On the State of ECN and TCP Options on the Internet*

Mirja Kühlewind[1], Sebastian Neuner[1], and Brian Trammell[2]

[1] Institute of Communication Networks and Computer Engineering (IKR)
University of Stuttgart, Germany
[2] Communication Systems Group, ETH Zürich, Switzerland

Abstract. Explicit Congestion Notification (ECN) is a TCP/IP extension that can avoid packet loss and thus improve network performance. Though standardized in 2001, it is barely used in today's Internet. This study, following on previous active measurement studies over the past decade, shows marked and continued increase in the deployment of ECN-capable servers, and usability of ECN on the majority of paths to such servers. We additionally present new measurements of ECN on IPv6, passive observation of actual ECN usage from flow data, and observations on other congestion-relevant TCP options (SACK, Timestamps and Window Scaling). We further present initial work on burst loss metrics for loss-based congestion control following from our findings.

1 Introduction

Since the initial design of TCP, there have been a number of extensions designed to improve its throughput and congestion control characteristics. Explicit Congestion Notification (ECN) is a TCP/IP extension that allows congestion signaling without packet loss. Though it has been shown to have performance benefits [1] and has been a standard since 2001 [2,3], ECN deployment lags significantly. Initial deployment problems where middleboxes cleared the ECN IP bits or even dropped packets indicating ECN-capability, as well as firewalls that would reset ECN-capable connections [4], led to mistrust of ECN.

In this work, we examine how much this situation has improved, adding another datapoint to a series of active measurements of ECN usage going back a decade. We also measured the usage of three other congestion-control-relevant TCP options: Selective Acknowledgment (SACK) [5], Timestamps (TS), and Window Scale (WS) [6]. SACK allows more precise signaling of loss, TS improves round-trip-time estimation, and WS allows a larger receiver windows.

Our measurement methodology consists of active probing of the ECN-readiness of a large set of popular web-servers (section 3.1) as well as passive measurement of ECN usage from flow data collected on a national-scale research and education network (section 3.2).

* This work is partly funded by ETICS and mPlane, FP7 research projects supported by the EU. Thanks to SWITCH for the flow data used in this study.

M. Roughan and R. Chang (Eds.) PAM 2013, LNCS 7799, pp. 135–144, 2013.

Table 1. ECN implementation status

year	OS	version
2007	Microsoft Windows	Server 2008, 7, Vista
2007	Mac OS X	10.5
2006	Cisco IOS	12.2(8)T
2001	Linux	2.4 (full support)
1999	Linux	2.3 (router support)

Table 2. History of ECN and options deployment

Reference	Date	ECN	SACK	TSOPT
Medina ea. [7]	2000	1.1%	28%	-
Medina ea. [7]	2004	2.1%	68%	30%
Langley ea. [8]	2008	1.06%	-	-
Bauer ea. [9]	2011	17.2%	-	-

Deployment of ECN and related TCP options has been periodically studied in the literature over the past decade [7,8,9]; the most relevant results for the present work are summarized in Table 2. Bauer *et al* [9] probed the same set of servers as in the present work, so these results are directly comparable. Also related are measurements on TCP extensibility, which focus on middlebox treatment of packets with TCP options. Here findings vary between 0.17% [8] and 70% [9] of hosts dropping packets with unknown options, and 4–14% of middleboxes dropping such packets [10].

We find a recent acceleration in deployment of ECN-capable servers (section 4.1) and greater ECN support on IPv6-enabled servers (section 4.2). We compare this to actual ECN usage, passively measured from flow data captured from the border of a national-scale network, and find that while ECN is more frequently deployed, it is still seldom used (section 4.3).

In section 5, we define a metric for *burst loss* taking into account the peridic probing of congestion-control algorithms, and show that different types of traffic have different burst loss characteristics. Given the continued lag of ECN usage, we advance this initial work as a way to better understand loss dynamics and its relation to application behavior. Section 6 presents our conclusions.

2 Explicit Congestion Notification (ECN): A Review

ECN allows routers using active queue management (AQM) (e.g., Random Early Detection (RED)) to mark packets in case of congestion instead of dropping them. Two bits in the IP header provide four possible marks: No-ECN (00), Congestion Experienced (CE, 11), and two codepoints for ECN-Capable Transport (ECT(0), 01; and ECT(1), 10). An ECN-capable sender sets ECT(0) or ECT(1), which can be changed to CE by a router to signal congestion.

ECN uses two additional flags in the TCP header: ECN-Echo (ECE) is set on all packets from the receiver back to the sender to signal the arrival of a CE-marked packet until the sender sets Congestion Window Reduced (CWR) to acknowledge the ECE. These flags are also used to negotiate ECN usage: a connection initiator requests ECN by setting ECE and CWR on the initial SYN, and the responder acknowledges by setting ECE on the SYN/ACK. After successfully completing the negotiation, the senders can set an ECT codepoint on all subsequent packets over the connection.

Today, ECN is implemented in most operating systems (see Table 1). However even if enabled by default, it is often in "server mode" only: ECN will be negotiated if requested by a remote node initiating a connection, but connections opened by the node will not attempt to negotiate ECN usage.

3 Measurement Methodology

3.1 Active Probing of Web Servers

We measure ECN-readiness and usage of options by sending a TCP SYN with ECN negotiation and the SACK, TS, and WS options enabled to a target server, immediately closing the connection by sending a FIN. The resulting SYN/ACK responses are captured using `tcpdump` and evaluated offline using `scapy`[1], an open source Python-based framework for manipulation and evaluation of TCP packets. The target servers were selected from the Alexa Top 100,000 webservers list, as resolved by the Google public DNS server. If more than one IP address was resolved, we choose the first under the assumption that all servers operated by one provider have the same configuration.

We implemented a tool, also based on `scapy`, to determine whether ECN is usable on a path to a target. First, it generates a SYN with ECN negotiation. If the target is ECN-capable, it then sends one data segment with the CE codepoint set, and evaluates whether ECE was set on the corresponding ACK.

We evaluated the IP Time-to-Live (TTL) of the response as an estimate of the operating system in use at the target. When the TTL is smaller than 64, we assume Linux/BSD, 128 for Windows, and 255 for Solaris. Moreover, we checked the number of hops to be smaller than 64 based on ICMP traceroute. Anyway, this is not a reliable indication, as the initial TTL is configurable; one conspicuous exception is Google, which generally uses Linux but a TTL of 255.

The measurements were performed on a Linux host located in the University of Stuttgart network, connected via the Baden-Württemberg extended LAN (BelWü) to the DE-CIX Internet exchange in Frankfurt. We also performed these measurements over two German mobile network providers (O2 and Vodafone) and got similar results for both.

3.2 Analysis of Aggregated Flow Data

Though active measurement shows increasing deployment of ECN-ready web servers, this gives no information on the actual use of ECN in the network. To measure this, we examine NetFlow version 9 flow data collected from the border of SWITCH[2], the Swiss national research and education network. This network originates about 2.4M IPv4 addresses (the rough equivalent of a /11), with typical daily traffic volumes on the order of 100 TB, and contains both client machines as well as servers for universities.

[1] `http://www.secdev.org/projects/scapy`
[2] `http://www.switch.ch/`

Table 3. April 25, 2012, 77969 unique hosts (of 93573 responding hosts)

	All		TTL < 64		64 ≤ TTL ≤ 128		TTL > 128	
hosts	77969	(100.00 %)	57610	(73.89 %)	12794	(16.41 %)	7590	(9.73 %)
hosts	77969	(100.00 %)	57610	(100.00 %)	12794	(100.00 %)	7590	(100.00 %)
ECN	19616	(25.16 %)	18954	(32.90 %)	521	(4.07 %)	143	(1.88 %)
SACK	69037	(88.54 %)	52409	(90.97 %)	11506	(89.93 %)	5145	(67.79 %)
TSOPT	65307	(83.76 %)	49667	(86.21 %)	10729	(83.86 %)	4928	(64.93 %)
WSOPT	68419	(87.75 %)	53137	(92.24 %)	10047	(78.53 %)	5258	(69.28 %)

Table 4. August 13, 2012, 77854 unique hosts (of 93756 responding hosts)

	All		TTL < 64		64 ≤ TTL ≤ 128		TTL > 128	
hosts	77854	(100.00 %)	57651	(74.05 %)	12471	(16.02 %)	7769	(9.98 %)
hosts	77854	(100.00 %)	57651	(100.00 %)	12471	(100.00 %)	7769	(100.00 %)
ECN	22948	(29.48 %)	22193	(38.50 %)	616	(4.94 %)	145	(1.87 %)
SACK	69334	(89.06 %)	52783	(91.56 %)	11226	(90.02 %)	5353	(68.90 %)
TSOPT	65220	(83.77 %)	49749	(86.29 %)	10379	(83.23 %)	5112	(65.80 %)
WSOPT	68684	(88.22 %)	53420	(92.66 %)	9846	(78.95 %)	5446	(70.10 %)

Our methodology focuses on counting distinct sources, to give us a number comparable to that produced by active measurements. Our flow data unfortunately does not include the TCP flags used for ECN negotiation[3]; however, it does include the ECN Field in the IP header for the first packet observed in each flow record. Since the first packet in a ECN TCP flow is not ECN-capable, we observe *continued* flows: records created after an existing record for a long-lived flow is exported on active timeout (in the measured data, 300s). These capture the ECN field from mid-flow. So, in a given time interval, we count any source address appearing in at least one continued TCP flow record with either the ECT(0) or ECT(1) codepoint set as an *ECN-capable source*. We note this presents only a lower bound for ECN-capable sources, as it will not count any source which never sends a flow longer than the active timeout.

4 Results

4.1 ECN and TCP Option Deployment

We first measured ECN and TCP option support in web servers in April 2012. As shown in Table 3, 25.16 % of web servers negotiated ECN, a substantial increase over that measured by Bauer [9] using a compatible methodology and comparable set of hosts. We measured again in August 2012 (Table 4) and found a further increase to 29.48 % using the current Alexa list, or 29.35 % using the set of targets probed in April. We presume that operating system upgrades are

[3] While the devices can be *configured* to export ECE and CWR, they are always *exported* as zero, due to apparent implementation faults.

the primary cause of increased ECN deployment, as ECN has been supported by all major OSes only since 2007 (see Table 1).

We find that ECN is still less supported than SACK, TS, and WS, though these latter three show no discernible trend between April and August. We also find that ECN is far better supported on Linux hosts (TTLs less than 64) than on Windows (TTL between 64 and 128) or Solaris (TTL greater than 128)[4].

To validate the start TTL estimates, we checked the path length of the top 10,000 servers to ensure less than 64 hops. The minimum path length was 10 hops, as there are 9 hops within the BelWü network; the median was 17.47 hops, the maximum 29, and the mode 13; further investigations are needed on this last point to check for caching or CDNs in Frankfurt.

With respect to ECN usability on the path, we tested 22487 hosts in August 2012 which had negotiated ECN. Of these, 20441 (90.9 %) sent an ECE in response to an CE. 1846 (8.2 %) replied with an ACK without ECE, and 200 (0.9 %) sent no ACK at all. These 9 % of cases where ECN is not usable represent middleboxes which clear CE, which drop packets with CE set, or implementation errors at the endpoints. Additionally, experiments on two UMTS network showed 100 % ECN support but 0 % ECN feedback; we presume due to an ECN-capable HTTP proxy setup and clearing of CE in the mobile network. In any case, these observations show that middleboxes can still significantly affect the end-to-end use of ECN in the network.

We observed one curiosity in our options measurements: with our latest measurement run in September 2012 (31.2 % ECN-capable), we also probed all servers without ECN or any options, to check general responsiveness. We found 429 more unique hosts responding to a SYN without any TCP extension. 828 out of 78204 unique hosts (1.06 %) attempted to use SACK in the SYN/ACK even if not requested. 294 (0.38 %) similarly attempted to use WS, most of them presumably Windows hosts. None responded with TS or ECT. Moreover, while probing `facebook.com` we observed oscillation in RTT between about 100 ms and 150 ms, with an irregular period on the order of hours. This is indicative of load balancing between data centers on the (US) east and west coasts.

4.2 ECN Deployment on IPv6

We investigated the use of ECN over IPv6, in April and August as well as during the World IPv6 Launch event on 6 June 2012; the results are shown in Table 5. Here we find more support for ECN (47.52 %) than over IPv4, as well as more support for other TCP options, but without a comparable increase over time. There was a significant increase in the proportion of Alexa Top 100,000 web servers supporting IPv6 after World IPv6 launch, though only 2.28% support IPv6 as of August 2012. Most IPv6 servers have been installed within the last two years, so we expect greater ECN support in IPv6: these systems should be more up-to-date than average.

[4] As noted above, Google uses an initial TTL of 255, but disables ECN.

Table 5. ECN and options deployment on IPv6

	IPv4 Aug'12		IPv6 April'12		IPv6 June'12		IPv6 Aug'12	
responding hosts	93573		980		1819		2132	
unique hosts	77854	(100.00 %)	785	(100.00 %)	1075	(100.00 %)	1208	(100.00 %)
ECN	22948	(29.48 %)	370	(47.13 %)	522	(48.56 %)	574	(47.52 %)
SACK	69334	(89.06 %)	733	(93.38 %)	1006	(93.58 %)	1093	(90.48 %)
TSOPT	65220	(83.77 %)	713	(90.83 %)	986	(91.72 %)	1049	(86.84 %)
WSOPT	68684	(88.22 %)	734	(93.50 %)	1011	(94.05 %)	1136	(94.04 %)

4.3 Passive Measurement of ECN Adoption

Using the methodology in section 3.2 we examine data for the full day Wednesday, August 29, 2012, from midnight local time, from four of six border routers. Our results are not particularly surprising: while hosts and devices supporting ECN are seeing increased deployment, we confirm that ECN is mostly not used.

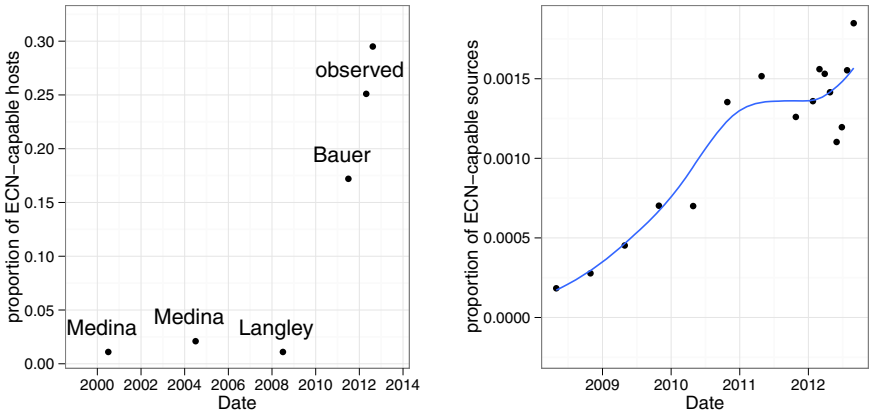
We observed 11,039 total distinct ECN-capable IPv4 sources. This is 0.774% of 1,426,152 distinct sources of continued flows, or 0.161% of 6,837,387 distinct sources observed in all TCP traffic. We estimate the true proportion is somewhere between these measurements. ECN-capable sources were responsible for 1.77TB (3.01%) of 58.84TB of measured TCP traffic.

Of the top 50 ECN-capable sources, there are 19 public-facing web servers, 13 of which appear in the Alexa list used in section 3.1; 12 DHCP clients; 8 servers apparently used for development, testing, or other non-public services; 6 network infrastructure machines, 2 of which are part of an active network performance measurement system; and 5 cloud servers.

Notably, the count of observed ECN-capable sources is on the same order of magnitude as clear errors in ECN usage: 24,580 sources set ECT(0), ECT(1), or CE on a TCP SYN packet. Most of these (16,911 or 68.9%) can be traced to a single ISP which sets the CE codepoint on 99.1% of its outgoing traffic. That there are more sources of persistent misuse of the ECN field from a misconfiguration at a single operator than sources of ECN-capable traffic is a discouraging sign for ECN adoption. We did not observe a single continued flow whose first packet had CE set, other than from sources which set CE on all packets: the extent of use of ECN on routers is too small to measure using this method.

To estimate the historical trend in ECN capability, we count all ECN-capable sources between 13:00 and 14:00 UTC on the last Wednesday of each month on six-month intervals leading up to October 27, 2010, and monthly intervals from January 25 to August 29, 2012[5]. We see a general increase in the proportion of ECN-capable sources, from 0.02% in April 2008 to 0.18% in August 2012. In Figure 1 we compare this trend to our datapoints as measured in section 4.1 as well as to prior measurements summarized in Table 2.

[5] We do not have TCP flags data prior to July 2012; therefore, historical trends detect ECN-capable sources on all flows. This leads to overcounting, as some sources set the ECT bits on the SYN packet as well. We treat these numbers as comparable, as they are all subject to the same overcounting.



(a) Actively measured (b) Passively measured

Fig. 1. Trends in ECN capability

5 Identifying Conditions of Congestion: Burst Loss Study

As ECN usage remains negligible, packet loss remains the only practical signal for congestion control. We therefore turn our attention to loss patterns in typical Internet usage scenarios in order to identify conditions of congestion. This information can be used to improve congestion control or network measurements.

In related work, Allman *et al* [11] showed that 0.6 % of connections experience a loss rate of more than 10 % with a loss of at least 1 packet in more than 50 % of the cases and a loss period, which is a number of losses in a row, of 1 packet in over 60 %. Mellia *et al* [12] measure an average total amount of anomalous segments, including loss and reordering, of 5 % of outgoing traffic and 8 % for incoming traffic on an enterprise network. However, these metrics are given independent of usage pattern and algorithm. Additionally, usage of known TCP congestion control algorithms has been investigated by [13,7,14].

Typically, the loss patterns depend not only on the usage scenario, as congestion control periodically induces overload to probe for available bandwidth. Therefore, the observed loss patterns themselves are also algorithm-dependent. Here we define a *burst loss* as an event consisting of all losses occurring on a TCP connection within one RTT of the first loss; counting these events provides a metric which captures packet loss in a congestion-control-aware way, as losses occurring within a single RTT will be treated as a single event by TCP.

Moreover, application behavior does influence the loss pattern as well. Thus we investigated three common classes of Internet activity – web browsing, download, and YouTube – to study their loss patterns individually. In initial trials we emulated these three types of network traffic on a residential access network with a maximum measured datarate of 5.7 MBit/s: web browsing of 33 common websites with a 12 second delay after each site, viewing of two YouTube videos

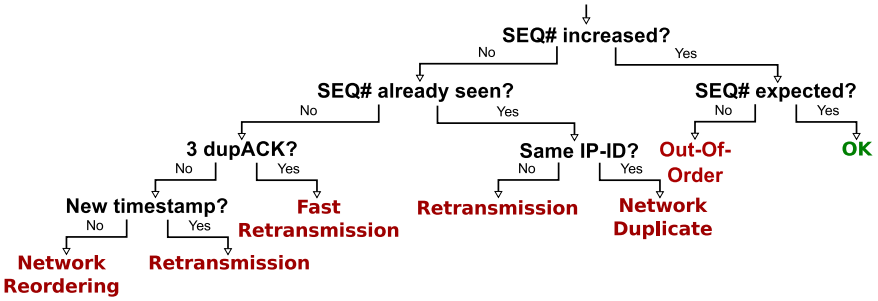


Fig. 2. Decision Diagram for TCP Loss/Retransmission Estimation

(4.62 MB and 11.59 MB), and FTP download of a 80.56 MB file from a host using cubic congestion control. 24 trials were conducted over a single day. The resulting traffic was captured, individual losses or retransmissions were estimated using an algorithm similar to those in the literature [11,15,12]; the decision tree is shown in Figure 2. Losses were then grouped into bursts.

Web browsing consists of many short flows; over all trials, we saw only 5.8% of flows experiencing any loss at all. 82.7% of bursts consists of only a single loss while also bursts of up to 71 losses occurred. The FTP download, on the other hand, involves one single, long flow, and a very regular loss pattern due to congestion control can be observed. As cubic congestion control was used, we observed 70.7 % of single losses as well as frequent bursts of up to 12 losses. In our 24h measurement series we found three probes (at 3am, 10am and 5pm) with a very large number of small burst losses (4058, 3905, and 4157, respectively). Those cases presumably show an anomaly in the network or at the server side. Youtube presents an entirely different pattern, including regular, larger bursts due to its block sending behavior [16] even though YouTube uses TCP congestion control. In 18 of 24 trials, the longer video experienced exactly five bursts, while we always observed one burst for the smaller video. But, given the application behavior, in both cases the mean burst size was around 33. These results are summarized in Table 6.

Table 6. Active measurement from September 10, 2012 [mean number of packets (PKTS), of retransmissions (RET), of burst losses (B), of packets per burst loss (P/B); mean loss rate (RATE); time between burst losses (TBB)]

	PKTS	RET	RATE	B	P/B	TBB
Web-browsing	80779	533.96	0.66 %	227.88	2.37	-
Download (all)	58643	703.04	1.2 %	535	2.10	2.88
Download (21 of 24)	58639	76.14	0.13 %	34.29	2.23	3.28
YouTube1 (11.59 MB)	8469.2	176.29	2.08 %	5.58	31.72	27.31
YouTube1 (18 of 24)	8469.4	159.83	1.89 %	5	31.97	29.40
YouTube2 (4,62 MB)	3386.2	34.04	1 %	1	34.04	-

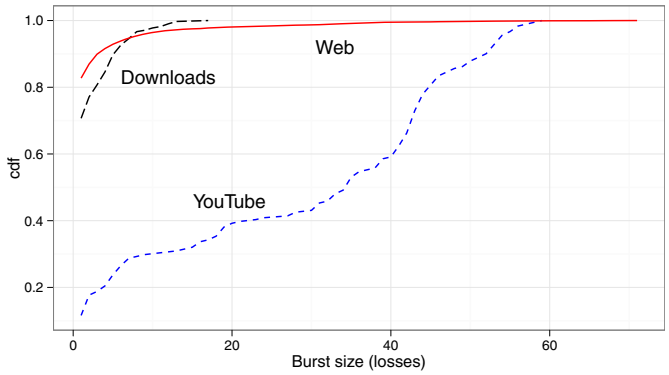


Fig. 3. Burst size (in losses) measured per scenario

These initial findings on loss patterns indicate burst losses to be a well-observable metric. As shown in the distribution of burst loss sizes in Figure 3, the burst loss sizes depend on the scenario. This distribution may therefore provide information to identify the origin of losses, not available with simple metrics such as average loss rate. E.g. a greedy flow using the Reno congestion control algorithm over different network paths (different available bandwidth and RTT) will lead to a different average loss rate but the same pattern in burst size and regularity. Further theoretical or simulation-based work is needed to develop a loss model for different traffic classes and then relate this model to the loss patterns observed in today's Internet to differentiate other sources of losses. Similar influence of congestion control and application behavior can be expected for ECN-based congestion marking, with the additional influence of the AQM at the bottleneck queue.

6 Conclusions and Future Work

This study has shown that deployment of ECN-capable hosts in the Internet continues: about 30% of the top 100,000 web servers can now negotiate ECN usage. We suspect this is due to normal upgrade and replacement cycles affecting the operating systems deployed. Of further interest is that Linux servers are far more likely to support ECN, as are IPv6 servers. Additionally we could measure a general increase in IPv6 support over the IPv6 Launch Day.

While we found 91% of paths to ECN-capable servers are ECN-capable, a failure rate of 9%, including 1% of paths where CE-marked or ECE-marked packets are lost in the network, indicates that earlier problems with ECN deployment are not completely solved. Further, passive measurements give a lower bound for actual ECN usage which was measured to be two orders of magnitude less common than ECN capability. Even worse: twice as many observed sources misused the CE codepoint as properly used the ECT codepoints. Of course, the ECN

readiness on network routers is necessary to realize the full benefits of ECN, as well. This is much more difficult to measure, and thus a problem for future work. Given the difficulty of passive measurement of ECN dynamics, work on the development and deployment of an ECN-aware flow meter is ongoing.

The deployment of ECN would have many benefits, not just for congestion control but for measurement studies of network congestion and traffic engineering, as well. To obtain better information on the conditions of congestion when ECN information is not available, we performed initial studies on the loss pattern of Internet traffic of certain usage scenarios. A broader analysis to understand the effects of congestion control and application behavior observable in the loss pattern resulting in a loss model of today's Internet is underway.

References

1. Salim, J.H., Ahmed, U.: Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks. RFC 2884, IETF (July 2000)
2. Ramakrishnan, K., Floyd, S., Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, IETF (September 2001)
3. Kuzmanovic, A.: The power of explicit congestion notification. SIGCOMM Comput. Commun. Rev. 35(4), 61–72 (2005)
4. Floyd, S.: Inappropriate TCP Resets Considered Harmful. RFC 3360 (Best Current Practice) (August 2002)
5. Mathis, M., Mahdavi, J., Floyd, S., Romanow, A.: TCP Selective Acknowledgement Options. RFC 2018, IETF (October 1996)
6. Jacobson, V., Braden, R., Borman, D.: TCP Extensions for High Performance. RFC 1323, IETF (May 1992)
7. Medina, A., Allman, M., Floyd, S.: Measuring the evolution of transport protocols in the Internet. SIGCOMM Comput. Commun. Rev. 35(2), 37–52 (2005)
8. Langley, A.: Probing the viability of TCP extensions (2008), `http://www.imperialviolet.org/binary/ecntest.pdf`
9. Bauer, S., Beverly, R., Berger, A.: Measuring the state of ECN readiness in servers, clients and routers. In: Proc. of Internet Measurement Conference (2011)
10. Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M., Tokuda, H.: Is it still possible to extend TCP? In: Proc. of IMC 2011, pp. 181–194. ACM, New York (2011)
11. Allman, M., Eddy, W.M., Ostermann, S.: Estimating loss rates with TCP. ACM Performance Evaluation Review 31 (2003)
12. Mellia, M., Meo, M., Muscariello, L., Rossi, D.: Passive analysis of TCP anomalies. Comput. Netw. 52(14), 2663–2676 (2008)
13. Padhye, J., FLoyd, S.: On Inferring TCP Behavior. In: Proceedings of ACM SIGCOMM, pp. 287–298 (2001)
14. Yang, P., Luo, W., Xu, L., Deogun, J., Lu, Y.: TCP Congestion Avoidance Algorithm Identification. In: 31st International Conference on Distributed Computing Systems (ICDCS), pp. 310–321 (June 2011)
15. Benko, P., Veres, A.: A passive method for estimating end-to-end TCP packet loss. In: Global Telecommunications Conference, GLOBECOM 2002, vol. 3, pp. 2609–2613. IEEE (November 2002)
16. Ghobadi, M., Cheng, Y., Jain, A., Mathis, M.: Trickle: Rate limiting YouTube video streaming. In: Proc. of the USENIX Annual Technical Conference (2012)