

MUST, SHOULD, DON'T CARE: TCP Conformance in the Wild

Mike Kosek, Leo Blöcher, Jan Rüth, Torsten Zimmermann [RWTH Aachen University]
Oliver Hohlfeld [Brandenburg University of Technology]



Yet another TCP study

- TCP in the Wild has been widely analyzed in the past decades
- Stack behavior
 - Tunings, e.g., IW Configuration
 - Extensions, e.g., SACK, ECN, TFO, MPTCP, Fast Retransmit
- Middlebox Interference
 - TCPExposure
 - Tracebox
 - PATHspider
- “TCP is not extendable”

Conformance to minimum requirements?



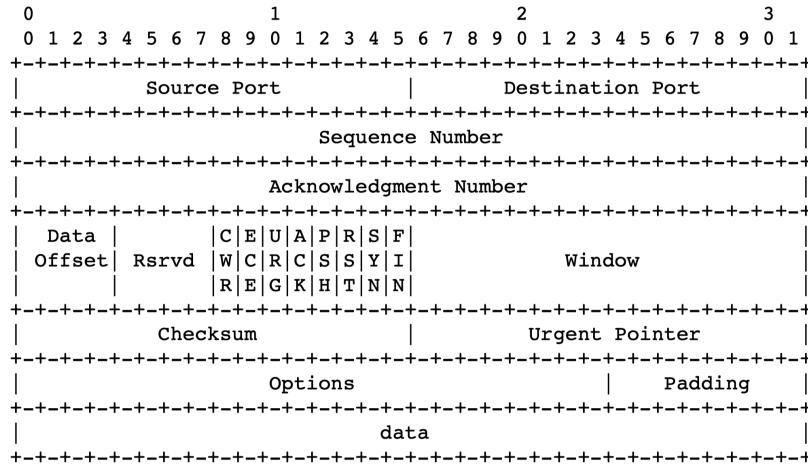
Why SHOULD we care?

- Major efforts are put into QUIC
 - TCP will be the fallback for 10+ years
 - QUIC will not cover all use cases, e.g., BGP
- Active Scanning
 - Controlled Testbed Environment
 - Large scale measurement campaign



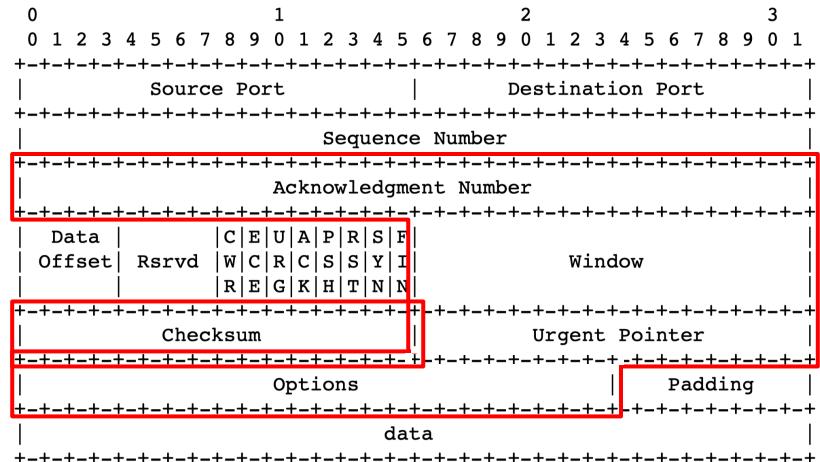
Methodology (1)

- Specifications extend over a multitude of RFCs
- RFC793bis
- What is basic functionality?
 - Requirements for protocol interoperability
- Target Conformance
- Path Conformance



Methodology (2)

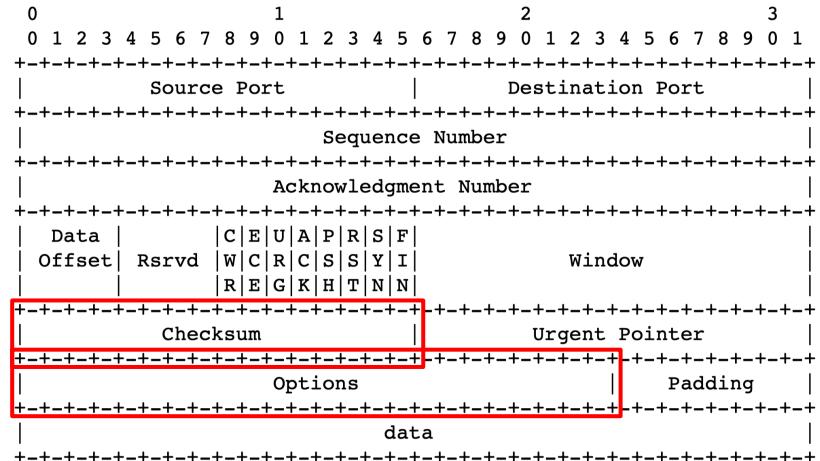
- Middlebox Interference
 - Tracebox approach
 - TTL encoded in multiple fields (e.g., TCP #ACK, Window Size, Urgent Pointer, NOOP Options)
 - Listen for ICMP time exceeded messages
 - Test case specific
- Test cases
 - RFC 793bis-Draft14 features 69 MUSTs
 - Majority addresses internal state handling
 - Requirements must be observable
 - Critical to interoperability, security, performance, or extensibility



Test Cases (1)

- Checksum
 - Computationally expensive
 - Most Layer 2 protocols already protect against segment corruption
 - *When sending a SYN or an ACK segment with an incorrect/zeroed checksum, a target must respond with a RST segment or ignore it.*

- Options
 - Up to 40 bytes of options for future extensibility
 - Most critical to extensibility are unassigned options
 - *When sending a SYN segment with an unassigned option, a target must respond with a SYN/ACK segment.*



Test Cases (2)

MSS Missing

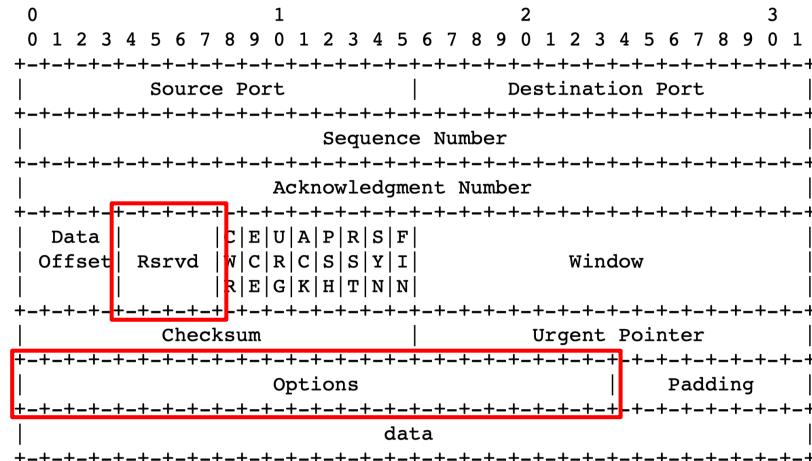
- When sending a SYN segment without an MSS, a target must not send segments exceeding 536 byte (IPv4) or 1220 byte (IPv6).

MSS Support

- When sending a SYN segment with an MSS of 515 byte, a target must not send segments exceeding 515 byte.

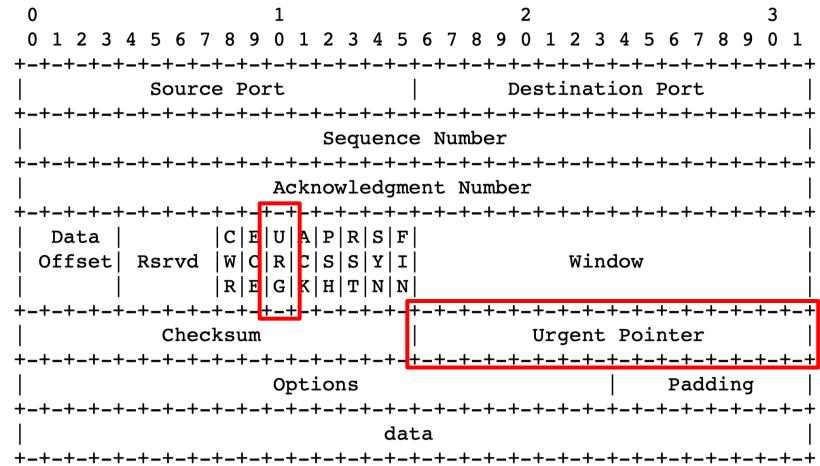
Reserved Flags

- When sending a SYN segment with a reserved flag set, a target must respond with a SYN/ACK segment with zeroed reserved flags.
- Subsequently, when sending an ACK segment with a reserved flag set, a target must not retransmit the SYN/ACK segment.



Test Cases (3)

- **Urgent Pointer**
 - Usage is discouraged for new applications
 - TCP implementations must still include support for arbitrary length
 - *When sending a sequence of segments flagged as urgent, a target must acknowledge them with an ACK segment.*



Controlled Testbed Measurements (1)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>						
<i>ChecksumZero</i>						
<i>OptionUnknown</i>						
<i>MSSMissing</i>						
<i>MSSSupport</i>						
<i>Reserved</i>						
<i>UrgentPointer</i>						



Controlled Testbed Measurements (1)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>	✓				✓	
<i>ChecksumZero</i>	✓				✓	
<i>OptionUnknown</i>	✓				✓	
<i>MSSMissing</i>	✓				✓	
<i>MSSSupport</i>	✓				✓	
<i>Reserved</i>	✓				✓	
<i>UrgentPointer</i>	✓				✓	

- Linux 5.2.10 and lwIP 2.1.2 achieve full conformance



Controlled Testbed Measurements (2)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>	✓	✓			✓	
<i>ChecksumZero</i>	✓	✓			✓	
<i>OptionUnknown</i>	✓	✓			✓	
<i>MSSMissing</i>	✓	✓			✓	
<i>MSSSupport</i>	✓	✗			✓	
<i>Reserved</i>	✓	✓			✓	
<i>UrgentPointer</i>	✓	✓			✓	

- Windows 10 1809 applies the RFC MSS defaults as lower bound



Controlled Testbed Measurements (3)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>	✓	✓	✓		✓	
<i>ChecksumZero</i>	✓	✓	✓		✓	
<i>OptionUnknown</i>	✓	✓	✓		✓	
<i>MSSMissing</i>	✓	✓	✗		✓	
<i>MSSSupport</i>	✓	✗	✓		✓	
<i>Reserved</i>	✓	✓	✓		✓	
<i>UrgentPointer</i>	✓	✓	✓		✓	

- macOS 10.14.6 defaults to 1024 bytes MSS regardless of IP Version



Controlled Testbed Measurements (4)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>	✓	✓	✓		✓	✗
<i>ChecksumZero</i>	✓	✓	✓		✓	✗
<i>OptionUnknown</i>	✓	✓	✓		✓	✓
<i>MSSMissing</i>	✓	✓	✗		✓	✓
<i>MSSSupport</i>	✓	✗	✓		✓	✓
<i>Reserved</i>	✓	✓	✓		✓	✓
<i>UrgentPointer</i>	✓	✓	✓		✓	✓

- Seastar 19.06 bypasses Linux L4 network stack
- Hardware offloading is enabled by default, software checksumming is supported
- Host OS support of offloaded features is not verified

Authors
notified

Controlled Testbed Measurements (5)

	Linux 5.2.10	Windows 1809	macOS 10.14.6	uIP 1.0	lwIP 2.1.2	Seastar 19.06
<i>ChecksumIncorrect</i>	✓	✓	✓	✓	✓	✗
<i>ChecksumZero</i>	✓	✓	✓	✓	✓	✗
<i>OptionUnknown</i>	✓	✓	✓	✓	✓	✓
<i>MSSMissing</i>	✓	✓	✗	✓	✓	✓
<i>MSSSupport</i>	✓	✗	✓	✓	✓	✓
<i>Reserved</i>	✓	✓	✓	✓	✓	✓
<i>UrgentPointer</i>	✓	✓	✓	✗	✓	✓

- uIP 1.0 crashes when receiving urgent data pointing beyond the segment's size
- Contiki-OS and Contiki-NG are also vulnerable

Pull request submitted

TCP Conformance in the Wild – Target Hosts

- **HTTP Archive**
 - CDN tagged URLs
 - Sampled to at most 10k per CDN
 - 27,795 target hosts
- **Alexa 1M**
 - Resolved every domain w and w/o `www.` prefix
 - Randomly selected one target w and w/o `www.` prefix per domain
 - 466,685 target hosts
- **Censys**
 - Internet-wide port scans
 - 3,237,086 target hosts



TCP Conformance in the Wild – Results (1)

	CDN $n = 27,795$			Alexa $n = 466,685$			Censys $n = 3,237,086$		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>									
<i>ChecksumZero</i>									
<i>OptionUnknown</i>									
<i>MSSMissing</i>									
<i>MSSSupport</i>									
<i>Reserved</i>									
<i>Reserved-SYN</i>									
<i>UrgentPointer</i>									



TCP Conformance in the Wild – Results (1)

	CDN $n = 27,795$			Alexa $n = 466,685$			Censys $n = 3,237,086$		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>	0.234	0.374	-	0.441	3.224	0.002	3.743	3.594	0.003
<i>ChecksumZero</i>	0.253	0.377	-	0.455	3.210	0.001	3.873	3.592	0.003
<i>OptionUnknown</i>									
<i>MSSMissing</i>									
<i>MSSSupport</i>									
<i>Reserved</i>									
<i>Reserved-SYN</i>									
<i>UrgentPointer</i>									

- F_{Target} Alexa and Censys

- 1st AS class: ~7% of hosts fail both tests (e.g., Amazon), hinting at purpose build high-performance VMs for, e.g., TCP-terminating proxies
- 2nd AS class: Nearly all hosts fail both tests (e.g., QRATOR AS), hinting at purpose build stack for DDoS protection



TCP Conformance in the Wild – Results (2)

	CDN $n = 27,795$			Alexa $n = 466,685$			Censys $n = 3,237,086$		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>	0.234	0.374	-	0.441	3.224	0.002	3.743	3.594	0.003
<i>ChecksumZero</i>	0.253	0.377	-	0.455	3.210	0.001	3.873	3.592	0.003
<i>OptionUnknown</i>	-	0.026	0.011	-	0.585	0.053	-	1.477	0.019
<i>MSSMissing</i>	0.026	-	0.018	0.303	0.299	0.136	1.423	0.388	0.416
<i>MSSSupport</i>	-	0.018	-	-	0.728	0.002	-	0.412	0.004
<i>Reserved</i>									
<i>Reserved-SYN</i>									
<i>UrgentPointer</i>									

- **Option Unknown**
 - No single AS stands out, highest failure rates are within ISP networks
- **MSS Missing**
 - Censys F_{Path} are primarily located in ISP networks
 - MSS is inserted, likely due to PPPoE encapsulation by access routers

TCP Conformance in the Wild – Results (3)

	CDN <i>n</i> = 27,795			Alexa <i>n</i> = 466,685			Censys <i>n</i> = 3,237,086		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>	0.234	0.374	-	0.441	3.224	0.002	3.743	3.594	0.003
<i>ChecksumZero</i>	0.253	0.377	-	0.455	3.210	0.001	3.873	3.592	0.003
<i>OptionUnknown</i>	-	0.026	0.011	-	0.585	0.053	-	1.477	0.019
<i>MSSMissing</i>	0.026	-	0.018	0.303	0.299	0.136	1.423	0.388	0.416
<i>MSSSupport</i>	-	0.018	-	-	0.728	0.002	-	0.412	0.004
<i>Reserved</i>	-	2.194	0.011	-	6.689	0.293	-	2.791	0.048
<i>Reserved-SYN</i>									
<i>UrgentPointer</i>									

- High F_{Target} across all datasets
 - No response to our probing packets
 - 10% of targeted Akamai hosts on CDN failed
 - Flags on probing SYN were correctly ignored
 - Tests failed on probing ACK by retransmitting the SYN/ACK → TCP_DEFER_ACCEPT

TCP Conformance in the Wild – Results (4)

	CDN <i>n</i> = 27,795			Alexa <i>n</i> = 466,685			Censys <i>n</i> = 3,237,086		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>	0.234	0.374	-	0.441	3.224	0.002	3.743	3.594	0.003
<i>ChecksumZero</i>	0.253	0.377	-	0.455	3.210	0.001	3.873	3.592	0.003
<i>OptionUnknown</i>	-	0.026	0.011	-	0.585	0.053	-	1.477	0.019
<i>MSSMissing</i>	0.026	-	0.018	0.303	0.299	0.136	1.423	0.388	0.416
<i>MSSSupport</i>	-	0.018	-	-	0.728	0.002	-	0.412	0.004
<i>Reserved</i>	-	2.194	0.011	-	6.689	0.293	-	2.791	0.048
<i>Reserved-SYN</i>	-	0.138	0.011	-	1.297	0.309	-	1.849	0.049
<i>UrgentPointer</i>									

- Reserved-SYN
 - Extendibility is limited
- Recap: No formal MUST requirement
 - Started a discussion within the IETF to add a formal MUST
 - Proposed a new MUST requirement to remove ambiguities regarding Reserved Flags

Connectivity
IS impaired

TCP Conformance in the Wild – Results (5)

	CDN <i>n</i> = 27,795			Alexa <i>n</i> = 466,685			Censys <i>n</i> = 3,237,086		
	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}	UNK	F _{Target}	F _{Path}
<i>ChecksumIncorrect</i>	0.234	0.374	-	0.441	3.224	0.002	3.743	3.594	0.003
<i>ChecksumZero</i>	0.253	0.377	-	0.455	3.210	0.001	3.873	3.592	0.003
<i>OptionUnknown</i>	-	0.026	0.011	-	0.585	0.053	-	1.477	0.019
<i>MSSMissing</i>	0.026	-	0.018	0.303	0.299	0.136	1.423	0.388	0.416
<i>MSSSupport</i>	-	0.018	-	-	0.728	0.002	-	0.412	0.004
<i>Reserved</i>	-	2.194	0.011	-	6.689	0.293	-	2.791	0.048
<i>Reserved-SYN</i>	-	0.138	0.011	-	1.297	0.309	-	1.849	0.049
<i>UrgentPointer</i>	0.150	0.330	0.022	0.804	3.179	0.208	3.815	7.300	0.042

- F_{Target} Censys
 - Primarily located in ISP networks
 - 98.8% of failures silently discarded the data
- Recap: Usage is discouraged, but implementation is mandatory
 - We posit to remove the mandatory implementation requirement to reflect its deprecation

Connectivity
IS impaired

Conclusion

- Only two out of six TCP stacks are fully conformant
 - Found and fixed/reported implementation bugs
- A multitude of Internet hosts and paths do not adhere to even basic requirements
- TCP options show the highest level of conformance
 - Access routers in ISP networks are problematic
- Using Reserved Flags or setting the Urgent Pointer can limit connectivity

Conformance to mandatory features
should not be taken for granted



Outlook

- Lessons learned
- Extension to multiple VPs
 - Higher path diversity
- Mobile VPs
 - TCP terminating proxies are deployed on the edge
- TCP Stack settings
 - How do socket options manifest on the wire
- Checksumming
 - Stone, J., Partridge, C.: When the CRC and TCP Checksum Disagree, SIGCOMM 2000
 - 20 years later: Do assumptions hold true in todays systems?



Thanks

Paper



Dataset



References (1)

1. Contiki-NG TCP URG Pull Request. <https://github.com/contiki-ng/contikiling/pull/1173>
2. Contiki-NG: The OS for Next Generation IoT Devices. <https://github.com/contiki-ng>
3. Contiki OS. <https://github.com/contiki-os>
4. Cowboyku, <https://github.com/heroku/cowboyku>
5. Dataset to "MUST, SHOULD, DON'T CARE: TCP Conformance in the Wild". <https://doi.org/10.18154/RWTH-2020-00809>
6. Heroku platform, <https://www.heroku.com/>
7. lwIP - A Lightweight TCP/IP stack. <http://savannah.nongnu.org/projects/lwip/>
8. Seastar. <https://github.com/scylladb/seastar>
9. Seastar: Virtio device reports features not supported by the OS. <https://github.com/scylladb/seastar/issues/719>
10. tcp(7) - linux man page, <https://linux.die.net/man/7/tcp>
11. TCPM Mailinglist: RFC793bis draft 14 Reserved Bits: Problem statement. <https://mailarchive.ietf.org/arch/msg/tcpm/s0LtY3Ce3QBBakJDuSH5VDNFMV>
12. TCPM Mailinglist: RFC793bis draft 14 Reserved Bits: Proposal. <https://mailarchive.ietf.org/arch/msg/tcpm/jpUQx0AjByR3UOgyX88RWoTxLO>
13. uIP. <https://github.com/adamdunkels/uip>
14. Vegur: Http proxy library, <https://github.com/heroku/vegur>
15. Virtio: Paravirtualized drivers for kvm/Linux. <https://www.linux-kvm.org/page/Virtio>
16. Alashwali, E.S., Szalachowski, P., Martin, A.: Does "www." Mean Better Transport Layer Security? In: ACM International Conference on Availability, Reliability and Security (ARES) (2019). <https://doi.org/10.1145/3339252.3339277>
17. Alexa Internet: About us, <https://www.alexa.com/about>
18. Bauer, S., Beverly, R., Berger, A.: Measuring the state of ECN readiness in servers, clients, and routers. In: ACM Internet Measurement Conference (IMC) (2011). <https://doi.org/10.1145/2068816.2068833>
19. Beverly, R.: A Robust Classifier for Passive TCP/IP Fingerprinting. In: Passive and Active Measurement Conference (PAM) (2004). https://doi.org/10.1007/978-3540-24668-8_16
20. Bradner, S.O.: Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Mar 1997). <https://doi.org/10.17487/RFC2119>
21. Cardwell, N., Cheng, Y., Brakmo, L., Mathis, M., Raghavan, B., Dukkipati, N., keng Jerry Chu, H., Terzis, A., Herbert, T.: packetdrill: Scriptable Network Stack Testing, from Sockets to Packets. In: USENIX Annual Technical Conference (ATC) (2013), <https://www.usenix.org/conference/atc13/technical-sessions/presentation/cardwell>



References (2)

22. Carpenter, B., Brim, S.: Middleboxes: Taxonomy and issues (2002). <https://doi.org/10.17487/RFC3234>
23. Craven, R., Beverly, R., Allman, M.: A middlebox-cooperative TCP for a non end-to-end internet. In: ACM SIGCOMM (2014). <https://doi.org/10.1145/2619239.2626321>
24. Detal, G., Hesmans, B., Bonaventure, O., Vanaubel, Y., Donnet, B.: Revealing Middlebox Interference with Tracebox. In: ACM Internet Measurement Conference (IMC) (2013). <https://doi.org/10.1145/2504730.2504757>
25. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A Search Engine Backed by Internet-Wide Scanning. In: ACM Conference on Computer and Communications Security (CCS) (2015). <https://doi.org/10.1145/2810103.2813703>
26. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: Fast Internet-wide Scanning and Its Security Applications. In: USENIX Security Symposium (2013), <https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric>
27. Eddy, W.: Transmission Control Protocol Specification. Internet-Draft draft-ietf-tcpm-rfc793bis-14, Internet Engineering Task Force (Jul 2019), <https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-rfc793bis-14>, work in Progress
28. Edeline, K., Donnet, B.: A Bottom-Up Investigation of the Transport-Layer Ossification. In: Network Traffic Measurement and Analysis Conference (TMA) (2019). <https://doi.org/10.23919/TMA.2019.8784690>
29. Floyd, S., Ramakrishnan, D.K.K., Black, D.L.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Sep 2001). <https://doi.org/10.17487/RFC3168>
30. Fyodor: Remote os detection via tcp/ip stack fingerprinting. <https://nmap.org/nmap-fingerprinting-article.txt> (1998)
31. Gilligan, R.E., McCann, J., Bound, J., Thomson, S.: Basic Socket Interface Extensions for IPv6. RFC 3493 (Mar 2003). <https://doi.org/10.17487/RFC3493>
32. Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M., Tokuda, H.: Is It Still Possible to Extend TCP? In: ACM Internet Measurement Conference (IMC) (2011). <https://doi.org/10.1145/2068816.2068834>
33. HTTP Archive: About HTTP Archive, <https://httparchive.org/about>
34. Knutson, A., Ramaiah, A., Ramasamy, A.: Tcp option for transparent middlebox negotiation. <https://tools.ietf.org/html/draft-ananth-middisc-tcpopt-02> (2013)
35. Külewind, M., Walter, M., Learmonth, I.R., Trammell, B.: Tracing Internet Path Transparency. In: Network Traffic Measurement and Analysis Conference (TMA) (2018). <https://doi.org/10.23919/TMA.2018.8506532>
36. Külewind, M., Neuner, S., Trammell, B.: On the state of ECN and TCP options on the internet. In: Passive and Active Measurement Conference (PAM) (2013). https://doi.org/10.1007/978-3-642-36516-4_14
37. Langley, A.: Probing the viability of TCP extensions. <http://www.imperialviolin.org/binary/ecntest.pdf> (2008)
38. Mandalari, A.M., Lutu, A., Briscoe, B., Bagnulo, M., Alay, O.: Measuring ECN++: Good News for ++, Bad News for ECN over Mobile. IEEE Communications Magazine 56(3), 180–186 (March 2018). <https://doi.org/10.1109/MCOM.2018.1700739>
39. Mandalari, A.M., Bagnulo, M., Lutu, A.: TCP Fast Open: initial measurements. In: ACM CoNEXT Student Workshop (2015)



References (3)

40. Marinos, I., Watson, R.N., Handley, M.: Network Stack Specialization for Performance. In: ACM SIGCOMM (2014). <https://doi.org/10.1145/2619239.2626311>
41. Marinos, I., Watson, R.N., Handley, M., Stewart, R.R.: Disk, Crypt, Net: Rethinking the Stack for High-performance Video Streaming. In: ACM SIGCOMM (2017). <https://doi.org/10.1145/3098822.3098844>
42. Medina, A., Allman, M., Floyd, S.: Measuring Interactions between Transport Protocols and Middleboxes. In: ACM Internet Measurement Conference (IMC) (2004). <https://doi.org/10.1145/1028788.1028835>
43. Medina, A., Allman, M., Floyd, S.: Measuring the Evolution of Transport Protocols in the Internet. *SIGCOMM Comput. Commun. Rev.* 35(2), 37–52 (Apr 2005)
44. Paasch, C.: Network support for tcp fast open. Presentation at NANOG 67 (2016)
45. Padhye, J., Floyd, S.: On Inferring TCP Behavior. In: ACM SIGCOMM (2001). <https://doi.org/10.1145/383059.383083>
46. Piraux, M., De Coninck, Q., Bonaventure, O.: Observing the Evolution of QUIC Implementations. In: ACM CoNEXT Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ) (2018). <https://doi.org/10.1145/3284850.3284852>
47. Postel, J.: Transmission Control Protocol. RFC 793 (Sep 1981). <https://doi.org/10.17487/RFC0793>
48. Rüth, J., Hohlfeld, O.: Demystifying TCP Initial Window Configurations of Content Distribution Networks. In: Network Traffic Measurement and Analysis Conference (TMA) (2018). <https://doi.org/10.23919/TMA.2018.8506549>
49. Rüth, J., Bormann, C., Hohlfeld, O.: Large-Scale Scanning of TCP's Initial Window. In: ACM Internet Measurement Conference (IMC) (2017). <https://doi.org/10.1145/3131365.3131370>
50. Rüth, J., Kunze, I., Hohlfeld, O.: TCP's Initial Window — Deployment in the Wild and its Impact on Performance. *IEEE Transactions on Network and Service Management (TNSM)* (2019). <https://doi.org/10.1109/TNSM.2019.2896335>
51. Rüth, J., Zimmermann, T., Hohlfeld, O.: Hidden Treasures — Recycling LargeScale Internet Measurements to Study the Internet's Control Plane. In: Passive and Active Measurement Conference (PAM) (2019). https://doi.org/10.1007/978-3030-15986-3_4
52. Scheitle, Q., Hohlfeld, O., Gamba, J., Jelten, J., Zimmermann, T., Strowes, S.D., Vallina-Rodriguez, N.: A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In: ACM Internet Measurement Conference (IMC) (2018). <https://doi.org/10.1145/3278532.3278574>
53. Smart, M., Malan, G.R., Jahanian, F.: Defeating TCP/IP Stack Fingerprinting. In: USENIX Security Symposium (2000)
54. Stevens, W.R., Thomas, M., Nordmark, E., Jinmei, T.: Advanced Sockets Application Program Interface (API) for IPv6. RFC 3542 (Jun 2003). <https://doi.org/10.17487/RFC3542>
55. Stone, J., Partridge, C.: When the CRC and TCP Checksum Disagree. In: ACM SIGCOMM (2000). <https://doi.org/10.1145/347059.347561>

