How to say that you're special: Can we use bits in the IPv4 header?

Runa Barik, Michael Welzl
University of Oslo, Norway

Ahmed Elmokashfi
Simula Research Laboratory, Norway

ABSTRACT

The IP header should be the ideal part of a packet that an end system could use to ask the network for special treatment. Recently, there has been renewed interest in using bits of this header – e.g. the ECN and the DSCP fields. But can we really use these bits? Or should we try to use other bits? We contribute to the body of work that tries to answer these questions by reporting on IPv4 measurements regarding the DSCP field and the Evil bit. Our findings show unexpected treatment to packets that set either of these fields and also confirm recent results on IP Options and ECN.

1. INTRODUCTION

The current Internet is full of middleboxes – devices that performing functions "other than the normal, standard functions of an IP router on the datagram path between a source host and destination host" [3]. For instance, a recent study [16] analyzing 57 enterprise networks revealed that they contain as many middleboxes as routers. The authors of [19] found that 82 out of 107 cellular networks have NAT devices. Measuring what these middleboxes do to packets has been a matter of much recent interest, and it is important, e.g. when designing protocol extensions in the IETF (e.g., [10] had an impact on the design of MPTCP).

However, in-band (per-packet) signaling from end systems to the network should ideally be done in the IP header – the part of the packet that any intermediate device, be it a middlebox or a regular router, *should* be able to analyze and modify. Recent IETF proposals utilize the bits of this header for such purposes – e.g. [5] defines how web browsers should directly set DiffServ Code Point (DSCP) in order to obtain a more suitable service for packets. Another example is the ECN field, which has been overloaded for various purposes (e.g. PCN [6] and ConEx [14]) – recently, it has been suggested to segregate traffic into two different queues depending on the value of this field [2]. The potential difficulty of using the IP header for signaling has also fueled work on other means for in-band signaling between end systems and network, e.g. SPUD [18].

Addressing this need, we present some measurement results that focus on IPv4 header fields, specifically the DSCP and the "Reserved" bit in the IP header – commonly, and in the following, called the "Evil bit" ([1], April 1). While directly setting the DSCP is now being proposed for WebRTC, the Evil bit may also become an opportunity for usage when we run out of available bits.

Our measurements, for which we asked private contacts to run a tool to communicate over raw sockets with our servers, point at some unexpected behavior regarding both DSCP (which can cause packet drops) and Evil bit (which works better end-to-end than the tested DSCP values). Our measurements also roughly confirm some previously published results regarding ECN and IP Options, and show a positive result regarding (mis-)use of Identification (ID) field as a side effect. We elaborate more on this in Sec. 4

2. TEST DESIGN

We implemented a tool based on *scapy* and Python *httplib*; the tool has both client and server side components. The tool executes a pre-specified exchange pattern between the client and the server. For each test, we prepared a packet trace and uploaded it to both the client and the server along with a description of how to exchange these packets. This flexibility allows testing different combinations of flags and options in the IP header.

In May 2016, we carried out a total of 1807 TCP SYN-SYN/ACK handshakes across 185 paths (IP address pairs), using various combinations of IP header flags and IP options. For some tests (e.g. ECN), the handshake was succeeded by an HTTP GET request, followed by an ACK. 35 people in 9 different countries Australia, Austria, Bangladesh, Germany, Norway, Spain, Sweden, Switzerland and United Kingdom installed and ran our scapy-based tool, which carried out several protocol dialogues over raw sockets with our 3 servers (15 hosts only communicated with 2 servers because the test was interrupted). Answering a query from our tool, about two thirds of the users stated that they ran the tests from their homes. One of our servers was based in Oregon (USA), the other two were based in Norway. We intend to do broader tests in the future, including differentiation between mobile and fixed networks.

To minimize the chance that congestion-based drops make us believe in a failure to communicate when using certain values in the IP header, we re-tried failed packet exchanges up to three times, and we sent an ICMP packet just ahead of every measurement packet. We only assumed a communication failure when the test failed three times and the ICMP packet succeeded.

3. RESULTS

Since our DSCP evaluation was motivated by the WebRTC QoS proposal [5], we used this Internet-draft to guide our tests and will discuss our results in its context. Considering table 1 in [5], we assumed the flow type "Interactive Video with or without Audio" with application priorities "Very low" (DSCP value CS1 (8)), "Low"

ANRW '16, July 16 2016, Berlin, Germany
© 2016 ACM. ISBN 978-1-4503-4443-2/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2959424.2959442

Table 1: *DSCP and Evil bit changes by source / destination countries*

DSCP Change	Src. Countries	Dst. Countries	# of paths
DF (0) -> CS1 (8)	Norway (ISP1); Norway (ISP2); Oregon, USA	Austria	5
AF42 (36) -> CS1 (8)			5
EF -> CS1 (8)			5
AF42 (36) -> AF41 (34)	Switzerland	Oregon, USA	2
CS1 -> DF (0)	Many	Many	112
AF42 (36) -> DF (0)			97
EF (46) -> DF (0)			98
Evil bit cleared	Norway (ISP1); Norway (ISP2); Oregon, USA	Switzerland	3
Evil bit cleared	Switzerland	Norway (ISP1); Norway (ISP2); Oregon, USA	3

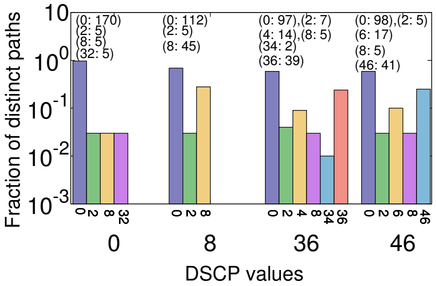

Figure 1: *DSCP value changes. x-axis: the lower (larger) number is the DSCP value that our senders used, the upper (smaller) number is the value that arrived at receivers. The brackets on the top show the absolute number of paths (IP address pairs) along which the change happened.*

(DF (0)) or “Medium” (AF42 (36)), and flow type “Audio” with application prioritiy “High” (EF PHB (46)).

Figure 1 shows the DSCP value changes that we saw. The most expected behavior is that DSCP values pass unchanged or are zeroed. Our results confirm this expectation, as the number of paths where packets were received with their original value (0, 8, 36 or 46) or set to 0 was always largest (and, irrespective of the input value, receiving DSCP=0 seems to be the most common behavior by far). We did, however, see consistent packet drops too: on 23 paths for DSCP value 8, 21 paths for 36, 19 for 46. This failure rate of approximately 10 - 13% is a reason to be concerned about WebRTC QoS; implementations should probably react to consistent failures with a fall-back to DSCP value 0.

On five paths, any DSCP value was changed to 8 – “Very low” in accordance with the table in [5]. Another value that occurred irrespective of the input value was 2, which is undefined [11]. Given the small number of paths, there may only have been a single or a handful of devices that produced these values. Much more interestingly, however, certain DSCP values appeared *only* when the sender applied a nonzero DSCP value. Marking packets as AF42 (36) provoked another undefined value (4) on 14 paths, but also AF41 (34), giving it a lower drop precedence and thereby potentially improving the service. Value 46 (EF PHB), on the other hand, was turned into 6 – yet another undefined value – on 17 paths.

Using the Evil bit provoked consistent packet loss on 11% of all 185 distinct paths – the same approximate range as the DSCP values. Among the successful tests, we observed that the Evil bit was zeroed on 4% of all paths (6 out of 164). This number is much lower than for the DSCP, which was zeroed in 62% of all cases (307 in the total 492 tests of distinct paths per DSCP value, for values 8, 36 and 46). This is perhaps expected, given that the Evil bit has so far been undefined, but it also means that it probably has a better chance to “survive” along a path than the tested DSCP values.

To better understand whether this zeroing and the DSCP value changes (to defined values, which are more interesting because they should also have a defined effect on packets) were done by the same devices, we examined the geographical location of source and destination IP addresses. Table 1 shows that, e.g., the AF42-AF41 change happened for two different source/destination IP addresses pairs between Switzerland and Oregon, USA, and nowhere else, indicating that there was probably only one device in Switzerland that made this change. Similarly, all the changes to CS1 (8) happened on paths to Austria, indicating that there might only have been a single device in Austria that made this change.

4. DISCUSSION AND CONCLUSION

Our measurements have shown some interesting behavior regarding the DSCP and the Evil bit. Perhaps the most important take-away is that using a nonzero DSCP value can provoke consistent packet drops, and hence opportunistically using them as suggested in [5] should come with a fall-back to DSCP 0 in case consistent packet loss is seen. It was also interesting to see that using a nonzero DSCP value can provoke different DSCP value changes than using DSCP 0, potentially leading to different behavior than expected, but also indicating that the DSCP value is indeed understood and reacted upon by the routers in the network.

As for the Evil bit, setting it caused approximately the same amount of consistent packet loss as with the various DSCP values that we tried, but the bit value seemed to have a much better chance to be correctly transmitted across a path. This can indicate that the Evil bit is a better option than the DSCP value for definitions of new behavior (e.g. the proposal in [20]).

There are several other bits and fields in the IP header that deserve a closer look. In particular, the ECN field has been the subject of many investigations (cf. [13] and references therein), and IP Options have also been investigated to some extent (cf. [9, 15]). Our measurements roughly confirm previous findings regarding IP Options: we repeated the tests from [9] but also added the Quick-Start Request [8] and Router Alert [12] IP options, and saw less than 6% of successful tests on distinct paths. For ECN, we repeated a test from [15], which involved sending an HTTP GET request that had the ECN field set to 11 after a successful TCP+ECN handshake. We ended up submitting 108 such GET requests, out of which 91 successfully reached the other side on 69 different paths, i.e. the ECN field being set to 11 caused a drop rate of around 16%.

Contradicting its “allowed” usage [17], our tool used the ID field to enumerate and identify packets of a test (we needed this for other measurements that we carried out in the same campaign). This means that we would categorize both a change of ID field or a drop of the packet as a packet drop in our tests. However, only one out of our 35 total test sources was entirely unable to communicate except for HTTPS signaling, which either points at an extremely restrictive middlebox behavior or failure to forward packets with an ID field value other than 0. Unless routers or middleboxes react to this field differently depending on other fields of the packet, this indicates a very large success rate when trying to send a value in the ID field across the Internet (3599 packets on 185 distinct paths), confirming a finding in [7].

Next, we plan to extend our tool with functionality similar to tracebox [4] such that we can learn the IP addresses of devices that caused packet drops or header changes, and give a better indication of the number of distinct devices that caused a certain behavior.

5. ACKNOWLEDGEMENTS

This work was partially funded by the European Union's Horizon 2020 Research and Innovation Programme through A New, Evolutive API and Transport-Layer Architecture for the Internet (NEAT) project under Grant Agreement no. 644334. The views expressed are solely those of the authors.

6. REFERENCES

[1] S. Bellovin. The Security Flag in the IPv4 Header. RFC 3514 (Informational), Apr. 2003.

[2] B. Briscoe, K. D. Schepper, and I. J. Tsang. Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay. Internet-Draft draft-briscoe-tsvwg-ecn-l4s-id-01, Internet Engineering Task Force, Mar. 2016. Work in Progress.

[3] B. Carpenter and S. Brim. Middleboxes: Taxonomy and Issues. RFC 3234 (Informational), Feb. 2002.

[4] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. Revealing middlebox interference with tracebox. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 1–8, New York, NY, USA, 2013. ACM.

[5] S. Dhesikan, D. Druta, P. Jones, and C. Jennings. DSCP Packet Markings for WebRTC QoS. Internet-Draft draft-ietf-tsvwg-rtcweb-qos-17, Internet Engineering Task Force, May 2016. Work in Progress.

[6] P. Eardley. Pre-Congestion Notification (PCN) Architecture. RFC 5559 (Informational), June 2009.

[7] K. Edeline and B. Donnet. Towards a middlebox policy taxonomy: Path impairments. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 402–407, April 2015.

[8] S. Floyd, M. Allman, A. Jain, and P. Sarolahti. Quick-Start for TCP and IP. RFC 4782 (Experimental), Jan. 2007.

[9] R. Fonseca, G. M. Porter, R. H. Katz, S. Shenker, and I. Stoica. IP options are not an option. Technical report, EECS Department, University of California, Berkeley, 2005.

[10] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda. Is it still possible to extend TCP? In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 181–194, New York, NY, USA, 2011. ACM.

[11] Differentiated Services Field Codepoints (DSCP). http://www.iana.org/assignments/dscp-registry.

[12] D. Katz. IP Router Alert Option. RFC 2113 (Proposed Standard), Feb. 1997. Updated by RFCs 5350, 6398.

[13] M. Kühlewind, S. Neuner, and B. Trammell. On the State of ECN and TCP Options on the Internet. In *Proceedings of the 14th International Conference on Passive and Active Measurement*, PAM'13, pages 135–144, Berlin, Heidelberg, 2013. Springer-Verlag.

[14] M. Mathis and B. Briscoe. Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements. RFC 7713 (Informational), Dec. 2015.

[15] J. Pahdye and S. Floyd. On inferring TCP behavior. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '01, pages 287–298, New York, NY, USA, 2001. ACM.

[16] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 13–24, New York, NY, USA, 2012. ACM.

[17] J. Touch. Updated Specification of the IPv4 ID Field. RFC 6864 (Proposed Standard), Feb. 2013.

[18] B. Trammell and M. Kühlewind. Requirements for the design of a Substrate Protocol for User Datagrams (SPUD). Internet-Draft draft-trammell-spud-req-04, Internet Engineering Task Force, May 2016. Work in Progress.

[19] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. An untold story of middleboxes in cellular networks. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 374–385, New York, NY, USA, 2011. ACM.

[20] J. You, M. Welzl, B. Trammell, M. Kühlewind, and K. Smith. Latency Loss Tradeoff PHB Group. Internet-Draft draft-you-tsvwg-latency-loss-tradeoff-00, Internet Engineering Task Force, Mar. 2016. Work in Progress.