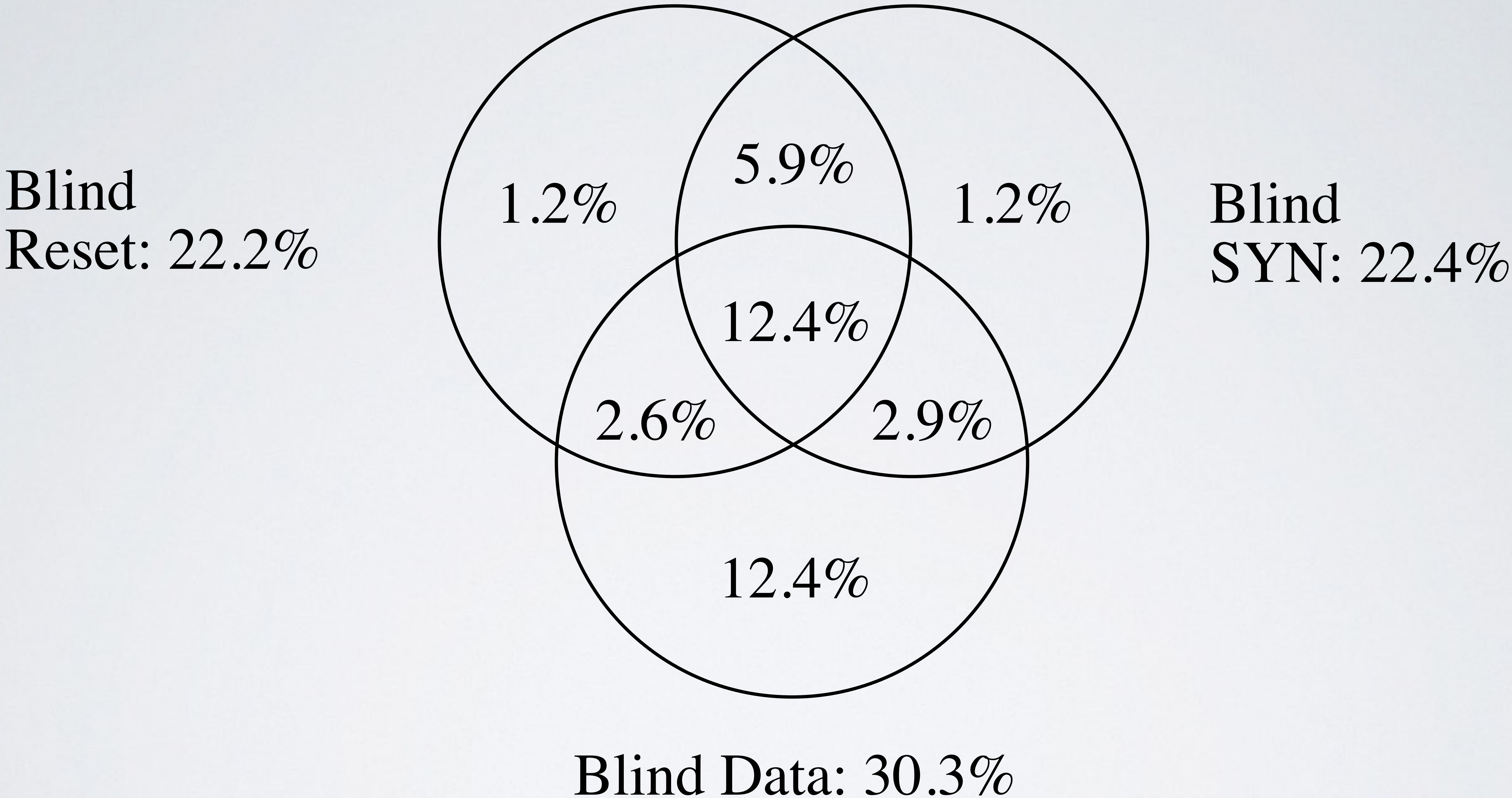
- Tested 14 BGP routers and OpenFlow switches
  - firmwares from 2004 to 2015
  - newer firmware generally does better in both ignoring packets that could have come from a blind attacker, as well as port selection strategies
- 12 were vulnerable to at least one attack
  - data injection attack is currently poorly addressed
- Implication: use GTSM and TCP MD5 where possible

Summary

- Paul Watson 2004 advice: strictly validate RST packets, choose ephemeral ports randomly
- September 2015: 38.3% of tested connections did not use best practices to reject TCP packets that could have come from off-path attacker
- Poor deployment of ephemeral port selection strategies in general population
  - Default behavior of Windows and MacOS is to choose TCP ephemeral ports sequentially
- TBIT tests for resilience to blind attacks available in scamper

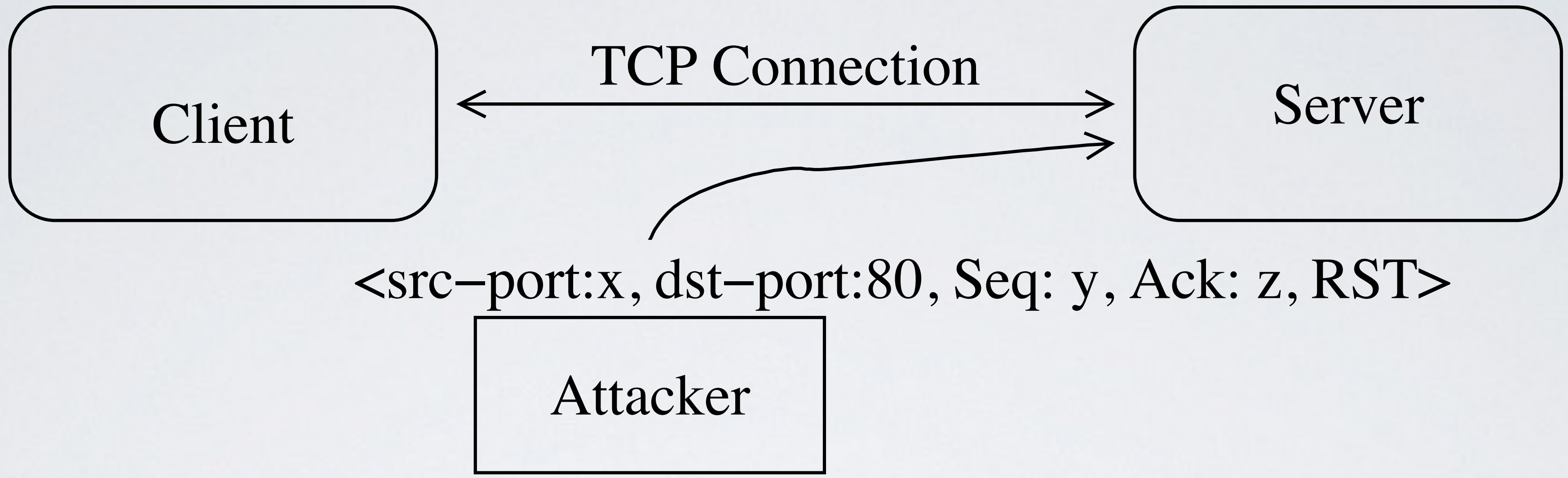
http://www.caida.org/tools/measurement/scamper/

Overlap of vulnerable web servers

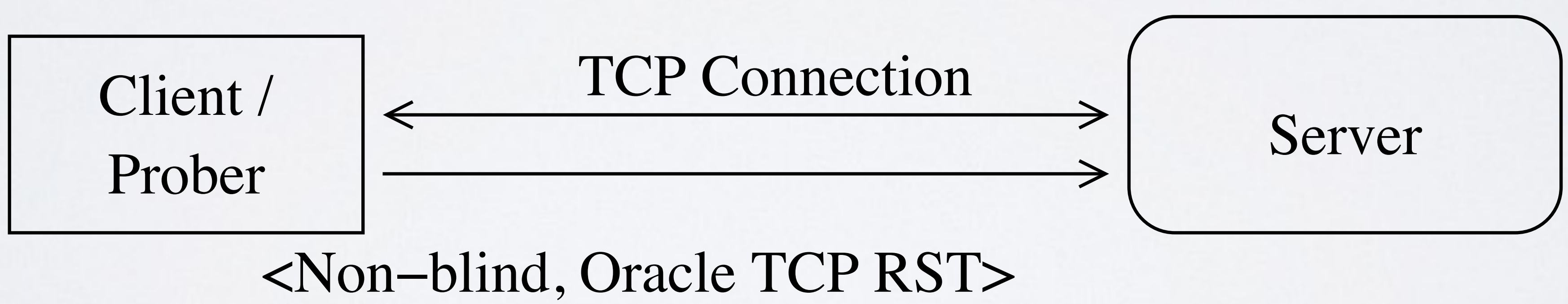


We inferred 38.4% of tested systems to be vulnerable to at least one of the three attacks in September 2015

Oracle vs. Attacker

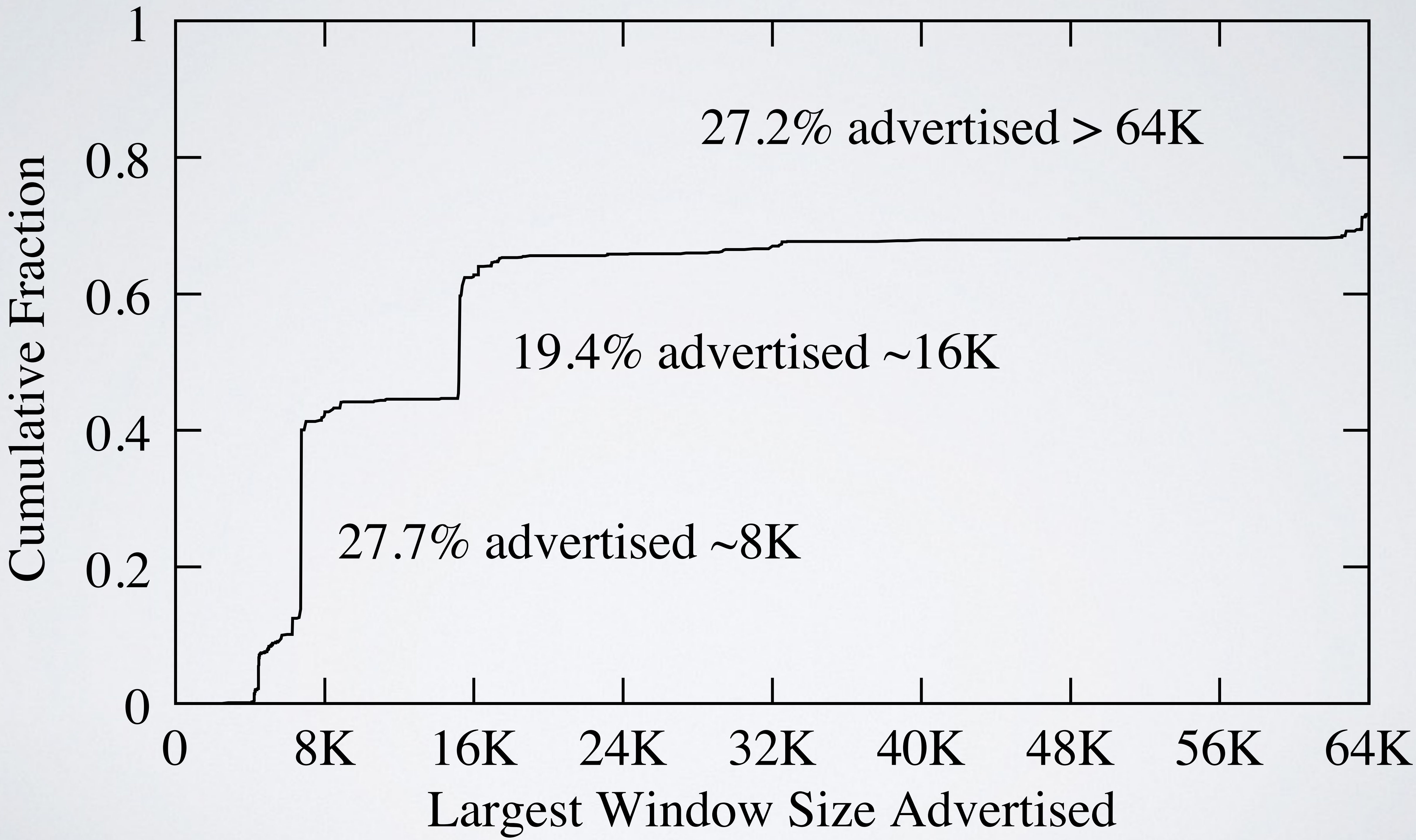


(a) Attacker Approach. **We do not do this.**



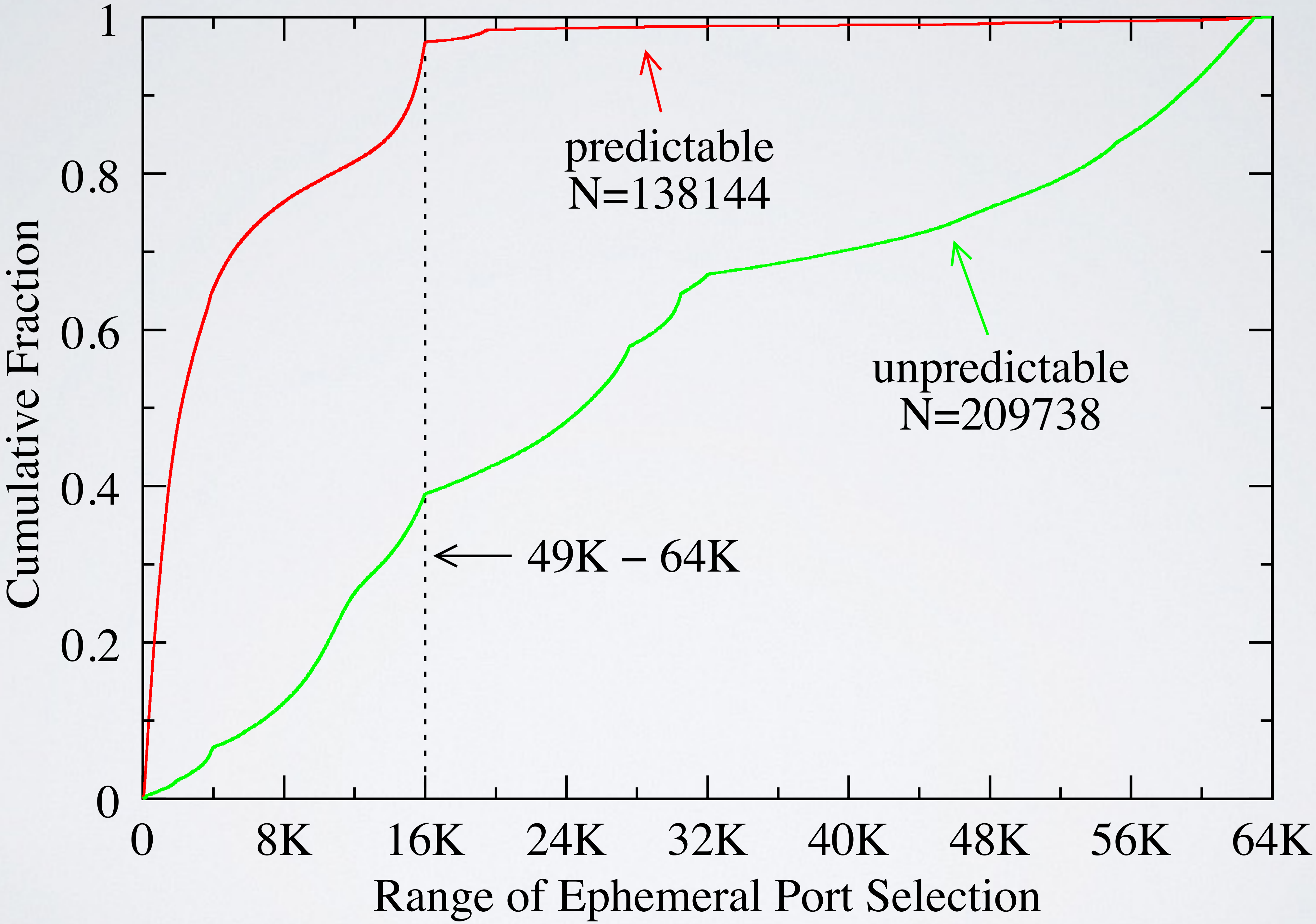
(b) Our Oracle Approach. We establish our own TCP connection and test response to packets that ***could*** have come from an attacker

Largest Observed Window Size for Vulnerable Population



Ephemeral Port Selection

Tier-1 ISP Backbone Link



Ephemeral Port Ranges

Port Range	Size	Operating System
1024-5000	3976	Windows XP and earlier
		FreeBSD <= 4.11 (Jan 2005)
		Linux <= 2.2
49152-65535	16384	FreeBSD >= 5.0 (Jan 2003)
		Windows Vista (Jan 2007)
		Apple MacOS X
		Apple IOS
32768-61000	28232	Linux >= 2.4
10000-65535	55535	FreeBSD >= 8.0 (Nov 2011)

MSS values observed

Server MSS	Vulnerable Portion		
	Blind Reset	Blind SYN	Blind Data
1460 (87.2%)	23.9%	24.7%	28.1%
1380 (5.4%)	2.0%	0.5%	58.8%
8961 (2.3%)	2.3%	2.3%	4.7%
1440 (0.8%)	5.9%	4.7%	57.5%
1436 (0.7%)	22.2%	5.8%	32.5%

Blind attacks by inferred OS (p0f)

Operating System	Blind reset		Blind SYN		Blind data		Total
	in	out	in	out	behind	ahead	
FreeBSD 8.x	19.2%	0.5%	**93.8%**	56.5%	**83.9%**	None	0.5%
FreeBSD 9.x	18.8%	1.0%	**88.1%**	22.2%	54.7%	None	1.5%
Linux 2.4-2.6	**87.4%**	3.0%	**83.6%**	0.4%	54.3%	40.5%	0.6%
Linux 2.6.x	**90.1%**	0.9%	**84.1%**	None	63.2%	35.8%	11.8%
Linux 3.x	15.3%	0.6%	14.0%	0.1%	11.6%	0.6%	43.4%
Windows 7/8	5.1%	2.1%	0.3%	0.3%	**88.7%**	0.9%	9.3%
Windows XP	7.9%	6.1%	3.0%	3.0%	6.3%	3.5%	2.0%
Unknown	9.6%	0.8%	12.7%	12.7%	23.9%	3.2%	30.2%

Blind attacks by router/switch

Device	OS	Blind Reset		Blind SYN		Blind Data	
	date	in	out	in	out	behind	ahead
C 2610	2002-01	✗	✓	✗	✓	✗	✓
C 2610	2002-01	✗	✓	✗	✓	✗	✓
C 2650	2005-08	✓	✓	✓	✓	✗	✓
C 7206	2008-07	✓	✓	✓	✓	✗	✓
C 2811	2010-10	✓	✓	✓	✓	✗	✓
C 2911	2012-03	✓	✓	✓	✓	✗	✓
J M7i	2007-01	✗	✓	✗	✓	✗	✓
J EX9208	2014-06	✓	✓	✓	✓	✗	✓
J MX960	2015-05	✓	✓	✓	✓	✗	✓
J J2350	2015-05	✓	✓	✓	✓	✗	✓
HP 2920	2015-01	✓	✓	✓	✓	✓	✓
HP e3500	2015-06	✗	✓	✗	✓	✓	✓
B MLX-4	2014-10	✓	✓	✓	✓	✓	✓
Pica8	2015-05	✗	✓	✗	✓	✗	✗