An Artifact Evaluation of NDP

Noa Zilberman
University of Oxford, UK
noa.zilberman@eng.ox.ac.uk

ABSTRACT

Artifact badging aims to rank the quality of submitted research artifacts and promote reproducibility. However, artifact badging may not indicate inherent design and evaluation limitations. This work explores current limits in artifact badging using a performance-based evaluation of the NDP [7] artifact. We evaluate the NDP artifact beyond the *Reusable* badge's level, investigating the effect of aspects such as packet size and random-number seed on throughput and flow completion time. Our evaluation demonstrates that while the NDP artifact is *reusable*, it is not *robust*, and we identify architectural, implementation and evaluation limitations.

CCS CONCEPTS

• **General and reference → Evaluation**; • **Networks → Data center networks**;

KEYWORDS

Reproducibility, Artifact Evaluation, Datacenters, Transport Protocols

1 INTRODUCTION

NDP, a novel data centre transport architecture, was proposed by Handley *et al.* [7], aiming to achieve both low latency and high throughput. NDP offers better short-flow performance than DCTCP [2] or DCQCN [16], achieving more than 95% of the maximum network capacity in a heavily loaded network, near-perfect delay and fairness in incast scenarios, minimal interference between flows to different hosts and effective prioritisation of straggler traffic during incast. For its contributions, the work won the ACM SIGCOMM'17 best paper award.

In this paper, we report an evaluation of the NDP artifact. The artifact was made available along with the publication [6], but was not badged; ACM SIGCOMM badging [14] started the following year. It is a high quality and easily reusable artifact, and is picked as an example of the challenges faced by researchers.

Our evaluation extends beyond the ACM *Reusable* badge level [1], focusing on the *robustness* of the artifact. In particular, we show how using sensitivity testing, limitations of the artifact are exposed. We provide examples of three types of shortcomings: limitations of the architecture, limitations of the implementation and limitations of the evaluation. These limitations are indicative of common evaluation pitfalls.

2 EVALUATION ENVIRONMENT

Simulation Environment. The simulation environment is based on htsim [12]. The simulator is provided as part of the NDP repository [6]. The code contains implementations of TCP NewReno (not SACK), a version of MPTCP [12], DCTCP [2], and PFC/DCQCN [16]. The DCQCN implementation is based on the DCTCP code and is window based rather than rate based. We use the simulation environment "as is", except for the minimum amount of changes required to evaluate a specific aspect, e.g., setting the packet size or changing packet size distribution. All the simulations were done on a Xeon E5-2660 v4 server, using 256GB of DDR4-2400 RAM, running at 3.2GHz, and using Ubuntu 14.04, kernel version 3.13.0-32-generic.

Hardware Environment. The Implementation of NDP switch on NetFPGA SUME [19] is based on the NetFPGA Reference Switch design. The NDP switch supports both NDP and non-NDP traffic. We compare the performance of NDP with the NetFPGA Reference Switch, running traffic through both designs. Both designs are synthesized using NetFPGA-SUME release 1.7.1. Our setup is composed of two identical NetFPGA SUME boards, one configured as OSNT [3], an open source network tester (release 1.7.0), and the other as the device under test. The boards are hosted within two identical i7-6700K machines running Ubuntu 14.04; although the host setup has no impact on the test.

3 THE NDP ARTIFACT

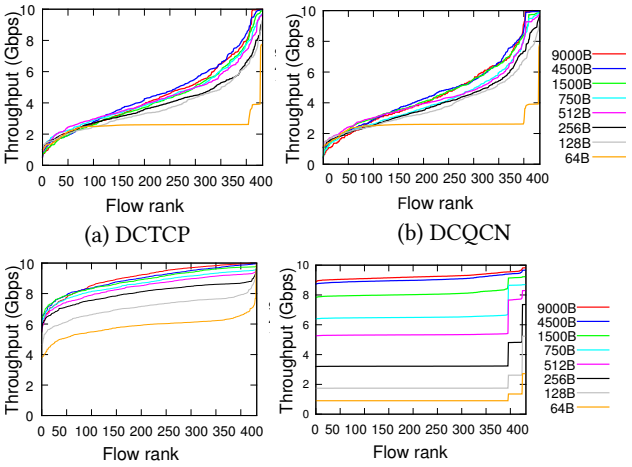
Unlike so much published work, the NDP artifact is open source and available [6]. The artifact contains a simulation environment, an implementation of NDP switch in both P4 and for the NetFPGA platform, and an implementation of the host side. No special licenses are required, and there are no ethical encumbrances. Current badging rules [1] consider the artifact *Available*.

In this work, we use NDP repository commit dated January 8th, 2018. The host side implementation was released after the completion of this work. No evaluation of the P4 switch was included in [7] and it was not evaluated as part of this work either. Both the NDP switch on NetFPGA and the simulation environment are easy to run, detailed documentation is provided in the repository's wiki. The code itself is not thoroughly commented. We only had to make a minor adjustment to the Makefile to be able to run experiments. Thus, the artifact can be deemed *Functional* (in addition to *Available*).

The artifact provides scripts for reproducing some of the paper's experiments, e.g., six of the simulation-based experiments. We approached the authors and received access to scripts used to run the rest of the simulations. The repository used in our evaluation did not include scripts for hardware evaluation. We believe that if the code available to us was made available to an artifact evaluation committee[1], it would have been deemed *Reusable* (in addition to *Available* and *Functional*).

Running the simulation examples provided in the NDP repository, we obtain results similar to those reported in the paper, except for cases where the authors acknowledge differences between the

[1] As the committee sometimes allows authors to update their artifacts based on reviewers' feedback.



(a) DCTCP (b) DCQCN

(c) MPTCP (d) NDP

Figure 1: Per-flow throughput, permutation traffic matrix, 432-node Fat-Tree. DCTCP, DCQCN and MPTCP show little sensitivity to packet size, while NDP loses throughput as packet size decreases.

provided scripts and the publication. As we use the artifact provided by the authors, this is a measure of results replication, rather than results reproduction. A full report of our artifact evaluation is available in [17] as well as the dataset [18].

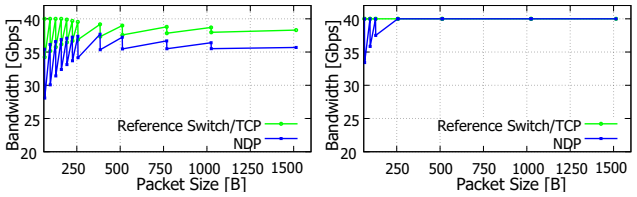
4 ARCHITECTURAL LIMITATIONS

One of the core concepts in NDP is packet trimming, where the header of a packet, payload removed, is forwarded when a queue becomes full. Forwarding just a small part of a packet reduces the load on the network, making it lossy for payload but not metadata. The ratio between a packet header and the entire packet is called the *compression ratio*.

NDP [7] briefly notes that packets smaller than MTU will achieve lower compression ratio upon trimming. The NDP artifact uses fixed packet size of 9000B in many of the simulations. However, 1500B is currently a common maximum transmission unit (MTU) in DCN [5, 11], and some cloud workloads have 97.8% packets of less than 576B [4]. This leads us to explore the throughput robustness of NDP.

We repeat the per-flow throughput experiment from [7], using the original scripts, and receive similar results. The experiment uses a 432-node fat-tree configuration, where each server has a single long-running connection to another random server, and each server has exactly one incoming connection. The server exchanges single size packets, originally a constant 9000B.

We then maintain the same experimental environment, but vary the packet size. We do not change any parameters within the simulation environment: wishing to find if the results will differ if the workload suddenly changes, not to identify the optimum setup for the new workload. Our results, presented in Figure 1, show that DCTCP and DCQCN are agnostic to packet size, with the exception of 64B, which may be a corner case of the simulation environment. MPTCP is also generally unaffected by packet sizes, though at packet sizes of 256B or less it starts to exhibit a higher throughput loss.



(a) 150MHz (b) 180MHz

Figure 2: The throughput of a Reference Switch compared with NDP switch. At 150MHz clock frequency the Reference Switch outperforms NDP. At 180MHz, the Reference Switch always supports line rate, while NDP drops traffic at some packet sizes.

In contrast with the other three protocols, NDP is very sensitive to packet size. When the packet size is changed from 9000B to 1500B, the minimum and average throughput drop by approximately 14%. Using 750B packets reduces the minimum and average throughput 28% and 29.5%, respectively. Smaller packet sizes lead to still further performance loss. This experiment indicates that using a workload such as [4] would result in low throughput when using NDP. The NDP paper [7] does utilise Facebook web workload [13], which uses small packets, but reports flow completion time, not throughput.

This is an example of a limitation of an architecture that can be exposed by varying an experiment's parameters, but will not be evident if the artifact evaluation is limited to reusability. Furthermore, varying the parameters explores core properties of a concept or a solution; ones that may be overlooked or render an approach unfixable.

5 IMPLEMENTATION LIMITATIONS

Research papers, and the associated artifacts, often evaluate highly complex aspects of a solution. At the same time, basic properties of an implementation may not be reported (nor possibly even validated). In this section, we evaluate the throughput performance of the NDP switch, in comparison with the NetFPGA Reference Switch, the baseline design. The test is based on NDP's switch code, but without available evaluation tests.

NetFPGA SUME Reference Switch supports $4 \times 10GE$, which is roughly 59.52Mpps for 64B packets. It is expected to support full line rate at 120MHz for 64B packets, and at 180MHz for 65B packets. The default frequency of the Reference switch is 200MHz, providing a small amount of headroom.

In the following experiment, also reported in [20], we study the throughput of the NetFPGA Reference Switch and the NDP switch for different packet sizes, at different core clock frequencies. We connect $4 \times 10GE$ links between OSNT and the NetFPGA board; OSNT generates traffic at full line rate. In every experiment, only one packet size is used, and a hundred million packets are sent. We use either TCP or NDP headers, generated by a script written by NDP's authors and used in [7]. The experiment is repeated multiple times, and replicates with minor variations.

We benchmark the designs at 150MHz[2], 180MHz and 200MHz. Designs are expected to support full line rate using the last two frequencies. We scan a range of packet sizes, from 64B to 1514B[3].

[2]Minimal frequency, see [9].
[3]Packet size excludes FCS.

Packet sizes are chosen on 32B granularity: perfectly aligned packet sizes, and misaligned packet sizes ($32B \times n + 1$). The results of our evaluation are presented in Figure 2, for core clock frequencies of 150MHz and 180MHz. The graph for 200MHz clock frequency is omitted as it adds little data. As the results show, the Reference Switch consistently outperforms NDP switch (using NDP packets). Not only does it achieve higher throughput at 150MHz, but it also achieves full line rate for all packet sizes at 180MHz in contrast, NDP does not achieve full line rate either at 180MHz nor at 200MHz (for 65B, 97B packets).

The NDP's performance loss is because the design is not fully pipelined; instead using a state machine that requires more clock cycles than packet propagation time. NDP's authors confirmed this performance loss is expected in their implementation, and that they had not evaluated throughput across a range of packet sizes.

While this result is interesting, as it shows that TCP may outperform NDP, it reflects only on the quality of implementation and evaluation, rather than on the architecture. The performance limitation would be addressed by changing the hardware implementation, without a change to any of the NDP's concepts. We would encourage researchers to invest in validation tests of this type, both to uncover design weaknesses, and as any follow up work will outperform their design for no good reason.

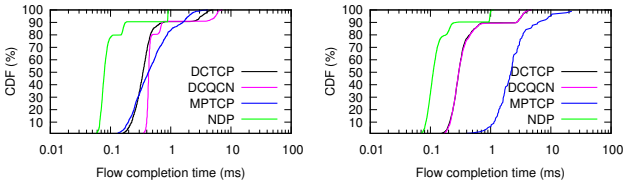
6 EVALUATION LIMITATIONS

A limited evaluation may hide problems in solution, evaluation environment or both. In the following example, flow completion time is evaluated while varying two parameters: workload and (random number) seeds. Our evaluation is slightly different to the experiments presented in the NDP paper as we use the same setup as [7, Fig. 15] ("FCT for 90KB flows with random background load, 432 node FatTree.") and [7, §4], but with variable flow size.

Our experiments utilise Web, Hadoop, and Cache flow size distributions derived from Roy *et al.* [13]. We explore both a fully loaded setup and an over-subscribed one. In the fully loaded setup, there is one outgoing flow from each node in the system, and one incoming flow. In the over-subscribed setup, there is a ratio of 4:1 of flows to nodes. These two configurations are part of the NDP artifact and described in [7]. We do not change the artifact's default packet size settings. We evaluate using 50 seeds, where our number of experiments is time and resources limited.

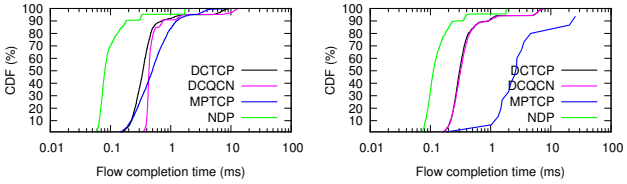
The results of our simulations largely reaffirm the claims made in [7]: The FCT of NDP outperforms other protocols both in the fully utilised (1:1) and the oversubscribed (4:1) scenarios, for different workloads. Figures 3, 4, 5 show the FCT of NDP, MPTCP, DCTCP and DCQCN under different workloads, using the default seed. We further study NDP's FCT as a function of flow size under the different workloads, and find that they are proportional, though the minimum FCT is not necessarily for the minimum flow size.

A surprising result that we find is that sometimes NDP flows (and MPTCP flows) time out. This means that some flows are never completed. The seed used in the original NDP evaluation, using the web workload, does not lead to any timeouts, but in both fully utilised configuration (seeds 26, 38) and oversubscribed configuration (seeds 8, 36) we find two seeds that lead to a single timeout. One specific seed in the oversubscribed scenario (10) has 153 timeouts.



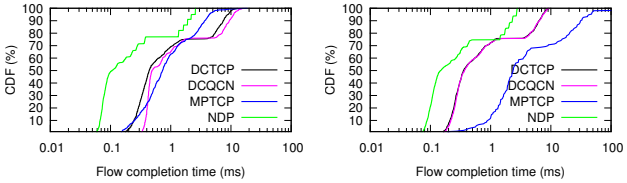
(a) 1:1 Ratio (b) 4:1 Ratio

Figure 3: FCT using Web distribution used in [7].



(a) 1:1 Ratio (b) 4:1 Ratio

Figure 4: FCT using Hadoop distribution extracted from [13].



(a) 1:1 Ratio (b) 4:1 Ratio

Figure 5: FCT using Cache distribution extracted from [13].

Furthermore, the number of completed flows is just 353, compared with 1636 to 1690 in all other runs. The command used to trigger this scenario is:

```
<path>/htsim_ndp_perm_shortflows -o <log file> -strat perm -nodes 432
-conns 1728 -cwnd 23 -q 8 -seed 10  > <debug file>
```

To reproduce this result using the original NDP repository, extend run time in *main_ndp_perm_shortf-lows.cpp* to two seconds. We have made the authors of NDP aware of this issue, but at the time of writing, it was not yet resolved.

We also report that we have found timeouts for MPTCP. In the over-subscribed scenario, 27 out of 50 seeds lead to timeouts. The number of completed flows ranged from 2 to 1055 with the median being 371. We didn't find a correlation between flow timeouts and the number of flows completed by MPTCP.

Next, we repeat the same experiment using Facebook's Cache workload. The Cache workload is selected as NDP had the highest FCT under this workload, and we want to check whether high FCT leads to greater performance variance under different seeds. When flows complete, the performance gap between seeds is small: the minimum FCT ranges between 55.8μs and 63.3μs (13%), and the maximum FCT ranges from 2.57ms to 2.73ms (6%).

In the fully utilised configuration we find 37 seeds (out of 50) with flow timeouts. In the oversubscribed scenario, 40 seeds lead to timeouts. Five of the seeds lead to more than twenty timeouts, and in the extreme case, there are 176 timeouts (seed=22). Here, just 111

flows are completed, compared with a median of 1222. We don't detect timeouts running DCTCP or DCQCN.

To look differently at the experiment, we consider the throughput, i.e., the number of bytes in completed flows. In the fully utilised configuration the average throughput per experiment is 780MB. In the oversubscribed scenario, the average is 718MB, and for seeds with no timeouts the throughput is around 780MB. However, timeouts often lead to significant throughput loss: in experiments with tens of timeouts, the throughput drops by 20% to around 580MB, while in the worst case (seed 22) the throughput drops to 75MB: an order of magnitude lower performance. To complete this evaluation, we study FCT under zero load and find no flow timeouts, while the number of flows completed is largely the same for all seeds.

We try and separate results that are a consequence of an algorithm, an implementation and a simulation environment. We don't have confirmed answers. Given the timeouts we see in MPTCP, and the code shared between NDP and MPTCP, it is possible that the cause for timeouts lies in the simulator. However, without a sensitivity analysis, this weakness of NDP was not previously reported.

7 RELATED WORK

Ours is not the first to explore the work of NDP; as NDP has become a staple comparison approach to data centre traffic control, due in no-small part to the availability of implementation; many others identify drawbacks of NDP in order to highlight the advantages of their own approach. Examples have included Homa [8]: observing that NDP's network utilisation limit was lower than comparison solutions (pias, pHost, pFabric and Homa), and that NDP had worse comparative median and 99% slowdown as a function of message size, yet Homa also failed to run their NDP comparison experiments with network loads beyond 70%. Shoal [15] demonstrated an improved performance when compared to NDP, and that NDP does not perform better than DCTCP or DCQCN for specific workloads. MDTCP [10] reproduced some of NDP's results using htsim, and had shown that for some scenarios, NDP's FCT was worse than DCTCP. Unsurprisingly, few works comparing to NDP present it in *good-light* as the authors of new publications may not publicise comparable or unexciting results.

8 CONCLUSION

In this paper, we have reported the results of an artifact evaluation of NDP. Our evaluation has focused on the performance of NDP, and in particular on the robustness of the artifact. The artifact is reusable and supports the results presented in [7]. However, badging up to *Reusable* level may not expose flaws in architecture, implementation and evaluation, as sensitivity testing and robustness experiments of the artifact have exposed. While our evaluation went beyond the current *Reusable* badging level, we do not consider it complete; a large gap remains between research-level and production-level evaluation, and researchers need to balance quality, time and resources. The broader implications of our results, and proposed improvements to artifact evaluation are introduced in [21].

Additional Information. A detailed report of NDP's performance evaluation is available in [17]. Our dataset is available at [18].

ACKNOWLEDGMENTS

We thank Andrew W. Moore, Marcin Wójcik, Gianni Antichi and Costin Raiciu for their assistance in the reproduction of NDP's evaluation and the discussion of the results. We thank Changhoon Kim and Robert Soulé for their feedback on the NDP reproducibility technical report [17].

This work was partly funded by the Leverhulme Trust (ECF-2016-289) and the Isaac Newton Trust.

REFERENCES

[1] ACM. 2018. Artifact Review and Badging. https://www.acm.org/publications/policies/artifact-review-badging.
[2] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center TCP (DCTCP). In *ACM SIGCOMM*. 63–74.
[3] Gianni Antichi, Muhammad Shahbaz, Yilong Geng, Noa Zilberman, Adam Covington, Marc Bruyere, Nick McKeown, Nick Feamster, et al. 2014. OSNT: Open source network tester. *IEEE Network* 28, 5 (2014), 6–12.
[4] Kari Clark, Hitesh Ballani, Polina Bayvel, Daniel Cletheroe, Thomas Gerard, Istvan Haller, Krzysztof Jozwik, Kai Shi, et al. 2018. Sub-nanosecond clock and data recovery in an optically-switched data centre network. In *2018 European Conference on Optical Communication (ECOC)*. IEEE, 1–3.
[5] Daniel Firestone, Andrew Putnam, Sambhrama Mundkur, Derek Chiou, Alireza Dabagh, Mike Andrewartha, Hari Angepat, Vivek Bhanu, Adrian Caulfield, Eric Chung, et al. 2018. Azure accelerated networking: SmartNICs in the public cloud. In *NSDI*. Renton, WA, USA, 51–66.
[6] Mark Handley, Costin Raiciu, Alexandru Agache, Andrei Voinescu, Andrew W Moore, Gianni Antichi, and Marcin Wójcik. 2017. *NDP*. Repository, https://github.com/nets-cs-pub-ro/NDP [accessed Jan. 2018].
[7] Mark Handley, Costin Raiciu, Alexandru Agache, Andrei Voinescu, Andrew W Moore, Gianni Antichi, and Marcin Wójcik. 2017. Re-architecting datacenter networks and stacks for low latency and high performance. In *SIGCOMM*. ACM, Los Angeles, CA, USA, 29–42.
[8] Behnam Montazeri, Yilong Li, Mohammad Alizadeh, and John Ousterhout. 2018. Homa: A Receiver-Driven Low-Latency Transport Protocol Using Network Priorities. In *SIGCOMM*. ACM.
[9] NetFPGA. 2017. *NetFPGA-SUME-live, issue 36: 10G port - Attachment unit - Rx side - inefficiency*. https://github.com/NetFPGA/NetFPGA-SUME-live/issues/36.
[10] Dejene Boru Oljira, Karl-Johan Grinnemo, Anna Brunstrom, and Javid Taheri. 2018. MDTCP: Towards a Practical Multipath Transport Protocol for Telco Cloud Datacenters. In *NOF*. 9–16.
[11] Igor Pagliai. 2017. My personal Azure FAQ on Azure Networking (v3). https://blogs.msdn.microsoft.com/igorpag/2017/04/06/my-personal-azure-faq-on-azure-networking-v3/.
[12] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. 2011. Improving datacenter performance and robustness with multipath TCP. In *SIGCOMM Comput. Commun. Rev.*, Vol. 41. 266–277.
[13] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C Snoeren. 2015. Inside the social network's (datacenter) network. In *SIGCOMM Comput. Commun. Rev.*, Vol. 45. ACM, 123–137.
[14] Damien Saucez, Luigi Iannone, and Olivier Bonaventure. 2019. Evaluating the artifacts of SIGCOMM papers. *SIGCOMM Comput. Commun. Rev.* 49, 2 (2019), 44–47.
[15] Vishal Shrivastav, Asaf Valadarsky, Hitesh Ballani, Paolo Costa, Ki Suh Lee, Han Wang, Rachit Agarwal, and Hakim Weatherspoon. 2019. Shoal: A Network Architecture for Disaggregated Racks. In *NSDI*.
[16] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. 2015. Congestion control for large-scale RDMA deployments. *SIGCOMM Comput. Commun. Rev.* 45, 4 (2015), 523–536.
[17] Noa Zilberman. 2019. *An evaluation of NDP performance* . Technical Report UCAM-CL-TR-933. University of Cambridge, Computer Laboratory. https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-933.html
[18] Noa Zilberman. 2020. An Artifact Evaluation of NDP, Dataset. https://doi.org/10.5287/bodleian:Ov8j62ENK
[19] Noa Zilberman, Yuri Audzevich, G. Adam Covington, and Andrew W. Moore. 2014. NetFPGA SUME: Toward 100 Gbps as Research Commodity. *IEEE MICRO* 34, 5 (Sept. 2014), 32–41.
[20] Noa Zilberman, Gabi Bracha, and Golan Schzukin. 2019. Stardust: Divide and Conquer in the Data Center Network. In *NSDI*. Boston, MA, USA, 141–160.
[21] Noa Zilberman and Andrew W Moore. 2020. Thoughts about Artifact Badging. *SIGCOMM Comput. Commun. Rev.* 50, 2 (2020).