Investigating the Delay Impact of the DiffServ Code Point (DSCP)

Michael Welzl, Safiqul Islam, Runa Barik, Stein Gjessing
Networks and Distributed Systems Group,
Department of Informatics, University of Oslo
{michawe, safiquli, runabk, steing}@ifi.uio.no

Ahmed Elmokashfi
Centre for Resilient Networks and Applications (CRNA)
Simula Metropolitan Center for Digital Engineering,
Oslo, Norway
ahmed@simula.no

***Abstract*—The DiffServ Code Point (DSCP) field in the IP header allows to specify a desired per-hop behavior as packets traverse routers. Setting the DSCP field opportunistically, without prior contractual agreement, has recently become accepted practice for Internet end hosts. Measurement studies find that there is reason to hope for a DSCP setting to have an effect on traffic, and at least configuring this value is not heavily detrimental: systematic drops of packets due to non-zero DSCP values are rare, and the value is often left intact along an end-to-end path. What these studies do not discuss is whether per-hop behaviors truly are honored: what happens to packets in terms of the delay they experience? In this paper, we make an attempt to find a first answer to this by mining a dataset of our own recent large-scale measurement study. Using a deep neural network, we obtain the importance of the factors which help us understand the delay impact of the DSCP.**

***Index Terms*—Fling, DSCP, Machine Learning**

I. Introduction

The DiffServ Code Point in the IP header, originally meant to be set to a nonzero value only when QoS contracts are in place, is now being used as a means for an application to simply indicate a desired behavior without expecting any guarantees (and without having paid for them either). This change was mainly brought about by the WebRTC standards suite, which includes a specification on how the DSCP field should be used to reflect the needs of streams such as interactive audio, video etc [1].

In a previous study [2], we transmitted packets between 225 clients and 52 servers around the world using our flexible ping ("fling") measurement tool[1] [3], in order to see what happens when they carry a non-zero DSCP. Specifically, in addition to DSCP=0, we used three values that are important for WebRTC: *CS1* (low-priority data), *AF42* (multimedia conferencing) and *EF* (telephony). Here, we build upon this study by further analyzing our data set, to try to understand whether the DSCP choice played an important role for the delay that packets experienced.

Measuring the delay impact of the DSCP value is tricky: on the one hand, a measurement study needs to be relatively large in order to be representative. On the other, a DSCP choice is not expected to have any effect whatsoever when the network is uncongested: it is only meant to influence how routers treat packets as they take them from their input queues and schedule them for transmission. However, congesting the Internet at a large scale from many vantage points is not usually a well accepted practice in measurement testbeds (in our case, we used the ARK[2], PLANETLAB[3] and NORNET CORE[4] platforms). Thus, we are left with a choice of: i) doing smaller-scale measurements, or ii) trying to understand if the DSCP value had an effect even in conditions when the network may not have been congested.

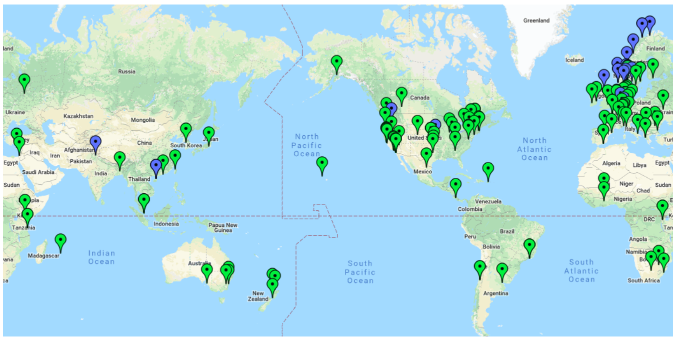

Fig. 1: Geographical *fling* host locations - Green: clients, Blue: servers.

Here, we do the latter. The idea is to understand if it is possible to find that DSCP has an impact on delay. We send small-scale bursts of two or three packets that will be consecutively enqueued when they traverse a router, and see it these packets are systematically treated differently according to their DSCP values. We elaborate on our findings in Section II. Related work is discussed in Section III, and Section IV concludes.

II. Results

As its name suggests, fling is reminiscent of "ping", making it very lightweight and non-intrusive for its users. Thus, a single *fling* DSCP measurement consists of sending just one UDP packet with a specified DSCP value along a path. We tested in both directions and used several values, including the value 0 (which we call the "baseline test"). Also, each of the

[1] *fling*: http://fling-frontend.nntb.no.

[2] ARK: https://www.caida.org/projects/ark.

[3] PLANETLAB: https://www.planet-lab.org.

[4] NORNET: https://www.nntb.no.

978-1-5386-9223-3/19/$31.00 ©2019 IEEE

IP version	forward	backward
IPv4	1954	2080
IPv6	895	1020

TABLE I: The total number of measurements that were used for analysis. “Forward” and “backward” refer to the client => server vs. server => client measurement directions, respectively.

non-zero tests were carried out up to three times in order to compensate for possible packet loss. To avoid being too biased by other factors, we limited our analysis to cases where the path always was the same irrespective of the DSCP value. We note that this eliminates the possibility of discovering that the DSCP value itself has provoked a router to send packets along different paths—however, the number of cases where the path changed was so small that we considered it less interesting to investigate further.

Table I shows the total number of measurements that we were left with, after removing cases where not all DSCP values managed to traverse the path at least once, and removing some outliers that did not seem to lend more meaning to our dataset. We separately consider IPv4 and IPv6 tests, each in the “forward” (client => server) and “backward” (server => client) direction.

After a series of measurements, fling results are collected in a single place, together with all sending and receiving timestamps; this allows us to calculate a One-Way Delay (OWD) value for each measurement. Clearly, because host clocks are not synchronized, the absolute OWD values are not meaningful per se—however, we do not care about the absolute magnitude, but only the impact of the DSCP on the OWD (this is similar to how the OWD is used in LEDBAT [4]). We therefore consider the DSCP=0 measurement as a baseline and only look at OWD differences $\delta_{OWD_x} = OWD_x - OWD_{baseline}$, where x is a specific DSCP value.

Note that the baseline OWD is just one out of several point measurements along a specific Internet path; as such, it could reflect the path’s minimum OWD, or it might just as well be the path’s only OWD value that was measured when the network was congested. The latter case would cause all δ_{OWD} values to be negative. To avoid such misinterpretations we try not to draw conclusions directly from the absolute δ_{OWD} either—instead, we focus on the difference between them (differences between $\delta_{OWD_{CS1}}$, $\delta_{OWD_{AF42}}$ and $\delta_{OWD_{EF}}$).

A. Raw results

Figures 2 and 3 show the distribution of δ_{OWD} values in the forward and backward direction, respectively. We can see that, as expected, the differences between the various DSCP values are generally tiny. In the forward direction, for values around -0.15 to -0.05 in the graph, we can see that the “CS1” line is slightly higher than the others, meaning that, when the DSCP value’s delay was around 50 to 150 ms smaller than the baseline delay, this delay reduction was less pronounced for CS1 which is supposed to indicate low-priority traffic—this appears to indicate routers honoring the code point. However, this is not visible in the backward direction.

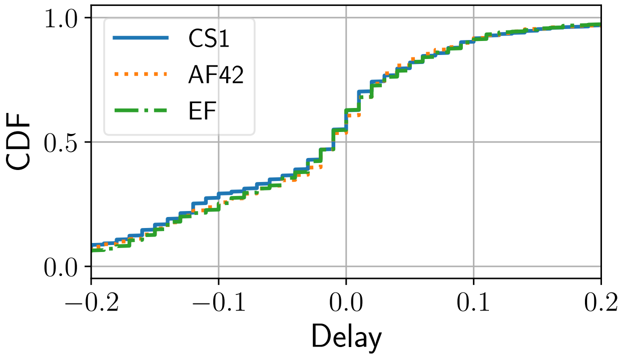

Fig. 2: Cumulative Distribution Function of δ_{OWD} values in the forward direction, IPv4

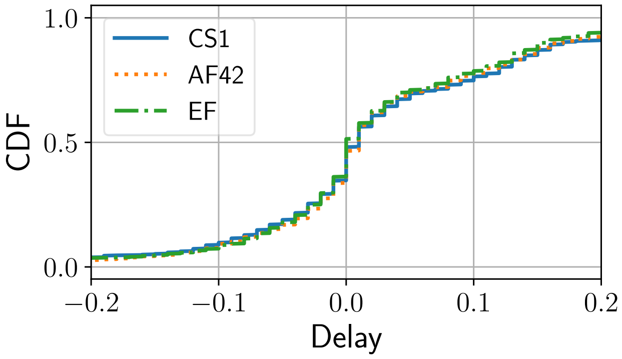

Fig. 3: Cumulative Distribution Function of δ_{OWD} values in the backward direction, IPv4

Things look quite different in the IPv6 case: in the forward direction (Figure 4), we can see that the CS1 line is generally lower than the others, indicating a lower delay for this DSCP choice. This behavior is surprising, and does not appear in the

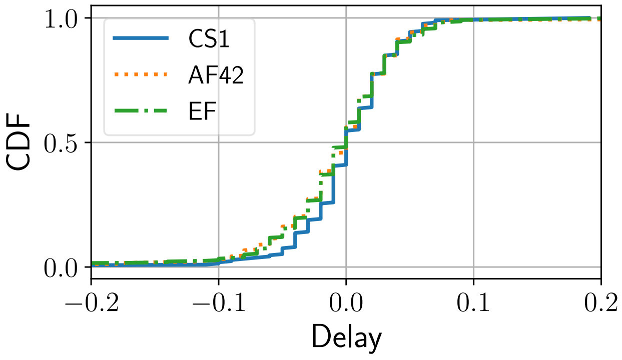

Fig. 4: Cumulative Distribution Function of δ_{OWD} values in the forward direction, IPv6

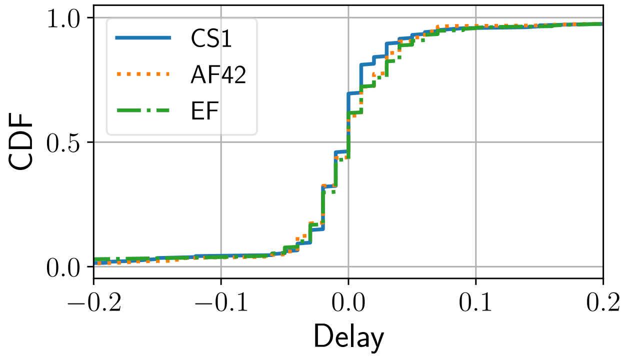

Fig. 5: Cumulative Distribution Function of δ_{OWD} values in the backward direction, IPv6

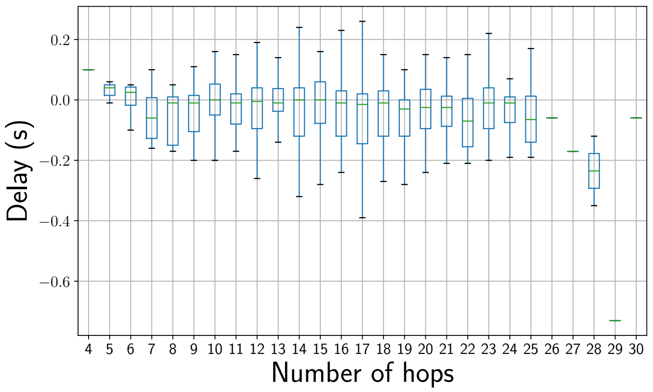

Fig. 6: Boxplot of $\delta_{OWD_{CS1}}$ as a function of the number of hops in the forward direction, IPv4

backward direction (Figure 5), where we can see the CS1 line exceeding the others for some part of the graph.

To conclude, the results are inconsistent: they seem to match the "correct" behavior, to some degree, with IPv4 in the forward direction only, and in the backward direction with IPv6. However, with IPv6 in the forward direction, we seem to have a counterproductive behavior of routers (the CS1 low priority traffic experiencing lower delay than the rest).

To better understand what the reason behind these results may be, we examined the behavior as a function of the number of hops along paths. Here, the idea is that, if multiple routers along the path implement the same DSCP policy, there would be an increasing trend in the resulting δ_{OWD} values as the number of hops increases. We consider $\delta_{OWD_{CS1}}$ because it seemed to stand out as the DSCP value with the most pronounced difference from the others. Figures 6 and 7 show the two most interesting cases from before, where the difference between $\delta_{OWD_{CS1}}$ and the other δ_{OWD} values was most significant: IPv4 and IPv6, each in the forward direction.

From visual inspection of these and other similar diagrams, we cannot see a clear positive or negative trend as a function of the number of hops. This may indicate that DSCP policies were only applied at the access link (a middlebox such as a home router), or that this was the only link experiencing some degree of congestion.

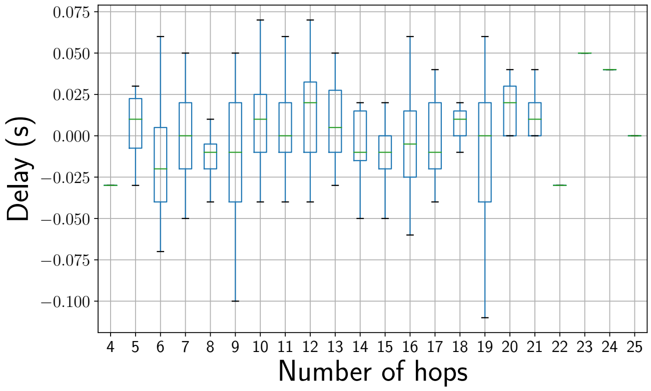

Fig. 7: Boxplot of $\delta_{OWD_{CS1}}$ as a function of the number of hops in the forward direction, IPv6

B. *Interpretation using Machine Learning*

Given the lack of a conclusive relationship between the DSCP value and the difference in OWD from our analysis in the previous subsection, we now use a deep neural network to try to find an answer to the following question: which factors among the following play the biggest role for the difference between a baseline delay and a delay seen when using a specific DSCP value—the DSCP value itself, the number of hops, whether the transmission was from a client to a server or vice versa, where the measurement took place (source and destination IP addresses), or whether IPv4 or IPv6 was used?

To this end, we created a deep neural network (7 layers with 100 neurons per layer, followed by a layer with 1 neuron for the single output value, input dimension 5: the parameters from above except IPv4/IPv6: because of the significantly different behavior that we have earlier seen between IPv4 and IPv6, we decided to investigate these two cases separately) using the Keras library with a TensorFlow backend, to perform a regression analysis. This neural network was trained to predict δ_{OWD}; because the DSCP value itself was used as an input variable, we did not distinguish between the three different types of δ_{OWD} values. Using a batch size of 30, after 3154 epochs, we attained a Root Mean Squared Error (RMSE) of less than 12%[5]. We did not divide the data into a training and test set because our goal was *not* to create a general model: instead, we used the neural network only as a way to *characterize* our data set. This allowed us to apply the connection weight approach from [5] to determine the importance of the input parameters, which can be a better hint about their influence on the output than the graphs that we previously examined. The result is shown in Fig. 8. The derived parameter importance values match our intuition: it shows that the δ_{OWD} mostly depends on the IP addresses and

[5] After carefully tuning the hyper-parameters and the number of neurons in the hidden layers, we have found that the trained neural network with the chosen settings minimizes the RMSE.

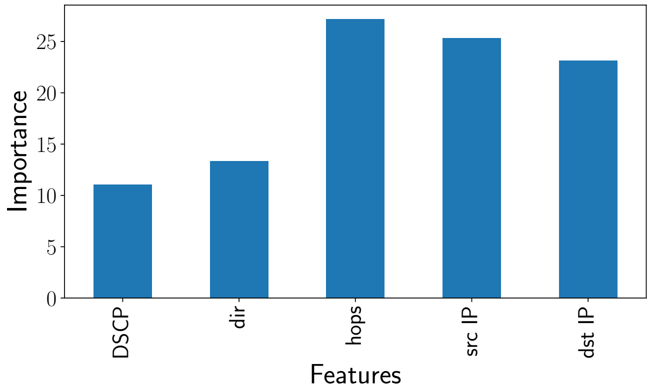

Fig. 8: Importance of the input features, IPv4

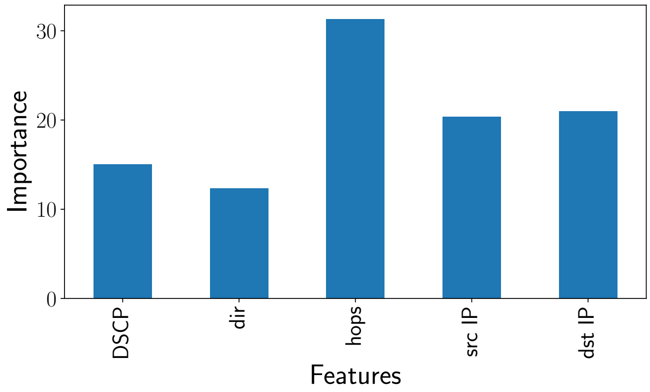

Fig. 9: Importance of the input features, IPv6

the number of hops. However, the DSCP also has a significant importance score (around half of the importance of the number of hops and the IP addresses), which could explain the relative DSCP-dependent differences shown in Fig. 2 and 3.

Fig. 9 highlights the importance of the input features of the IPv6 tests. Given the total number of measurements used for IPv6 is 50% less than the IPv4 measurements, the attained RMSE value (19%, after 4000 iterations) is not as low as IPv4. However, it also matches our intuition: it essentially depends on the number of hops and IP addresses. Moreover, DSCP also has a significant value as a feature (around half of the importance of the number of hops), which could describe the relative differences shown in Fig. 4 and 5.

These previously discussed differences seem not to be a random occurrence—they also occur when considering the more complete parameter set of source and destination IP addresses, the measurement direction and the number of hops. Interestingly, from Fig. 8 and 9, the measurement direction seems equally important as the DSCP value.

III. Related Work

Before trying to understand whether the DSCP choice has an impact on delay (as we do here), it makes sense to investigate what happens to the field itself when packets carrying a non-zero DSCP value are sent across Internet. For instance, are they then consistently dropped (filtered, “blackholed”)? Indeed, we have found such consistent dropping, first in a smaller-scale study [6], and then in [2], but it was very rare (in the order of half a percent of all measured paths, depending on the DSCP value). As described in [7], fall-back logic to switch to DSCP=0 could be implemented for such cases.

While the blackholing finding was not confirmed by other work, the rest of the results are more similar: in line with [2], the authors of [8], [9] (focusing on mobile edge networks) and [10] find that the DSCP field is often rewritten, and particularly often zeroed, by routers along a path. Setting the field to zero means that any possible effect from the input DSCP value further downstream is eliminated, but otherwise this is not getting in the way of “normal” communication. While we did not document evidence of a DSCP choice being counterproductive in [2], the authors of [10] have identified cases of remarking that seem to reflect a historic interpretation of the header field, sometimes leading to an undesirable result which they call “priority inversion”. Even so, the overall recommendation in this and the other related work is that end systems should use the DSCP—if not to immediately benefit from the choice, then to provoke service providers to implement suitable policies to honor the DSCP setting.

IV. Conclusions

The DSCP setting should only play a role for traffic in the presence of some degree of congestion. It was therefore clear from the outset that a conclusive picture of the impact of the DSCP value can only be drawn if the large-scale ping-style measurement study that we have investigated here is complemented with smaller-scale studies that involve sending a larger amount of traffic. It was also not surprising that finding traces of a DSCP impact on the delay difference experienced by packets in our dataset is a difficult task.

We found some hints of a DSCP impact upon first inspection of the relative OWD values (the difference in One-Way Delay between pings carried out with different DSCP settings). Some of these hints were in line with the expected behavior (low priority traffic being treated as such, in the client-to-server direction of our measurement, with IPv4), whereas others were not (low priority traffic experiencing lower delay than the rest with IPv6). Investigating the delay as a function of the number of hops did not show a trend, and overall, the results seemed to be quite inconclusive. We therefore applied a heavier tool (a deep neural network) to try to obtain an idea of the importance that various factors may play. With this, we found that, among the source and destination IP addresses, measurement direction, the number of hops and the DSCP value itself, the number of hops and addresses play the biggest roles for the observed OWD difference—but the measurement direction and the DSCP value itself do play a significant role as well (about half as important as the others).

We can now conclude that the DSCP-specific OWD differences that we have seen in the raw data are probably not a coincidence, and setting the DSCP value probably does have some effect on the delay that packets experience. This further underlines the need for further studies with a large enough amount of traffic such that congestion would happen and the impact of the DSCP becomes significant.

Once more data is available, it makes sense to turn to machine learning as an analysis tool again. In this paper, we have investigated the impact of a limited set of input features: the source and destination IP addresses, direction, number of hops and DSCP values. In future work, we plan to incorporate other header fields, and consider representing IP addresses differently (as a geolocation instead of a number).

V. Acknowledgments

The authors would like to thank Andrea Merlina for his suggestions regarding the importance of input parameters.

References

[1] P. Jones, S. Dhesikan, C. Jennings, and D. Druta, "DSCP Packet Markings for WebRTC QoS," Internet Engineering Task Force, Internet-Draft draft-ietf-tsvwg-rtcweb-qos-18, Aug. 2016, work in Progress.

[2] R. Barik, M. Welzl, A. M. Elmokashfi, T. Dreibholz, and S. Gjessing, "Can WebRTC QoS work? a DSCP measurement study," in *30th International Teletraffic Congress (ITC 30)*, Vienna, Austria, Sep. 2018.

[3] R. Barik, M. Welzl, A. M. Elmokashfi, S. Gjessing, and S. Islam, "fling: A flexible ping for middlebox measurements," in *29th International Teletraffic Congress (ITC 29)*, Genoa, Italy, Sep. 2017.

[4] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817 (Experimental), RFC Editor, Fremont, CA, USA, pp. 1–25, Dec. 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6817.txt

[5] J. D. Olden and D. A. Jackson, "Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1-2, pp. 135–150, 2002.

[6] R. Barik, M. Welzl, and A. Elmokashfi, "How to Say That You'Re Special: Can We Use Bits in the IPv4 Header?" in *ANRW '16*, 2016, ISBN 978-1-4503-4443-2.

[7] H. T. Alvestrand, "Transports for WebRTC," Internet Engineering Task Force, Internet-Draft draft-ietf-rtcweb-transports-17, Oct. 2016, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-rtcweb-transports-17

[8] B. Trammell, M. Kuhlewind, P. De Vaere, I. R. Learmonth, and G. Fairhurst, "Tracking Transport-layer Evolution with PATHspider," in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '17. New York, NY, USA: ACM, 2017, pp. 20–26. [Online]. Available: http://doi.acm.org/10.1145/3106328.3106336

[9] A. Custura, A. Venne, and G. Fairhurst, "Exploring DSCP Modification Pathologies in Mobile Edge Networks," in *2017 Network Traffic Measurement and Analysis Conference (TMA)*, Jun. 2017, pp. 1–6. [Online]. Available: http://tma.ifip.org/wordpress/wp-content/uploads/2017/06/mnm2017_paper13.pdf

[10] A. Custura, R. Secchi, and G. Fairhurst, "Exploring dscp modification pathologies in the internet," *Computer Communications*, vol. 127, pp. 86 – 94, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366417312835